

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

Bakalářská práce

Mobilní aplikace s rozšířenou realitou pro budovu FAV

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Vladimír HOLÝ**
Osobní číslo: **A19B0247P**
Studijní program: **B0613A140015 Informatika a výpočetní technika**
Specializace: **Výpočetní technika**
Téma práce: **Mobilní aplikace s rozšířenou realitou pro budovu FAV**
Zadávací katedra: **Katedra informatiky a výpočetní techniky**

Zásady pro vypracování

1. Prostudujte vhodné frameworky pro rozšířenou realitu (AR) pro platformu Android.
2. Proveďte srovnání těchto frameworků z různých hledisek použití (jaké typy objektů lze pro AR využít, komplexnost a snadnost použití aj.).
3. Navrhněte mobilní aplikaci využívající rozšířené reality k mapování pohyblivých obrazů do clip ráků na chodbách budovy FAV.
4. Navrhněte možnost přidávání dalších animací (pro administrátory) do databáze pohyblivých obrazů. Přidané pohyblivé obrazy si budou moci uživatelé stáhnout do své aplikace, aniž by bylo nutné vydávat nový build aplikace.
5. Oba návrhy implementujte pro vybranou platformu a aplikaci otestujte na reprezentativním vzorku dat.

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. Bc. Zuzana Majdišová, Ph.D.**
Katedra informatiky a výpočetní techniky

Datum zadání bakalářské práce: **3. října 2022**
Termín odevzdání bakalářské práce: **4. května 2023**

L.S.

Doc. Ing. Miloš Železný, Ph.D.
děkan

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

V Plzni dne 25. října 2022

Prohlášení

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

V Plzni dne 4. května 2023

Vladimír Holý

Abstract

This thesis deals with the development of an Android mobile app, using features of Augmented reality for the purpose of mapping virtual objects onto physical paintings. This app is used to enhance guided tours of the Faculty of Applied Sciences building, where the paintings mentioned above are located. As part of this thesis, a complimentary administrator application was also developed, which can be used to easily add, remove or otherwise manage the mapped virtual objects by system administrators. After conducting an initial market analysis of similar or analogous applications currently available, the thesis proceeds to introduce and explain the tools and processes used during the applications development, and their eventual testing and deployment.

Abstrakt

V této bakalářské práci se zabývám vývojem aplikace na mobilní zařízení Android využívající prvky rozšířené reality k mapování virtuálních objektů do fyzických obrazů. Aplikace slouží k zpestření prohlídky budovy Fakulty aplikovaných věd v Plzni, kde se obrazy nachází. Součástí řešení je i administrátorská aplikace, pomocí které mohou administrátoři snadno upravovat, mazat a přidávat další objekty. Po počátečním průzkumu dostupných nástrojů a existujících podobných aplikací seznamuji čtenáře s využitými nástroji pro dosažení cíle, implementací a s testováním obou aplikací.

Poděkování

Rád bych poděkoval vedoucí mé bakalářské práce Ing. Bc. Zuzaně Majdišové, Ph D., za poskytnutí velkého množství rad, připomínek a doporučení během celého průběhu práce a za její dobrou vůli a trpělivost při společných konzultacích.

Obsah

1	Úvod	10
2	Úvod do problematiky	11
2.1	Android	11
2.1.1	API level	11
2.2	Podobné aplikace	12
2.2.1	Jarit	12
2.2.2	WebAR	12
2.2.3	Artivive	12
2.2.4	Google Lens	13
2.2.5	AR Books LibrARy	13
2.2.6	Shrnutí	13
2.3	Frameworky pro rozšířenou realitu	14
2.3.1	Vuforia Engine	14
2.3.2	ARCore	14
2.3.3	ARToolKit	15
2.3.4	ARToolKitX	15
2.3.5	EasyAR Sence	16
2.3.6	DroidAR	16
2.3.7	AR.js	16
2.3.8	Shrnutí	17
3	Návrh aplikací	18
3.1	Aplikace na Android	18
3.1.1	Spuštění aplikace	18
3.1.2	Běh aplikace	19
3.1.3	Ukončení aplikace	20
3.2	Administrátorská aplikace	21
3.2.1	Spuštění aplikace	21
3.2.2	Běh aplikace	21
3.2.3	Ukončení aplikace	24
4	Data aplikací	25
4.1	Umístění dat	25
4.1.1	Ares	25
4.1.2	Google Drive	26

4.1.3	Shrnutí	27
4.2	Knihovny pro XLSX formát	28
4.2.1	Apache POI	28
4.2.2	FastExcel	28
4.2.3	DOCKX4J	28
4.2.4	Shrnutí	29
4.3	Rozhraní API	29
4.3.1	Příklad s číšníkem	29
4.3.2	Výhody API	29
4.3.3	Google Drive API	30
4.3.4	Autorizační protokol OAuth	30
4.4	Struktura dat na Google Drive	31
4.4.1	Složka <code>images</code>	31
4.4.2	Složka <code>models</code>	31
4.4.3	Složka <code>textures</code>	31
4.4.4	Soubor <code>image_database.imgdb</code>	32
4.4.5	Soubor <code>database.xlsx</code>	32
5	Google ARCore nástroje	35
5.1	Kontrola kvality obrazu	35
5.2	Vytváření databáze obrazů	35
6	Implementace	37
6.1	Mobilní aplikace	37
6.2	Aplikace pro administrátory	37
6.3	Řešené problémy	38
6.3.1	Přesun z Ares na Google Drive	38
6.3.2	Stažení modelů do telefonu	38
6.3.3	Přehrávání videí	39
6.3.4	Názvy modelů	39
6.3.5	Synchronizace	39
6.3.6	Generování databáze obrazů	40
7	Testování	41
7.1	Mobilní aplikace	41
7.1.1	Otočení displeje	41
7.1.2	Připojení k internetu	41
7.1.3	Zamítnutí práv ke kameře	42
7.1.4	Vymazání dat aplikace	43
7.1.5	Uspání aplikace	43

7.1.6	Vyrušení aplikace	44
7.1.7	Nestažený model	44
7.1.8	Testování na různých zařízeních	44
7.2	Administrátorská aplikace	45
7.3	Skenování obrazů	45
7.4	Shrnutí	46
8	Závěr	47
	Literatura	48
	Uživatelská příručka	52

1 Úvod

Vývoj technologií se v moderní době posouvá stále rapidně dopředu. To, co se lidstvu před několika lety zdálo jako nemožné, je v dnešní době zcela normálně využívané každý den. S postupem času se technologie stávají i dostupnějšími pro většinu lidí. Příkladem může být i technologie rozšířené reality (AR), kterou může dnes využít každý člověk, který vlastní chytrý telefon. Co to ale ta rozšířená realita vlastně je?

Rozšířená realita je technologie, která dokáže promítnout 2D a 3D objekty do fyzického světa. Jde o vytvoření a umístění virtuálního objektu v reálném čase na reálné místo. Technologie slouží k rozšíření fyzického světa a jeho usnadnění. Celý proces je kombinací počítačového vidění, zpracování obrazu a počítačové grafiky. Technologie využívá pro orientaci ve fyzickém prostředí velké množství hardwarových komponentů. Příkladem jsou různé senzory, akcelerometr a gyroskop pro snímání vychýlení a pohybu zařízení, GPS pro lepší orientaci v prostoru a optické senzory nebo kamery pro snímání okolního prostředí.

Mým úkolem je za pomoci této technologie vytvořit aplikaci, která bude používaná v budově Západočeské univerzity v Plzni, konkrétně v budově Fakulty aplikovaných věd (FAV). Pomocí této aplikace si bude moci uživatel zobrazovat různé pohyblivé obrazy v reálných clip rámech. Bude se jednat o animované obrazy (GIF a podobné), videa a v některých případech i 3D modely. Aplikace by měla primárně sloužit k oživení prohlídky budovy při akcích pro studenty středních a základních škol. Součástí práce je i administrátorská aplikace, pomocí které budou moci administrátoři upravovat, mazat a přidávat do aplikace další pohyblivé obrazy. Součástí řešení je i zajištění systému, který by se staral o to, aby si mohl uživatel stáhnout nová nebo upravená data do zařízení bez nutnosti vydání nové verze aplikace.

V první části práce se budu věnovat operačnímu systému Android a představím podobné aplikace. Dále se zaměřím na první dva body zadání, tedy prostudování a srovnání různých frameworků pro rozšířenou realitu na zařízení Android. Další části věnuji samotným aplikacím, jejich návrhům, implementaci a použití. V předposlední části otestuji jak mobilní aplikaci, tak aplikaci pro administrátory. Nakonec celou práci shrnu a zhodnotím, zdali se mi podařilo splnit všechny body zadání nebo nikoli.

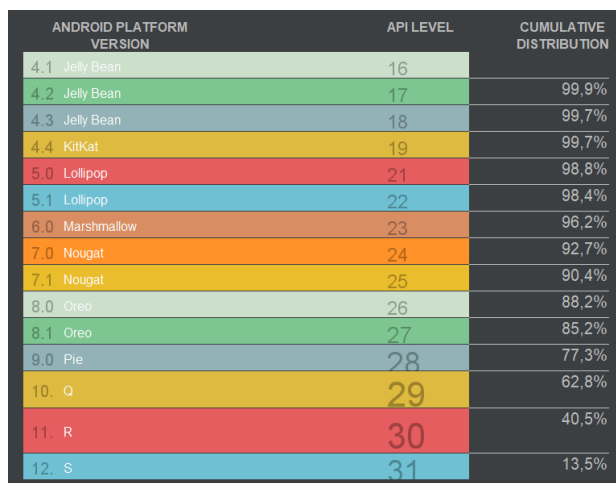
2 Úvod do problematiky

2.1 Android

Android je operační systém od společnosti Google, který je založený na Linuxovém jádře [1]. Android je rozšířen téměř po celém světě a patří k nejpopulárnějšímu operačnímu systému na světě. Operační systém Android se dnes nachází ve spoustě chytrých zařízeních, jako jsou mobilní telefony, tablety, televize a další. Podle statistik z roku 2023 používá Android přes 2,5 miliard aktivních uživatelů, což tvoří přibližně 75% podílu na světovém trhu [2]. První verze Androidu byla vydána v roce 2008 a nesla název Apple Pie. K dnešnímu dni je nejnovější verzí Androidu verze 14, která byla uvedena v únoru roku 2023 [3].

2.1.1 API level

API level u operačního systému Android je číselné vyjádření verze Androidu. Používá se při programování aplikací hlavně pro zjednodušení názvu verze Androidu. Příkladem může být verze 5.1 Lollipop, kterou lze vyjádřit jako API level 22 [4]. Programátor totiž při vývoji nové aplikace musí určit minimální verzi API, na které aplikace poběží. Novější verze mají více nástrojů a vlastností, zatímco staré verze jsou více rozšířené. K určení správného kompromisu slouží tabulka distribuce jednotlivých verzí, viz obr. 2.1.



ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.1 Jelly Bean	16	
4.2 Jelly Bean	17	99,9%
4.3 Jelly Bean	18	99,7%
4.4 KitKat	19	99,7%
5.0 Lollipop	21	98,8%
5.1 Lollipop	22	98,4%
6.0 Marshmallow	23	96,2%
7.0 Nougat	24	92,7%
7.1 Nougat	25	90,4%
8.0 Oreo	26	88,2%
8.1 Oreo	27	85,2%
9.0 Pie	28	77,3%
10. Q	29	62,8%
11. R	30	40,5%
12. S	31	13,5%

Obrázek 2.1: Snímek obrazovky z aplikace Android studio ukazující rozšíření verzí Android

2.2 Podobné aplikace

Protože v dnešní době je rozšířená realita již velice známá, existuje na trhu hned několik podobných aplikací. Snažil jsem se najít takové aplikace, kde dochází nejprve k rozpoznání určitého obrazu nebo znaku a k následnému mapování objektů do obrazu. Stejně jako je tomu u mé aplikace. Zároveň jsem se snažil, aby aplikace nebyly všechny stejné, ale demonstrovaly využití technologie AR v různých odvětvích. Z těchto aplikací si lze brát inspiraci pro návrh a vývoj aplikace.

2.2.1 Jarit

První aplikací je aplikace s názvem Jarit [5]. Jarit slouží k zobrazení jídel z jídelního menu v telefonu přímo na stole před hosty restaurace. Zapotřebí je zmíněná aplikace a speciální značka, která volně leží na stole. Pomocí aplikace si uživatel vybere z menu jídlo, které si chce zobrazit a namíří telefon na zmíněnou značku. Aplikace značku rozezná a namapuje na ni model jídla, který si uživatel zvolil. Značkou je možné otáčet a prohlédnout si tedy jídlo ze všech stran.

2.2.2 WebAR

Druhou aplikací je aplikace WebAR [6] od společnosti AirCards. Jedná se o webovou aplikaci umožňující využití AR technologie na všech zařízeních. Společnost AirCards nabízí hned několik možností využití aplikace. Příkladem mohou být hry využívající AR, přemístění do různých lokací nebo 'virtuální' vizitky, které jsou nejvíce zajímavé. Jedná se o fyzické vizitky, které kromě údajů obsahují i QR kód, který přenese uživatele na aplikaci WebAR. V okně aplikace se mu otevře pohled kamery, kterým uživatel namíří na vizitku. Vizitka se rozpozná a namapuje se do ní virtuální obsah. Obsahem mohou být odkazy na sociální sítě, videa nebo obrázky. Obsah nemusí být nutně mapován pouze do vizitky ale i do jejího nejbližšího okolí. Stále si ale drží stejnou pozici vůči vizitce i při hýbání s vizitkou.

2.2.3 Artivive

Artivive [7] je mobilní aplikace, která se nejvíce podobá mé aplikaci. Aplikace slouží pro oživení umění, hlavně obrazů a maleb. Umělcům tak dodává další možnosti, jak svoje díla posunout na vyšší úroveň. Uživatel se při spuštění aplikace zobrazí pohled kamery, kterým sleduje zmíněné obrazy. Do obrazů se mapují různé animace, modely nebo se celý obraz překryje jiným. Ty do

aplikace mohou přidávat samotní autoři děl. Aplikace se používá v muzeích a galeriích, kterým přidává velice moderní a atraktivní dojem. V mém případě při využití této aplikace by budova FAV sloužila jako pomyslná galerie obrazů, které by návštěvník skenoval.

2.2.4 Google Lens

Aplikace Google Lens [8] je známá aplikace od společnosti Google, která slouží k rozpoznávání obrazů. Aplikace má mnoho schopností, dokáže vyhledávat na internetu podle shody obrazu, naskenovat text nebo najít místo na mapě podle fotky. Pro mě nejdůležitější vlastností je překládání textu. Uživatel nejprve za pomoci aplikace vyfotí požadovaný text k přeložení, ta ho následně z obrazu získá a převede do textové podoby. Dále aplikace text přeloží a překryje původní text boxem s přeloženým textem. Rozdílem oproti mé aplikaci je, že Google Lens pracuje pouze s jedním snímkem. Stále se ale jedná o rozpoznávání, v tomto případě znaků textu, z obrazu a následného mapování objektu.

2.2.5 AR Books LibrARy

Jak již název aplikace AR Books LibrARy [9] napovídá, jedná se o spojení knížek s technologií rozšířené reality. Uživatel si zakoupí speciální knížku a stáhne požadovaný software. Na stránkách v knížce se nacházejí obrázky, které uživatel skenuje staženou aplikací. Do obrázků se mu následně rámují jiné obrázky, videa nebo modely k tématu, které zrovna čte. Knížky jsou zaměřené na vzdělání dětí a studentů. Existují tedy knížky o lidském těle, kde se zobrazují modely lidských orgánů, knížka o geometrii, kde si uživatel může zobrazovat 3D geometrické modely, nebo fyzikální knížka, kde jsou ukázková videa a videa pokusů.

2.2.6 Shrnutí

Jak již bylo zmíněno, nejvíce inspirace lze čerpat z aplikace Artivive, jež je pomalu až totožná s mé aplikací. AR Books LibrARy je zase krásným příkladem, kdy se mapují nejenom modely, ale zároveň i obrázky a videa, stejně tedy jako v mé aplikaci. Aplikace Jarit je zase pěkným příkladem zobrazování 3D modelů. U aplikace WebAR, konkrétně u oživení vizitek, je možné brát inspiraci ve využití nejenom prostoru v obraze, ale i kolem něj.

2.3 Frameworky pro rozšířenou realitu

Prvním úkolem ze zadání je vybrání vhodného frameworku nebo SDK pro samotnou realizaci. Hlavní podmínkou je, aby plně podporoval platformu Android a měl dostatečné funkce a vlastnosti k realizaci projektu.

2.3.1 Vuforia Engine

Prvním a jedním z nejvíce rozšířených SDK pro tvorbu AR aplikací je Vuforia [10]. Vuforia patří pod společnost PTC Inc.. Vuforia je multiplatformní a podporuje tedy zařízení s platformou Android, iOS a UWP (Universal Windows Platform) od firmy Microsoft. Vuforia obsahuje nástroje pro rozpoznávání a sledování obrazů, 3D objektů a okolního prostředí, vkládání virtuálních objektů, jako jsou obrazy, videa a 3D modely, do scény reálného světa a mnoho dalšího. Vuforia nabízí API, které rozšiřuje multiplatformní engine Unity, který je pro tvorbu aplikací s tímto SDK velice vhodný. Vuforii lze využít i v jiných vývojových prostředích, jako jsou například Android Studio pro platformu Android nebo Xcode pro iOS. Výhodou je, že Vuforia nabízí webovou službu 'Cloud Services', která umožňuje snadnou správu databáze objektů. V základní bezplatné verzi pro nekomerční využití pojme databáze objektů tisíc položek a je možné provést pouze tisíc rozpoznávání obrazů za měsíc, což by mohlo být velkou nevýhodou při konání větší akce, jako je např. den otevřených dveří. Více rozpoznávání za měsíc a větší databázi je možné dokoupit. Vuforia podporuje Android zařízení s minimální verzí 8.0 (API 26). Vuforia také podporuje 3D modely v mnoha formátech jako je například OBJ, STL nebo JT [11].

2.3.2 ARCore

Druhým velmi rozšířeným SDK pro rozšířenou realitu je ARCore [12] od společnosti Google. ARCore je primárně zaměřený na Android zařízení, ale podporuje i zařízení s operačním systémem iOS. Oproti Vuforii má ARCore méně nástrojů, ale ty potřebné pro práci, rozpoznávání, sledování a mapování objektů, zde jsou. K renderování 2D a 3D objektů je používáný interface OpenGL. Díky němu lze renderovat obrazy, videa a 3D modely různých formátů. Databáze obrazů u ARCoru pojme tisíc obrazů, ale není problém těchto databází vytvořit více, není tedy omezen počet obrazů. ARCore nemá žádnou webovou službu pro správu databáze. ARCore je pod open source licencí (Apache 2.0 licence) [13]. Velkou výhodou ARCoru je, že je vyvíjen stejnou společností jako operační systém Android a je zde tedy velká

kompatibilita. Jako u Vuforia je možné použít k vývoji Android Studio pro Android, Xcode pro iOS a Unity pro obě platformy s využitím pluginů a rozšíření. ARCore podporuje Android zařízení s minimální verzí 7.0 (API 24).

OpenGL

OpenGL je multiplatformní a open source rozhraní pro počítačovou grafiku, které bylo vydáno roku 1992 [14]. Jedná se o velice rychlé a flexibilní rozhraní, pomocí kterého je možné vykreslit 2D a 3D grafiku, obrázky, různé animace a mnoho dalšího. Jeho rychlost spočívá ve výborné komunikaci mezi centrálním a grafickým procesorem pomocí optimalizace grafické pipeline a možnosti paralelního zpracování procesů [15]. V současné době je nejnovější verzí verze 4.6 z roku 2017 [16].

Na mobilní zařízení existuje speciální verze OpenGL ES. Jedná se o grafické rozhraní OpenGL upravené speciálně pro mobilní zařízení. Celkem jsou tři verze. První verze 1.0 nenabízela příliš velkou flexibilitu, zatímco druhá verze 2.0 nabídla programátorům možnost programovatelné grafické pipeline mezi centrálním a grafickým procesorem. Poslední verzí je verze 3.0, která je nikterak velkou nadstavbou předchozí verze. Tato verze zatím není tolik rozšířena.

2.3.3 ARToolkit

ARToolkit [17] patří k nejstarším a je jedním z nejvíce rozšířených SDK ze seznamu. Jedná se o multiplatformní knihovnu, která podporuje nejenom Android a iOS, ale i Linux, Windows a Mac OS. Podpora platform Android a iOS patřila do knihovny ARToolkitPlus, která byla pod placenou licenci. V dnešní době je již ARToolkit plně přístupný veřejnosti pro všechny platformy pod open source licenci [18]. Stejně jako ARCore využívá ARToolkit k renderování 2D a 3D objektů interface OpenGL. Minimální verzí Androidu pro funkčnost SDK je verze 5.0 (API 21). Poslední verzí je verze 5.4 z roku 2017, kterou lze sehnat z archivního repozitáře na GitHubu [19].

2.3.4 ARToolkitX

ARToolkitX [20] je projekt, který navazuje na předchozí SDK ARToolkit. Projekt byl založen za účelem zajištění dalšího vývoje, údržby a podpory komunity zmiňovaného archivovaného softwaru. ARToolkitX vyžaduje stejnou minimální verzi Android a podporuje stejné formáty souborů. Při rozhodování mezi defakto stejnými knihovnami ARToolkit a ARToolkitX je jistě lepší zvolit tu, která je stále v aktivním vývoji, tedy ARToolkitX.

2.3.5 EasyAR Sence

EasyAR [21] je SDK, které je velice podobné Vuforia. EasyAR také nabízí rozpoznávání, sledování a mapování objektů v reálném čase. EasyAR podporuje stejné platformy jako zmiňovaná Vuforia a nabízí plugin do Unity enginu. Základní bezplatná verze s limitací některých funkcí slouží pouze k nekomerčním účelům. Pro komerční použití je možné zakoupit jednorázově licenční klíč nebo platit měsíční předplatné. Webová služba pro obsluhu databáze objektů je také pod měsíčním předplatným. Stejně jako u Vuforie je předplatné velkou nevýhodou pro toto SDK. EasyAR podporuje Android zařízení s minimální verzí Androidu 4.2 (API 17) [22].

2.3.6 DroidAR

DroidAR [23] je nejstarší framework ze všech zmiňovaných. První prototyp tohoto frameworku vznikl již v roce 2010 a od té doby byl dále vyvíjen. V porovnání se zbylými frameworky podporuje DroidAR pouze Android zařízení s nejnižší minimální verzí, a to verzí 1.6 (API 4) [24]. DroidAR je pod open source licencí pro nekomerční použití. DroidAR nabízí pouze základní nástroje pro rozpoznávání a sledování objektů. DroidAR nenabízí žádnou službu pro správu databáze objektů. Framework podporuje celou řadu formátů modelů, jako je OBJ, G3D, MD2 a COLLADA. Je tedy zřejmé, že použití tohoto frameworku by pokrylo maximální počet Android zařízení, bohužel je již několik let nevyvíjen. Tvůrci pracují na druhé verzi DroidAR 2, ta je ale zatím veřejnosti nepřístupná.

2.3.7 AR.js

Posledním a nejvíce odlišným frameworkem je AR.js [25]. Jak je z názvu patrné, jedná se o JavaScript knihovnu, která nabízí vývojářům vytváření aplikací s prvky rozšířené reality. AR.js nabízí vytvoření webové aplikace, která je funkční jak na zařízeních Android, tak i iOS. Stačí, aby zařízení podporovalo technologii WebGL nebo technologii WebRTC. AR.js je nadstavbou již zmíněného frameworku ARToolKit, knihovny Three.js, která slouží k vykreslování 3D grafiky, a frameworku A-Frame, pomocí kterého lze vytvářet aplikace s prvky virtuální reality. Nevýhodou frameworku je, že obrazy k rozpoznávání musí být ve speciálním formátu a je tedy nutné je přeformátovat. K tomu slouží internetová stránka [26], kam se jednoduše nahraje obrázek a stáhne se požadovaný formát. AR.js podporuje modely ve formátu fTF, OBJ a COLLADA [27].

2.3.8 Shrnutí

Při vybírání nejlepšího frameworku pro tutu práci jsem se zaměřil na to, aby byl co nejméně omezen. Z tohoto důvodu jsem vyřadil Vuforia, která má velice omezený počet rozpoznávání obrazů za měsíc. Stejně tak i EasyAR Sence, který má některé funkce a službu pro správu databáze objektů schované pod měsíčním předplatným. Dalším kritériem bylo, aby byl framework co nejvíce vhodný pro platformu Android. Díky tomu jsem vyřadil framework AR.js, pomocí kterého se vytvářejí webové aplikace. DroidAR by sice dokázal pokrýt nejvíc zařízení, ale bohužel již není v aktivním vývoji a proto jsem ho při rozhodování také vyřadil. Zbyly mi tedy dva frameworky, ARCore a ARToolKitX. Tyto frameworky si jsou velice podobné. Oba podporují nejenom Android, ale i iOS, což by bylo užitečné při dalším pokračování projektu a rozšíření na zmíněnou platformu. ARToolKitX sice podporuje více zařízení, ale při finálním rozhodování jsem dal na přední výhodu knihovny ARCore, a to na maximální kompatibilitu.

3 Návrh aplikací

Mým cílem je vytvořit celkem dvě aplikace. První aplikace bude uživatelská aplikace na platformu Android. Tato aplikace bude využívat prvky rozšířené reality k mapování obrazů do clip rámců. Druhou aplikací bude desktopová aplikace, která bude určena administrátorům pro správu mapovaných obrazů.

3.1 Aplikace na Android

Jak již bylo v úvodu zmíněno, jedná se o aplikaci, která bude primárně sloužit návštěvníkům při různých akcích pořádaných v budově FAV. Využívat jí ovšem bude moct kdokoli, kdo se bude v budově FAV pohybovat. Aplikace je tedy zaměřena primárně na mladé lidi a zaměstnance. Základní vlastností aplikace je rozpoznávání určitých obrazů v clip rámech rozmístěných po budově a následného mapování jiného obrazu, videa či modelu do něj.

3.1.1 Spuštění aplikace

Při spuštění se u aplikace očekává zobrazení úvodní obrazovky, viz obr. 3.1. Pokud se bude jednat o první spuštění, měla by se aplikace uživatele zeptat na všechna potřebná povolení k využití chráněných dat, jako je například přístup ke kontaktům, nebo k využití omezených činností, jako je například nahrávání audia [28]. V našem případě bude určitě potřebné povolení k využití kamery. Během zobrazení úvodní obrazovky by měla aplikace být schopná se aktualizovat. Konkrétně by se mělo jednat o aktualizaci databáze a stáhnutí potřebných souborů, jako jsou například fotky mapovaných obrazů nebo modelů, které se budou do obrazů mapovat. Touto aktualizací se zařídí, že při přidání, upravení nebo odebrání obrazů a jejich modelů, nebude muset vycházet pokaždé nový build aplikace. Druhou variantou řešení je během načítání aktualizovat pouze databázi a obrazy a jejich modely následně stáhnout během používání.



Obrázek 3.1: Návrhy úvodní obrazovky aplikace

3.1.2 Běh aplikace

Po úspěšném načtení aplikace a její aktualizaci by se uživateli měla zobrazit obrazovka s pohledem kamery. Tímto pohledem kamery uživatel může následně skenovat obrazy. Při naskenování obrazu v clip rámu, který bude mít záznam v databázi a pro který bude stažen model, se za běhu aplikace namapuje do zmíněného obrazu přiřazený model. Zobrazovaný model by měl být viditelný pouze při přímém záběru na obraz. Modelem se rozumí jiný obraz, animovaný obraz (gif), video nebo 3D model. Uživatel by měl mít možnost bez jakékoliv interakce s aplikací přejít k jinému obrazu a zobrazit si v něm další model. V případě naskenování více mapovaných obrazů existuje několik možností řešení. Prvním způsobem řešení je zobrazovat modely všech současně rozpoznávaných obrazů. Druhým řešením může být zobrazování modelu, jehož obraz je nejbližší středu obrazovky nebo je například největší. U tohoto problému záleží hodně na vlastnostech zvoleného frameworku. V případě, že model nebude poměrem stran odpovídat obrazu, měl by se model zmenšit tak, aby se vždy do obrazu vešel. Příkladem může být obraz v poměru stran 2:1 a model s poměrem 1:1. Výška modelu se tedy nastaví stejná jako výška

obrazu a šířka se nastaví podle původního poměru stran modelu. Zbyde tedy okolo modelu prázdné místo.

Aplikace by také měla obsahovat určitý informační popis. Jednalo by se o informace, jako jsou například použití aplikace, informace o vývoji aplikace nebo za jakým účelem aplikace vůbec vznikla. Tyto informace by měly být přístupné za běhu aplikace. Ideálním řešením je tlačítko, kterým by si mohl uživatel informace zobrazit. Uživateli by se v tomto případě po stisknutí tlačítka objevil box, viz obr. 3.2 (vlevo), překrývající pohled kamery, ve kterém by byly veškeré informace. Při větším obsahu informací by bylo lepším řešením po stisku tlačítka zobrazit uživateli obrazovku obsahující pouze tyto informace a tlačítko k návratu, viz obr. 3.2 (vpravo).



Obrázek 3.2: Návrhy zobrazení informací o aplikaci

3.1.3 Ukončení aplikace

Pro ukončení aplikace nebude v aplikaci žádné speciální tlačítko. Aplikace půjde ukončit standardním způsobem z navigační lišty.

3.2 Administrátorská aplikace

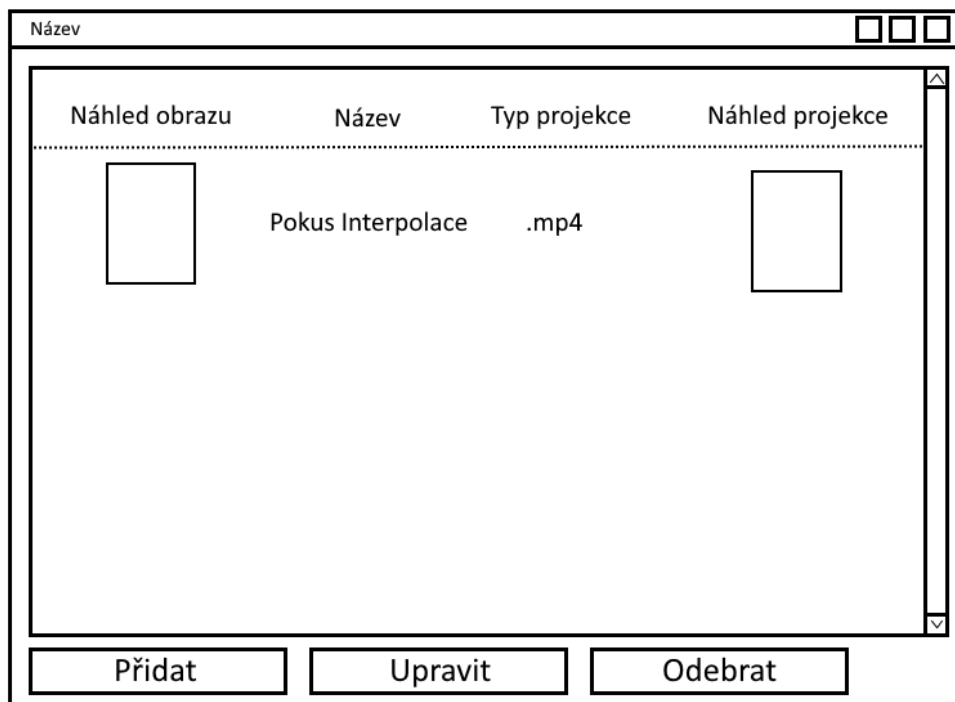
Tato aplikace by měla být přístupná pouze pro administrátory. Aplikaci budou využívat zaměstnanci z různých kateder na Fakultě aplikovaných věd. Takových lidí bude v řádu desítek, možná ani to ne. Aplikace by měla těmto administrátorům umožnit spravovat databázi obrazů a jejich modelů.

3.2.1 Spuštění aplikace

Stejně jako u mobilní aplikace by se měla aplikace aktualizovat a stáhnout si nejnovější data ze serveru. Administrátor by měl být o této události informován.

3.2.2 Běh aplikace

Po aktualizaci všech dat by se administrátorovi mělo zobrazit okno, ve kterém by měl mít možnost prohlížet si všechny záznamy z databáze. Záznam v databázi by sloužil k mapování obrazu a jeho modelu. Dále by záznamy držely další potřebné informace jak pro aplikaci na telefon, tak pro aplikaci pro administrátory. Ideálním způsobem, jak zobrazit uživateli data, je tabulka, viz obr. 3.3. Každá řádka tabulky by představovala jeden záznam v databázi. V tabulce by uživatel viděl náhled obrázku, název záznamu v databázi pro lepší orientaci a typ modelu. Tabulka může obsahovat i další data jako například náhled modelu nebo určitou skupinu, do které by záznam patřil. Podle těchto skupin by se následně mohla tabulka filtrovat, aby měl uživatel větší přehled. V mém případě se bude jednat o dvě skupiny. První skupinou bude pracoviště, které daný záznam přidá, a druhou podlaží, na kterém se mapovaný obraz nachází. Kromě tabulky a jejího filtru by se v okně měla objevit také možnost vytvoření nového záznamu a upravení nebo smazání již existujícího záznamu. Tyto akce lze řešit pomocí tlačítek.



Obrázek 3.3: Návrh vzhledu pro zobrazení záznamů z databáze

Přidání nového záznamu

Po stisknutí tlačítka pro přidání záznamu by se uživateli mělo otevřít nové okno, viz obr. 3.4. Toto okno by mělo obsahovat:

- Textový box pro zadání názvu záznamu. Uživatel by měl mít možnost zadat libovolný název s libovolnými znaky. Tento název poté slouží uživateli pro lepší orientaci.
- ComboBox, v češtině rozbalovací seznam [29], obsahující přidávající pracoviště. Jedná se celkem o šest kateder (KFY, KGM, KIV, KKY, KMA, KME) a jedno nezávislé pracoviště (FAV), které bude sloužit pro děkanát. Uživatel má možnost zvolit jedno ze sedmi zmíněných pracovišť.
- Druhý rozbalovací seznam, který tentokrát nabídne uživateli možnost určit, na jakém podlaží se obraz záznamu nachází.
- Možnost přidání souboru s obrazem. Vhodným řešením může být tlačítko, které uživateli zobrazí okno pro vybrání souboru. Framework ARCore podporuje pro vytvoření databáze obrazů formáty PNG, JPG

a JPEG. Bylo by tedy vhodné přidat do okna s výběrem souboru filtr, který by zobrazoval uživateli pouze zmíněné formáty.

- Možnost přidání souboru s modelem. Velice obdobně jako v předchozím bodě by mohla fungovat volba typu souboru s modelem. Jedinou změnou by byl filtr, který by navíc obsahoval formáty GIF, MP4, AVI a OBJ.
- Tlačítko pro návrat zpět.
- Tlačítko pro vytvoření záznamu.

Okno může dále obsahovat náhledy obrazů a modelů a další komponenty pro lepší přehlednost nebo funkčnost.

Po stisknutí tlačítka pro vytvoření záznamu by se mělo zkontrolovat, jestli uživatel vyplnil všechny požadované údaje, konkrétně že zadal jméno, zvolil pracoviště a patro a přidal soubor s obrazem a modelem. Dále by se mělo pro jistotu zkontrolovat, zdali vybrané soubory jsou vážně ve správném formátu. Pokud je vše v pořádku, nový záznam by se měl nahrát do databáze a soubory by se měly nahrát na server.

The diagram shows a window titled "Název" with standard window controls. The main area contains a "Název:" label followed by a text input field. Below this are two rows of controls. Each row consists of a "Náhled" (preview) box, a "název_souboru" label with a text input field, and a "Vybrat s." (select file) button. At the bottom of the window, there are two buttons: "Návrat" (back) on the left and "Vytvořit/Uložit" (create/save) on the right.

Obrázek 3.4: Návrh vzhledu aplikace pro přidání a upravení záznamu

Upravení záznamu

Pro upravení záznamu lze využít stejné okno jako pro vytvoření záznamu s tím rozdílem, že všechny údaje by byly již předvyplněné existujícím záznamem a namísto tlačítka pro vytvoření záznamu by zde bylo tlačítko pro uložení změn. Po stisku tohoto tlačítka by se opět zkontrolovalo, zdali je vše v pořádku a podle množství úprav se upravila databáze a nahrály se nové soubory na server.

Smazání záznamu

Po stisknutí tlačítka na vymazání záznamu by se měl otevřít dialog pro potvrzení akce, aby se předešlo nechtěným vymazáním záznamů. V případě, že uživatel potvrdí vymazání, záznam se vymaže z databáze a ze serveru se smažou jeho soubory obrazu a modelu.

3.2.3 Ukončení aplikace

Při ukončení aplikace uživatelem by měl program ještě vygenerovat novou databázi obrazů. Uživateli by se mělo otevřít okno, ve kterém bude o této události informován. Po vygenerování databáze obrazů by se všechny okna měla zavřít a program by se měl ukončit. Druhou možností, kdy generovat databázi obrazů je po každém vytvoření, upravení nebo smazání záznamu. V tomto případě by se nemusela databáze obrazů generovat při ukončení aplikace a zároveň by v každý moment byla databáze obrazů na serveru aktuální.

4 Data aplikací

4.1 Umístění dat

K propojení administrátorské a uživatelské aplikace je zapotřebí mít někde uložená data, ke kterým budou mít obě aplikace přístup a které budou aplikace sdílet. Ideálním místem k uložení těchto dat je server popřípadě cloud. Hlavním souborem je databáze prvků, která bude sloužit k mapování obrazu prvku a jeho modelu. Dále bude databáze obsahovat důležitá data jak pro administrátorskou aplikaci, tak uživatelskou. Dalšími soubory jsou samostatné obrazy, jejich modely a databáze obrazů, se kterou pracuje ARCore.

4.1.1 Ares

Prvním řešením je využití služeb serveru Ares. Ares je server Západočeské univerzity v Plzni, který je přístupný všem studentům Katedry informatiky a výpočetní techniky. Server slouží studentům k zpracování jejich semestrálních prací, ale lze ho využít i na práce bakalářské a magisterské. Ares poskytuje velkou řadu služeb a softwaru. V případě chybějícího softwaru je možné kontaktovat KIV support, který má právo instalovat nový software.

Řešení databáze

Na serveru Ares jsou předinstalovány tři systémy na řízení báze dat, zkráceně DBMS z anglického výrazu Database Management System. Jedná se o PostgreSQL, MySQL a Oracle. Ideálním řešením je tedy využití jednoho ze tří zmíněných systémů.

MySQL

Prvním systémem na řízení báze dat na mém seznamu je MySQL. Jedná se o relační systém, který je bezplatný a open source. Jeho základem je jazyk SQL, který rozšiřuje o další funkce [30]. MySQL nabízí velmi jednoduchý a rychlý přístup k datům. MySQL je vhodný na správu malých a nikterak komplexních databází.

PostgreSQL

PostgreSQL je druhý DBMS z mého seznamu. Stejně jako MySQL se jedná o open source systém a je také postavený na jazyku SQL. Oproti MySQL se jedná o objektově relační systém. Ten má stejné vlastnosti jako normální relační systém, ale navíc nabízí více sofistikované datové typy. PostgreSQL nabízí oproti MySQL i více funkcí pro práci s daty [31].

Oracle

Oracle je poslední nabízený systém na řízení báze dat. Oproti MySQL a PostgreSQL se jedná o closed source systém a je tedy nabízený pod placenou licenci. Oracle je též vybudovaný na jazyku SQL a je stejně jako MySQL relačním systémem [32]. Jak Oracle, tak MySQL, spadají pod stejnou společnost Oracle Corporation. Je tedy zřejmé, že Oracle díky placené licenci nabízí více funkcí než zmiňovaný MySQL. V porovnání s PostgreSQL nabízí podobné funkce, ovšem má ze všech třech systémů nejlepší zabezpečení [33].

Umístění souborů

Server Ares umožňuje využití distribuovaného souborového systému k uložení obrazů a modelů. Absolutní cesta k těmto souborům by byla uložena v databázi. Za pomoci této cesty by mohly obě aplikace k souborům přistupovat, stahovat je, měnit nebo mazat.

4.1.2 Google Drive

Google Drive, jak název napovídá, je cloudové úložiště nabízené od společnosti Google a druhé řešení mého problému. Jedná se o službu, díky které může uživatel na virtuální úložiště ukládat soubory. K těmto souborům poté má přístup z jakéhokoliv zařízení s připojením k internetu. Google nabízí pro běžný účet 15 GB úložiště zdarma, větší úložiště je nabízené za měsíční předplatné. V rámci školního účtu je možné vytvořit sdílený disk o velikosti úložiště 5 TB, který by byl ideální k využití.

Řešení databáze

Protože se jedná o cloudové úložiště a nikoli server, není zde možnost využití systému na řízení báze dat. Řešením je tedy ukládat na cloudové úložiště soubor, ze kterého si budeme data brát a bude nám představovat pomyslnou databázi. Soubor by měl být v takovém formátu, aby bylo snadné z něho data číst a zapisovat.

XLSX formát

XLSX je formát souboru od firmy Microsoft používaný v aplikacích Excel od stejné firmy. Vznikl v roce 2007 z jeho předchůdce XLS formátu. Tento formát je velmi známý a celosvětově používaný. XLSX je tabulkový formát, takzvaný spreadsheet. Jedná se o soubory, které jsou formátovány na řádky a sloupce v tabulce. Jeden XLSX soubor může obsahovat více jak jednu tabulku. Pomocí jednotlivých řádek a sloupců můžeme přistupovat k buňkám, které obsahují data. V mém případě by řádky představovaly jednotlivé prvky a sloupce jejich různé atributy. XLSX formát používá formátování XML, které převádí do tabulkové podoby, která je pro uživatele příjemnější na čtení.[34]

CSV formát

Druhým formátem je CSV formát, který se používal již v roce 1972. Jedná se o čistý text, kde jsou hodnoty odděleny čárkami. Každý řádek tohoto souboru představuje nový záznam. CSV je velmi jednoduchý a primitivní formát.[35]

JSON formát

JSON formát je velice podobný formátu XML. JSON vznikl na základě programovacího jazyka JavaScript pro rychlou komunikaci mezi klientem a serverem. První zmínky jsou o něm již v roce 2001. Data v JSON formátu jsou postavené na principu klíče a hodnoty. Klíč si lze představit jako například 'jméno', hodnota by potom byla 'Vašek'. Klíč a jeho hodnota jsou od sebe oddělené dvojtečkou, jednotlivé páry klíč-hodnota poté čárkou. Objekt s různými páry klíč-hodnota začíná složenou závorkou a končí stejným znakem.[36]

Umístění souborů

Soubory pro obrazy a modely by byly uloženy na cloudovém úložišti a jejich názvy by se nacházely v databázi.

4.1.3 Shrnutí

Na začátku práce jsem se rozhodl pro první variantu řešení, tedy využít školní server Ares. Výhodou bylo generování databáze obrazů, která je potřebná pro fungování rozpoznávání obrazů u frameworku ARCore, přímo na

serveru. Již proběhly na mém kontě první pokusy o připojení, nahrání a stažení souborů a vše fungovalo správně. Bohužel jsem nenašel optimální řešení pro to, aby přihlašovací údaje nebyly součástí kódu aplikace. Musel jsem tedy přijít s jiným řešením, jímž bylo využití zmíněného cloudového úložiště Google Drive. Nevýhodou je, že databáze obrazů se musí generovat na zařízeních s administrátorskou aplikací a následně se nahrávat na úložiště. Jako formát databázového souboru na úložišti jsem zvolil XLSX, který je ze všech souborů pro člověka nejpříjemnější pro čtení a vyznání se v datech.

4.2 Knihovny pro XLSX formát

Pro práci se soubory ve formátu XLSX je důležité vybrat vhodnou knihovnu. V mém případě budu potřebovat se soubory provádět pouze základní operace jako je přidávání nových řádek, jejich mazání a upravování. Knihovna musí být implementována pro jazyk Java, protože v něm budou napsány obě aplikace.

4.2.1 Apache POI

Apache POI knihovna je jedna z nejvíce komplexních knihoven pro práci s Excel soubory [37]. Oproti většině ostatních knihoven podporuje jak nový formát souborů XLSX, tak i starý formát XLS. Tato knihovna má velké množství funkcí, jako je vytváření a mazání buněk, upravování jejich stylu nebo klonování [38]. Knihovna je pod open source licencí, konkrétně pod Apache Licence version 2.0 [39].

4.2.2 FastExcel

FastExcel je druhou knihovnou pro jazyk Java. Jak je z názvu patrné, tato knihovna se zaměřuje na rychlost zpracování dat. Při čtení a zapisování velkého objemu dat, bavíme se o desítkách záznamů, dokáže být FastExcel až několikrát rychlejší než předchozí POI knihovna. Zároveň je oproti ní více jednoduchá a nenabízí tolik funkcí. Zajímavou vlastností této knihovny je možnost upravovat dvě tabulky jednoho souboru zároveň v různých vláknech. [40]

4.2.3 DOCKX4J

Knihovna je pod stejnou open source licencí jako Apache POI knihovna. Tato knihovna se zaměřuje primárně jenom na formát XLSX, z kterého jako

jediného dokáže číst. Zapisovat dokáže dále i například do formátů HTML, TXT nebo PDF. Knihovna má všechny potřebné funkce pro upravování buněk. [41]

4.2.4 Shrnutí

Pro svoji práci jsem zvolil knihovnu Apache POI. V porovnání s knihovnou DOCKX4J mi přišla srozumitelnější, hlavně co se čitelnosti kódu týče. Těžké rozhodování bylo poté mezi POI knihovnou a knihovnou FastExcel. Za předpokladu, že by práce obsahovala až desetitisíce záznamů, zvolil bych rychlejší, ale primitivnější knihovnu FastExcel. Ale protože náš Excel soubor bude obsahovat maximálně stovky záznamů, zvolil jsem knihovnu, která nám nabízí více funkcí na úkor rychlosti.

4.3 Rozhraní API

API (Application Programming Interface) je rozhraní, které umožňuje komunikaci mezi dvěma aplikacemi a to včetně práce s daty [42]. I když o nich nevíte, většina aplikací, které každý den používáte, využívají služeb API. Jedná se například o bankovní aplikace, které používají API na komunikaci s bankou, nebo o aplikace pro chytrou domácnost, kde se služby API používají například pro rozsvícení a zhasnutí chytré žárovky. Krásný příklad pro lepší pochopení, co vlastně API je a k čemu slouží, je ukázka s číšníkem.

4.3.1 Příklad s číšníkem

Představte si, že vstoupíte do restaurace a sednete si ke stolu. Během chvíle k vám přistoupí číšník s otázkou, co si objednáte. Poté co číšník přijme vaši objednávku, donese ji do kuchyně, kde se začne objednávka zpracovávat. Jakmile je hotová, číšník vám ji opět přinese [43]. Stejným způsobem funguje i API, které zde hraje roli číšníka. Aplikace bude požadovat určitá data ze serveru, vznesne tedy požadavek API, které vše zařídí a data ze serveru vrátí.

4.3.2 Výhody API

Jednou z největších výhod je zabezpečení systému. Protože rozhraní API slouží jako prostředník, aplikace nevidí přímo do systému a ani s ním nijak nezachází. Stejně je tomu tak u zmíněného zákazníka, který také nevidí do kuchyně a nemůže do přípravy jídla zasahovat. Tímto je chráněna vnitřní

struktura systému. Dále je také volání požadavku API doprovázeno různými autentizačními ověřeními, jako jsou například HTTPS hlavičky nebo soubory cookie. V opačném případě může API zabezpečit i uživatele. Příkladem může být požadavek od serveru o poskytnutí určitých osobních dat. Ten může být uživatelem buď zamítnut nebo povolen [44]. Další velkou výhodou API rozhraní je efektivita a dosah. Pomocí veřejného API můžete snadno sdílet informace, ale zároveň mít pod kontrolou, co vše bude veřejnosti přístupné [45]. Rozhraní API má ještě spoustu dalších výhod, díky kterým je v dnešní době tolik rozšířené.

4.3.3 Google Drive API

V mém případě budu využívat Google Drive API [46] pro práci se soubory na cloudovém úložišti. Pomocí tohoto rozhraní můžu na úložiště Google Drive nahrávat, stahovat, upravovat soubory a mnoho dalšího. Bezpečnost připojení do Google Drive je zařízeno pomocí OAuth autorizačního protokolu. K využití Google Drive API je nutné vytvořit projekt na Google Cloud. V mé práci budu využívat nejnovější verzi rozhraní, kterou je verze 3.

4.3.4 Autorizační protokol OAuth

K povolení přístupu do rozhraní Google API je zapotřebí využít autorizační protokol OAuth. Ten slouží k udělení přístupu do rozhraní. Tento autorizační protokol využívá identifikační údaje [47] k získání přístupového tokenu do rozhraní. Společnost Google nabízí 3 druhy identifikačních údajů:

- API key - API key představuje textový řetězec, který slouží jako klíč do veřejně přístupných dat.
- OAuth client ID - Jedná se o autentizační mechanismus k ověření identity a autorizaci přístupu k osobním datům.
- Service account - Jedná se o speciální druh účtu, který je zaměřený pro použití aplikacemi. Umožňuje provádění akcí nebo přístup k osobním datům jiných účtů. Tomuto účtu lze nastavit přístupová práva.

V mé aplikaci budu potřebovat přistupovat k datům celkem ze dvou aplikací. Pomocí desktopové aplikace budu chtít data nejenom číst, ale i upravovat. Ideálním řešením je tedy využití OAuth client ID autentizačního mechanismu s přihlášením na účet, se kterým budou data sdílena. V případě mobilní aplikace chci, aby šla data pouze číst. Ideálním řešením je vytvoření Service účtu s minimálními přístupovými právy, konkrétně pouze čtení.

Uživatel mobilní aplikace nebude nijak zatěžován přihlašованиеm do žádného Google účtu. Využití API klíče by bylo v obou případech nevhodné, protože by šlo tento klíč snadno získat a šel by použít k neoprávněnému přístupu k datům.

4.4 Struktura dat na Google Drive

Všechna data potřebná pro běh obou aplikací jsou uložena ve složce Virtual-Tour. Tato složka musí být sdílená pro servisní účet, který využívá aplikace na Android zařízení. Stačí, když složka bude sdílená pouze v modu Čtenář, čímž se zaručí, že uživatel bude moci soubory stahovat, ale nikoli nahrávat nebo upravovat. Bez tohoto povolení nemůže aplikace stahovat potřebné soubory z cloudového úložiště, protože nemá přístup ke složce. Tato složka je uložena na sdíleném disku, ke kterému musejí mít administrátoři přístup. V této složce se nachází další tři složky a dva soubory.

4.4.1 Složka `images`

V této složce jsou uloženy všechny obrázky, které se skenují pomocí uživatelské aplikace. Z těchto obrázků se vytváří databáze obrazů pro Google ARCore. Obrazy jsou pojmenované vždy ve formátu `image<číslo prvku v databázi>.<přípona souboru>`. Obrazy mohou být ve formátu `.png`, `.jpg` nebo ve formátu `.jpeg`.

4.4.2 Složka `models`

Jak již název složky napovídá, jedná se o složku s modely, které se zobrazují po naskenování v obrazech ze složky `images`. Modely jsou pojmenované ve formátu `model<číslo prvku v databázi>`. Obraz a model se stejným číslem patří k sobě. Modely mohou být ve formátech stejných jako obrazy, ale navíc ještě ve video formátech `.mp4` a `.avi`, v Gif formátu `.gif` a ve formátu 3D modelu `.obj`.

4.4.3 Složka `textures`

Pokud mapujeme do obrazu 3D model, potřebujeme k tomu samotný model a jeho texturu. A právě textury k 3D modelům jsou uloženy v této složce. Textury mají formát názvu ve tvaru `texture<číslo prvku v databázi>`. Textury jsou ve stejných formátech jako obrazy.

4.4.4 Soubor `image_database.imgdb`

Soubor `image_database.imgdb` obsahuje obrázky ze složky `images` a slouží jako databáze obrázků pro Google ARCore. Soubor je nahráván z administrátorské aplikace, kde se generuje. Uživatelská aplikace na Android si databázi stahuje a pracuje s ní.

4.4.5 Soubor `database.xlsx`

Jak již bylo zmíněno, soubor `database.xlsx` bude představovat databázi prvků. Databáze drží všechny potřebné údaje pro práci s prvky, jejich mazání, vytváření a upravování. Soubor má celkem dvě tabulky.

První tabulka

V první tabulce jsou uložena data k aktivním prvkům.

1. sloupec

V prvním sloupci je uložené id prvku. Každý prvek má své unikátní id, které je generované při vytvoření prvku. Při generování nového id se nejdříve využívají id smazaných prvků. Pokud je ale druhý list se smazanými prvky prázdný, vygeneruje se id o 1 větší než je nejvyšší aktivní id.

2. sloupec

Ve druhém sloupci se nachází název prvku. Název není nijak limitován, takže uživatel může zadat jakýkoliv název si usmyslí. Názvy nemusí být ani unikátní. Slouží pouze pro lepší orientaci administrátorů mezi prvky.

3. sloupec

Ve třetím sloupci je uložena přípona formátu obrazu. Pomocí této přípony lze najít obraz ve složce `images`. Příponu si musíme držet kvůli generování databáze obrázků.

4. sloupec

Čtvrtý sloupec obsahuje datum a čas poslední úpravy obrazu prvku. Slouží v administrátorské aplikaci pro kontrolu aktuálnosti dat, konkrétně obrázků při vytváření databáze obrázků.

5. sloupec

V pátém sloupci je uložen čas a datum poslední úpravy modelu prvku. Tento sloupec je kontrolován při aktualizaci dat na uživatelské aplikaci Android. Pokud nejsou data aktuální, stáhne se do zařízení nový model.

6. sloupec

Šestý sloupec obsahuje číselnou hodnotu od 0 do 3 v závislosti na formátu modelu. Slouží k rychlejší orientaci v uživatelské aplikaci na Android, jakou renderovací třídou má být model vykreslen. Jedná se o:

- 0 - obraz
- 1 - video
- 2 - 3D model
- 3 - gif

7. sloupec

Sedmý sloupec také obsahuje číselnou hodnotu v rozmezí od 0 do 6 a určuje pracoviště, které přidalo prvek do databáze. Upravovat se dá v administrátorské aplikaci a slouží pro lepší orientaci pro uživatele.

- 0 - KFY
- 1 - KGM
- 2 - KIV
- 3 - KKY
- 4 - KMA
- 5 - KME
- 6 - FAV

8. sloupec

Osmý sloupec slouží jako předchozí sloupec pro lepší orientaci pro administrátory a v administrátorské aplikaci mezi prvky. Určuje na jakém poschodí se obraz nachází. Má hodnoty od 0 do 5, kde 0 představuje první podlaží, zatímco 5 představuje poslední šesté poschodí.

9. sloupec

Poslední sloupec obsahuje celé číslo určující přibližný formát obrazu v reálném světě. Údaj se využívá pouze při generování databáze obrazů a slouží k lepšímu rozpoznání a mapování.

- 0 - formát A4
- 1 - formát A3
- 2 - formát A2
- 3 - formát A1
- 4 - formát A0

Příklad

V tabulce 4.1 můžeme vidět příklad záznamu z první tabulky. Jedná se o záznam, který se nachází v přízemí budovy a vložila ho katedra KIV. Obraz je nahrán ve formátu .jpg a v reálném světě má přibližnou velikost formátu A2. Do obrazu je mapováno video.

0	video	.jpg	45019,7613165162	45019,7613165162	1	2	0	2
---	-------	------	------------------	------------------	---	---	---	---

Tabulka 4.1: Ukázka záznamu z první tabulky

Druhá tabulka

Ve druhé tabulce jsou data smazaných prvků. První sloupec je stejný jako v první tabulce a obsahuje id prvku. Druhý sloupec obsahuje čas vymazání prvku. Ten slouží v uživatelské aplikaci na Android pro kontrolu, zdali byl již prvek vymazán nebo nikoli. Ve třetím sloupci je uložena přípona souboru ke smazání, konkrétně přípona obrazu. Příklad se nachází v tabulce 4.2.

4	45032,6245127315	.jpg
---	------------------	------

Tabulka 4.2: Ukázka záznamu z druhé tabulky

5 Google ARCore nástroje

Při využití ARCore SDK pro rozšířenou realitu je součástí knihovny i nástroj zvaný `arccoreimg` od společnosti Google. Jedná se o aplikaci ve formě příkazového řádku, která má dvě důležitá využití. První vlastností je kontrola kvality obrazu a druhou je generování databáze obrazů. V závislosti na zadaných parametrech při spuštění programu se vykoná jedna ze dvou uvedených funkcí.[48]

5.1 Kontrola kvality obrazu

Kontrola obrazu se provádí za pomoci příkazu se dvěma parametry. Prvním parametrem je `eval-img`, který určuje, že se bude jednat o kontrolu kvality obrazu. Druhým parametrem je parametr `--input_image_path=<cesta>`, který určuje cestu kontrolovaného obrazu. Příkaz vrací bodové ohodnocení obrazu ve formě celého čísla v rozsahu od 0 do 100. Je doporučováno využívat obrazy od bodového ohodnocení 75 a výše. V tomto případě ARCore snadněji dokáže obraz rozpoznat a sledovat ho. Lze využít i obrazy s hodnotou 0, ale rozpoznávání a sledování není tak kvalitní jako u lépe ohodnocených obrazů.

5.2 Vytváření databáze obrazů

Databáze se dá vytvářet dvěma způsoby. V obou případech je prvním parametrem `build-db`, který určuje, že se bude jednat o vytváření databáze.

První způsob je jednodušší a stačí k němu pouze složka s obrazy. Zvoláním příkazu s dalšími parametry se vytvoří databáze z obrazů zvolené složky. Druhý parametr ve tvaru `--input_images_directory=<cesta do složky>` určuje cestu do složky s obrazy. Třetím parametrem je parametr `--output_db_path==<cesta nové databáze>`, který určuje, kam a s jakým názvem se vytvoří nová databáze. Každý obraz v databázi bude obsahovat svůj vlastní index, který začíná na nule a je přiřazován podle pořadí souborů ve složce, a název, který je shodný s názvem souboru.

Druhým způsobem se nejdříve musí vygenerovat textový soubor představující list obrazů, které se nahrají do databáze. Jedna řádka zmíněného textového souboru představuje jeden obraz v databázi a má formát `<název obrazu>|<cesta k obrazu>|<šířka obrazu v metrech>`. Stejně jako u předchozího způsobu vytváření databáze i tady obraz obsahuje název

a vlastní unikátní index, který je automaticky generován od první řádky a roste s dalšími. Třetím dobrovolným parametrem řádky je jeho šířka v reálném světě v metrech. Tento parametr slouží poté k lepšímu rozpoznávání a mapování obrazu. To je důvodem, proč společnost Google tento způsob generování databáze obrazů preferuje. S takto vytvořeným seznamem obrazů se pro vytvoření databáze zavolá příkaz s druhým parametrem `--input_image_list_path=<cesta k textovému souboru>` určující textový soubor, podle kterého se bude generovat. Třetí parametr je stejný jako v předchozím příkladu a určuje, kam a s jakým názvem se uloží nová databáze.

Ke generování databáze jsem zvolil druhý způsob. Prvním důvodem je to, že při nahrání shodných obrazů do databáze se při rozpoznávání obrazu vrací obraz s nižším indexem v databázi. A protože v administrátorské aplikaci není nijak ošetřena duplicita obrazů, je vhodným řešením tohoto problému, aby se při duplicitě použil novější model. Před vytvářením databáze tedy generuji textový soubor obrazů od nejnovějších po nejstarší. Druhým důvodem je, že je zde možnost zadání přibližné velikosti obrazu pro lepší rozpoznávání a mapování.

6 Implementace

6.1 Mobilní aplikace

Pro vývoj mobilní aplikace jsem se rozhodoval mezi dvěma vývojovými prostředími. Jednalo se o Unity a Android Studio. Nakonec jsem zvolil Android Studio, se kterým jsem měl ještě před touto prací pár zkušeností. Android Studio nabízí k vývoji aplikace programovací jazyk Kotlin a Javu. Tady jsem zvolil Javu, protože s Kotlinem jsem se nikdy předtím nesetkal. Program je rozdělen do šesti balíčků.

- balíček `activities` - Balík obsahuje pouze jednu hlavní třídu, která představuje jedinou aktivitu v programu.
- balíček `db` - V balíku se nachází třída pro práci s databází telefonu a pomocné třídy typu Enum a databázové modely.
- balíček `fragments` - V tomto balíku se nachází pouze tři třídy představující fragmenty, se kterými aplikace pracuje. Jedná se o fragment pro načítání aplikace, pohled kamery a pro zobrazení informací o aplikaci.
- balíček `helpers` - V balíku se nachází dvě třídy. Jedna slouží pro práci s Excelem, druhá pro práci s rozhraním Google Drive.
- balíček `renderers` - Tento balík obsahuje velké množství tříd, kde všechny slouží k vykreslování objektů pomocí OpenGL nebo jako pomocné třídy.
- balíček `service` - Poslední balík obsahuje třídu servisu, která se stará o stažení dat.

6.2 Aplikace pro administrátory

Tato aplikace byla vytvořena ve vývojovém prostředí IntelliJ IDEA v jazyce Java. Program je strukturovaný celkem do čtyř balíčků pro větší přehlednost.

- balíček `enums` - Balík obsahuje pouze třídy typu Enum. Celkem se zde nachází pět tříd.
- balíček `gui` - Balík obsahuje veškeré třídy grafického rozhraní. Mimo jiné obsahuje i hlavní třídu. Celkem se jedná o čtyři třídy.

- balíček `helper` - V balíku se nachází čtyři třídy, které pomáhají při práci s Excelem, rozhraním Google Drive a při generování databáze obrazů.
- balíček `model` - Balík obsahuje pouze jednu třídu, která představuje model záznamu v databázi.

6.3 Řešené problémy

Ve své práci jsem se během implementace setkal s několika problémy, které bylo potřeba vyřešit.

6.3.1 Přesun z Ares na Google Drive

Prvním již zmíněným problémem byl přechod ze serveru Ares na cloudové úložiště Google Drive. Aby uživatel mohl stahovat, nahrávat a upravovat soubory na serveru, bylo zapotřebí přihlašovací jméno a heslo. Tyto údaje by se dalo snadno sdílet mezi administrátory, ale v mobilní aplikaci by byly přímo obsažené v kódu aplikace. Hrozilo tedy snadné zneužití těchto údajů. Řešením bylo přejít na cloudové úložiště Google Drive, kde se pro různé účty dají nastavovat různá práva. Tím je zabezpečeno, že pouze oprávnění uživatelé mohou upravovat soubory.

6.3.2 Stažení modelů do telefonu

Protože je nutné stáhnout do telefonu velké množství dat v podobě modelů, je zapotřebí, aby stahování neblokovalo vlákno, které se stará o vykreslování uživatelského rozhraní. K tomuto účelu jsem využil speciální komponentu zvanou `Service` [49], která umožňuje provádět operace na pozadí aplikace. Výhodou je, že se operace provádí i při přepnutí na jinou aplikaci. To umožňuje uživateli během stahování trávit čas jinou aplikací. Stahování probíhá v samostatném vlákně uvnitř servisu. S tímto problémem souvisel i další problém, kterým bylo, zdali stahovat modely během úvodní obrazovky nebo již během používání aplikace. Možné byly obě varianty a každá měla své klady i zápory. Při řešení, kdy by se nejdříve stáhly všechny potřebné modely a následně se až mohla aplikace řádně používat, byl nevýhodou dlouhý čekací čas uživatele. Oproti tomu výhodou bylo, že po stažení všech modelů již aplikace fungovala bez problémů. Druhým způsobem řešení by sice uživatel mohl začít používat aplikaci skoro okamžitě, ale při skenování obrazů by se mu zobrazovaly pouze modely, které by již byly stažené. Otázkou je,

zdali by uživatelé nepřestalo používání aplikace bavit. Z toho důvodu jsem se rozhodl veškeré modely nejdříve stáhnout a až poté povolit uživateli použití aplikace jako takové.

6.3.3 Přehrávání videí

Další problém, který jsem musel vyřešit, byl s přehráváním a stopováním videí. Jednalo se o to, že v momentě, kdy se přehrávalo v obrazu video a namířilo se kamerou mimo obraz, video stále běželo, bylo slyšet, ale na obrazovce nebylo vidět. Toto jsem vyřešil tím, že pokud se v pěti po sobě jdoucích snímcích kamery nerozpozná obraz, video se zastaví. Pokud se obraz znovu objeví před kamerou, video pokračuje tam, kde skončilo. Pět po sobě jdoucích snímků jsem nastavil, protože může dojít k tomu, že se u některého snímku obraz nerozpozná. V ten moment by se video zastavilo, zatímco by ale mělo běžet. Tímto způsobem jsem vyřešil i problém se zbylými modely, které se vykreslovaly například přes zeď, pokud se uživatel nacházel v jiném patře nebo místnosti. To protože si ARCore pamatoval poslední pozici modelu a mapoval ho, i když nebyl obraz snímán. Model se tedy zobrazuje jedině v momentě, kdy ARCore rozpoznává jeho obraz.

6.3.4 Názvy modelů

Velký problém nastával při upravení modelu záznamu. Samotné upravení probíhalo v pořádku, záznam se upravil, na Google Drivu se starý model smazal a nahrál se nový. Problém nastával v aplikaci na telefonu. V momentě, kdy se změnil formát modelu, aplikace nedokázala starý model vymazat a ten zůstal uložený v telefonu. Aplikace totiž nejdříve aktualizovala databázi a následně stahovala nestahované nebo změněné modely. Takto byla aplikace připravena na stahování modelů za chodu. Řešením bylo pojmenování souborů s modely bez přípony. Tím se zařídilo, že při změně formátu souboru, název zůstal stále stejný. Pokud aplikace na telefon zjistila, že se model změnil, snadno smazala starý model a stáhla nový. V momentě, kdy se rozhodlo, že aplikace bude modely stahovat během úvodní obrazovky, se změnil i kód aplikace a modely se stahují současně s aktualizací databáze. Nebyl by tedy problém nejdříve smazat starý model a následně záznam v databázi aktualizovat. Již jsem ale nechal modely uložené bez přípony.

6.3.5 Synchronizace

Protože může nastat situace, kdy bude k datům aplikace přistupovat více administrátorů zároveň, bylo důležité vyřešit problém se vzájemnou syn-

chronizací dat. Z tohoto důvodu se před každou úpravou, vytvořením nebo smazáním prvku zkontroluje, zdali data na cloudovém úložišti jsou aktuální. Porovnání aktuálnosti dat se provádí pomocí porovnání poslední úpravy souboru s databází na serveru a na zařízení. V případě, že data skutečně aktuální jsou, upraví se soubor s databází a popřípadě se nahrají nebo smažou související soubory. V opačném případě je o této události administrátor informován dialogem, jeho změny nejsou provedeny a aktualizují se data s cloudovým úložištěm.

6.3.6 Generování databáze obrazů

Posledním problémem, který jsem musel vyřešit bylo generování databáze. V průběhu vývoje aplikace jsem generoval novou databázi pokaždé při vytvoření, upravení nebo smazání záznamu. Bohužel bylo toto řešení ne příliš efektivní. Při větším množství obrazů v databázi se databáze generuje dlouho. Testoval jsem generování databáze, kde bylo sto obrazů a průměrný čas generování trval 13 vteřin. Z tohoto důvodu jsem nechal generování databáze obrazů až při ukončení aplikace.

7 Testování

Již hotové aplikace je nutné otestovat. Protože se jedná o zcela odlišné aplikace s jinou funkcionalitou, každá aplikace se bude testovat jinak.

7.1 Mobilní aplikace

U mobilní aplikace jsem testoval její chování při různých situacích na různých Android zařízeních.

7.1.1 Otočení displeje

Při testování otočení telefonu na šířku jsem přišel na to, že aplikace padala. Nastavil jsem tedy pro celou aplikaci fixní orientaci displeje, protože v našem případě nepotřebujeme, aby uživatel mohl displejem otáčet. Při dalším testování se již aplikace chovala správně, displej se neotáčel a aplikace nepadala na chybě.

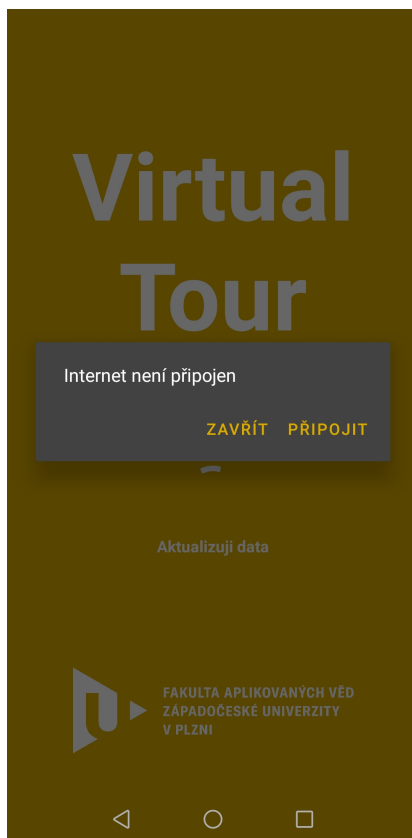
7.1.2 Připojení k internetu

U aplikace je zapotřebí, aby během aktualizace databáze a stahování dat z úložiště byla připojena k internetu. Stahovat data lze buď přes Wifi nebo mobilní data. Zkoušel jsem dohromady několik scénářů na otestování funkčnosti aplikace jak s připojením, tak bez připojení k internetu:

- Spuštění se zapnutou Wifi nebo mobilními daty - Aplikace se v pořádku spustí a stáhne si bez problémů všechna potřebná data.
- Spuštění bez připojení k internetu - Aplikace se spustí a správně zobrazí okno se zprávou o nepřipojení k internetu, viz obr. 7.1. Po následném připojení k internetu aplikace funguje stejně, jako v předchozím případě.
- Vypnutí připojení k internetu během aktualizace dat - Aplikace stejně jako v předchozím případě zobrazí varovné okno, že není připojena k internetu. Po opětovném připojení aplikace pokračuje s aktualizací dat tam, kde skončila.

- Vypnutí připojení k internetu během snímání obrazů - Protože již připojení k internetu není zapotřebí, aplikace nevyhodí žádnou varovnou hlášku a funguje bez problémů.

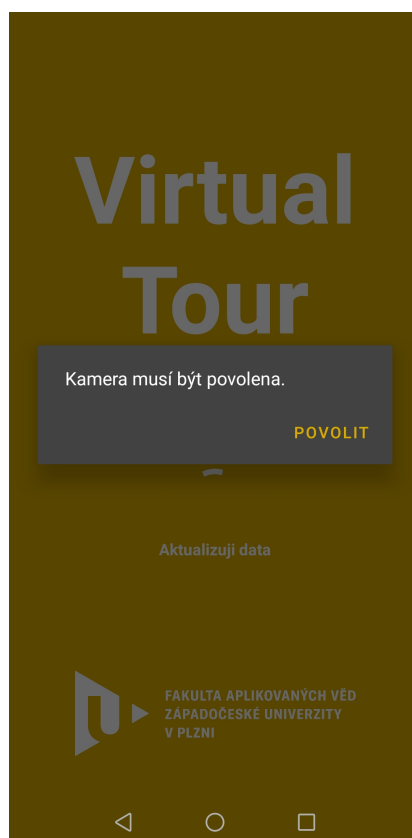
Všechny scénáře proběhly bez problémů.



Obrázek 7.1: Snímek obrazovky s informací o nedostupnosti internetového připojení

7.1.3 Zamítnutí práv ke kameře

Protože aplikace potřebuje přístup ke kameře, je zapotřebí povolení uživatele. Toto povolení se provádí při spuštění aplikace. Pokud uživatel povolení odmítne, aplikace zobrazí okno, viz obr. 7.2, které informuje uživatele, že je kamera potřebná k fungování aplikace. Z tohoto okna může uživatel kameru opět povolit. V případě, že uživatel odebere práva na kameru během chodu aplikace, konkrétně během obrazovky s pohledem kamery a vrátí se zpět, aplikace spadne. Bohužel se mi nepodařilo najít přesnou chybu, na které aplikace padá. V kombinaci s tím, že je tento scénář velice nepravděpodobný, rozhodl jsem se chybu ignorovat. Při dalším spuštění se aplikace správně zeptá na chybějící práva.



Obrázek 7.2: Snímek obrazovky s informací o problému s přístupem ke kamere

7.1.4 Vymazání dat aplikace

V případě, že uživatel k ušetření úložiště na telefonu vymaže data aplikace, stáhne si aplikace při dalším spuštění úspěšně všechna potřebná data.

7.1.5 Uspání aplikace

Aplikaci jsem zkusil uspat během úvodní obrazovky a během pohledu s kamerou. V prvním případě pokračuje stahování dat na pozadí a při vrácení do aplikace se buď bude uživatel nacházet stále na úvodní obrazovce, nebo v případě, že se již všechna potřebná data stáhla, zobrazí se uživateli pohled s kamerou. V druhém případě se uživatel vrátí bez problémů zpět do aplikace.

7.1.6 Vyrušení aplikace

Aplikace může být vyrušena z několika důvodů a je zapotřebí, aby dokázala na takové akce reagovat.

- Budík - V případě vyrušení aplikace budíkem může uživatel za chodu aplikace budík vypnout nebo posunout. Aplikace funguje bez problémů.
- Příchozí hovor - Při příchozím hovoru se zobrazí příslušná obrazovka a aplikace se přerušuje. Po ukončení hovoru je uživatel opět navrácen do aplikace, která v pořádku funguje.
- Příchozí SMS - Po přijetí zprávy se zobrazí upozornění, ze kterého lze na zprávu odpovědět, zobrazit jako přečtenou nebo přejít do aplikace na správu zpráv. Ve všech případech funguje aplikace správně.
- Upozornění od aplikace - Při vyrušení aplikace upozorněním od jiné aplikace se upozornění uživateli zobrazí, zatímco aplikace stále bez problémů funguje.
- Messenger - Při příchozí zprávě od aplikace Messenger se zobrazí okno chatu. S tímto okénkem chatu aplikace dokáže pracovat bez jakýchkoliv problémů a chyb.

7.1.7 Nestažený model

V případě, že aplikace nemá stažený model k obrazu, který je v databázi, aplikace při naskenování tohoto obrazu nepadá a v pořádku se s ní dají skenovat ostatní obrazy, které modely stažené mají.

7.1.8 Testování na různých zařízeních

Aplikaci bylo zapotřebí otestovat i na několika zařízeních, protože každé zařízení se může chovat trochu jinak a aplikace na něm nemusí správně běžet.

Huawei P20 lite

Na tomto telefonu byl prováděn celý vývoj a následné testování všech funkcí. Telefon má verzi Androidu 8.0 s API verzí 26. Jedná se tedy o minimální podporovanou API verzi. Celá aplikace fungovala na telefonu bez problémů.

Huawei P30 lite

Huawei P30 lite spadá do skupiny telefonů od firmy Huawei, kde jsou již omezeny služby od firmy Google. Tímto je způsobeno, že aplikace na tomto telefonu nefunguje, protože pro správný chod frameworku ARCore je zapotřebí mít v telefonu nainstalovanou nejnovější verzi Služeb Google Play. Znamená to tedy, že aplikace nefunguje na všech zařízeních od společnosti Huawei, kde jsou omezeny služby od firmy Google.

Samsung A52

Na tomto telefonu s Android verzí 13 a API verzí 33 fungovala aplikace bez potíží.

7.2 Administrátorská aplikace

Administrátorskou aplikaci jsem testoval převážně opakovaným používáním. Přidával jsem prvky, mazal a upravoval je. Vše jsem průběžně kontroloval s úložištěm Google Drive, zdali se všechna data shodují. Během používání jsem nenarazil na žádnou chybu. Také jsem testoval, co se s aplikací stane v momentě, kdy ztratí připojení k internetu. V tomto případě, pokud zrovna je připojení nutné k chodu aplikace, zobrazí se okno s chybou. Jedná se o situace při stahování dat z cloudového úložiště a jejich nahrávání. Po opětovném připojení aplikace bez problémů dokončí vykonávanou práci. U této aplikace jsem také testoval synchronizaci dat při používání více uživateli zároveň. I v tomto případě fungovala aplikace bez problémů.

7.3 Skenování obrazů

Důležité také bylo otestovat skenování obrazů. Hlavním místem, kde jsem skenování testoval bylo v budově Fakulty aplikovaných věd. Testoval jsem skenování obrazů s různou hodnotou kvality. Aplikace nemá problém s rozpoznáváním obrazu s hodnotou kvality nula. Co se mapování týče, do obrazů s větší kvalitou se objekty mapují s mnohem větší přesností. U obrazů s nízkou hodnotou se chvílemi objekt nemapuje úplně přesně. Problémy s mapováním jsou i ve chvíli, kdy je v místnosti s obrazem velká tma. Folie, které překrývají obrazy, nedělají aplikaci žádné problémy a skenování funguje stejně dobře, jako když obrazy folii na sobě nemají. Při testování jsem také zkoušel skenovat obrazy z různých dálek. Skenování obrazu z velké dálky není funkční. Je to způsobené tím, že obraz musí pokrývat minimálně 25%

obrazu z kamery [50]. Po naskenování ale již mapování objektů jde i z větší dálky. Ovšem čím dál od obrazu uživatel stojí, tím je mapování nepřesnější. Další věcí, kterou jsem testoval, je nastavení jiné přibližné šířky obrazu. Aplikace s tímto nemá problém a i při nastavení úplně jiné přibližné šířky obrazu je skenování obrazu mnohem účinnější, než pokud není přibližná šířka při generování databáze obrazů nastavena. Skenování obrazů i mapování objektů funguje v aplikaci pro naše potřeby správně.

7.4 Shrnutí

Při testování jsem aplikaci otestoval scénáři, během kterých by aplikace mohli mít problémy. U mobilní aplikace jsem řádně otestoval snímání obrazů a mapování objektů a otestoval jsem její funkčnost na několika zařízeních. Obě aplikace fungují tak, jak se očekává a nejsou s nimi žádné problémy.

8 Závěr

Na začátku jsem našel a prostudoval podobné aplikace, ze kterých by se dala brát inspirace. Dále jsem porovnal několik frameworků na základě různých parametrů a zvolil jsem vhodný framework pro naši práci. V další části jsem provedl podrobný návrh obou aplikací a jejich funkcí. Řádně jsem prostudoval všechny nástroje a aplikace, které jsem pro práci potřeboval. Dále jsem provedl implementaci obou aplikací. V poslední části jsem se věnoval testování aplikací a ujistil jsem se, že aplikace nevykazují žádné chybné chování.

Projekt by bylo určitě vhodné rozšířit i na platformu iOS. Další možná vylepšení aplikací vidím v přesunutí dat z cloudového úložiště na server a nahradit tak excelový soubor představující databázi systémem na řízení báze dat. Tímto by se zařídila lepší synchronizace dat mezi uživateli. Pro lepší přehled administrátorů by bylo možné přidat do tabulky se záznamy sloupec s náhledem modelu.

Práce splňuje všechny body zadání a je připravena sloužit jako zpestření prohlídky budovy Fakulty aplikovaných věd.

Literatura

- [1] GOOGLE. *Kernel overview* [online]. Google, 2023/03/24. [cit. 2023/04/10].
Dostupné z: <https://source.android.com/docs/core/architecture/kernel>.
- [2] CURRY, D. *Android Statistics (2023)* [online]. Business of Apps, 2023/02/27. [cit. 2023/04/10]. Dostupné z: <https://www.businessofapps.com/data/android-statistics/>.
- [3] RAPHAEL, J. *Android versions: A living history from 1.0 to 14* [online]. Computerworld, 2023/04/07. [cit. 2023/04/10]. Dostupné z: <https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html>.
- [4] GOOGLE. *Codenames, Tags, and Build Numbers* [online]. Google, 2023/03/20. [cit. 2023/04/10]. Dostupné z: <https://source.android.com/docs/setup/about/build-numbers>.
- [5] *Jarit* [online]. ARLOOPA. [cit. 2023/02/08]. Jarit website. Dostupné z: <https://jarit.app/>.
- [6] AIRCARDS. *What is WebAR and why is it so exciting?* [online]. AirCards, 2020. [cit. 2023/02/08]. WebAR website. Dostupné z: <https://www.aircards.co/blog/what-is-webar>.
- [7] *Artivive* [online]. Artivive. [cit. 2023/02/08]. Artivive website. Dostupné z: <https://artivive.com/>.
- [8] *Google Lens* [online]. Google. [cit. 2023/02/08]. Google Lens website. Dostupné z: <https://lens.google/>.
- [9] *Google Lens* [online]. ARBooksLibrary. [cit. 2023/02/08]. ARBooksLibrary website. Dostupné z: <https://lens.google/>.
- [10] *Getting Started* [online]. Vuforia. [cit. 2023/03/05]. Vuforia developer website. Dostupné z: <https://library.vuforia.com/>.
- [11] *3D Models* [online]. Vuforia. [cit. 2023/03/05]. Vuforia support website. Dostupné z: https://support.ptc.com/help/vuforia/editor/en/index.html#page/vuforia_editor%2Fassets_models.html%23.
- [12] *Overview of ARCore and supported development environments* [online]. Google, 2022/09/15. [cit. 2023/03/05]. ARCore documentation website. Dostupné z: <https://developers.google.com/ar/develop>.

- [13] *LICENSE* [online]. Google. [cit. 2023/03/05]. ARCore license file.
Dostupné z: <https://github.com/google-ar/arcore-android-sdk/blob/master/LICENSE>.
- [14] *Preface: What is OpenGL?* [online]. OpenGLBook.com. [cit. 2023/04/23].
Dostupné z: <https://openglbook.com/chapter-0-preface-what-is-opengl.html>.
- [15] DONZE, D. *5 Reasons to Start Using OpenGL* [online]. Celerity.
[cit. 2023/04/23]. Dostupné z: <https://www.celerity.com/insights/start-using-open-gl>.
- [16] *History of OpenGL* [online]. Khronos, 2022/02/13. [cit. 2023/04/23].
Dostupné z: <https://openglbook.com/chapter-0-preface-what-is-opengl.html>.
- [17] *LICENSE* [online]. HIT Lab. [cit. 2023/03/05]. ARToolKit website.
Dostupné z: <http://www.hitl.washington.edu/artoolkit.html>.
- [18] *ARToolKit* [online]. Wikipedia, 2022/05/12. [cit. 2023/03/05]. Dostupné z: <https://en.wikipedia.org/wiki/ARToolKit>.
- [19] *ARToolKit5* [online]. Github. [cit. 2023/03/05]. ARToolKit repository.
Dostupné z: <https://github.com/artoolkit/ARToolKit5>.
- [20] *ARToolKitX* [online]. ARToolKitX. [cit. 2023/03/05]. ARToolKitX website.
Dostupné z: <http://www.artoolkitx.org/>.
- [21] *EasyAR* [online]. EasyAR. [cit. 2023/03/05]. EasyAR website. Dostupné z: <https://www.easyar.com/>.
- [22] *Platform Requirements* [online]. VisionStar Information Technology.
[cit. 2023/03/05]. Dostupné z: <https://help.easyar.com/EasyAR%20Sense/v4/GettingStarted/Platform-Requirements.html>.
- [23] *DroidAR* [online]. Github. [cit. 2023/03/05]. ARToolKitX repository.
Dostupné z: <https://github.com/bitstars/droidar>.
- [24] *FAQ* [online]. Github. [cit. 2023/03/05]. ARToolKitX wiki. Dostupné z: <https://github.com/bitstars/droidar/wiki/FAQ>.
- [25] *AR.js - Augmented Reality on the Web* [online]. Github. [cit. 2023/03/05].
AR.js website. Dostupné z: <https://ar-js-org.github.io/AR.js-Docs/#:~:text=on%20the%20Web-,AR.,based%20AR%20and%20Marker%20tracking>.

- [26] *AR.js Marker Training* [online]. Github. [cit. 2023/03/05]. AR.js marker website. Dostupné z: <https://jeromeetienne.github.io/AR.js/three.js/examples/marker-training/examples/generator.html>.
- [27] KUTTAPPA, A. *Using 3D models with AR.js and A-Frame* [online]. Medium, 2017/08/20. [cit. 2023/03/05]. Dostupné z: <https://medium.com/@akashkuttappa/using-3d-models-with-ar-js-and-a-frame-84d462efe498>.
- [28] GOOGLE. *Permissions on Android* [online]. Google, 2023/04/06. [cit. 2023/04/11]. Dostupné z: <https://developer.android.com/guide/topics/permissions/overview>.
- [29] GOOGLE. *Permissions on Android* [online]. Google, 2023/04/06. [cit. 2023/04/11]. Dostupné z: <https://office.lasakovi.com/excel/vba-formulare/combobox-prakticky-excel-vba/>.
- [30] TALEND. *What is MySQL? Everything You Need to Know* [online]. Talend. [cit. 2023/01/22]. Dostupné z: <https://www.talend.com/resources/what-is-mysql/>.
- [31] FIVETRAN. *PostgreSQL vs. MySQL: What you need to know* [online]. Fivetran, 2021/09/02. [cit. 2023/01/22]. Dostupné z: <https://www.fivetran.com/blog/postgresql-vs-mysql>.
- [32] WOLFE, M. *MySQL vs Oracle SQL* [online]. Towards Data Science, 2021/08/21. [cit. 2023/01/22]. Dostupné z: <https://towardsdatascience.com/mysql-vs-oracle-sql-a97a7659f992>.
- [33] AYUSHJOSHI599. *Difference between Oracle and PostgreSQL* [online]. GeeksforGeeks. [cit. 2023/01/22]. Dostupné z: <https://www.geeksforgeeks.org/difference-between-oracle-and-postgresql/>.
- [34] [online]. Fileformat. [cit. 2023/02/08]. Dostupné z: <https://docs.fileformat.com/spreadsheet/xlsx/>.
- [35] [online]. Fileformat. [cit. 2023/02/08]. Dostupné z: <https://docs.fileformat.com/spreadsheet/csv/>.
- [36] [online]. Fileformat. [cit. 2023/02/08]. Dostupné z: <https://docs.fileformat.com/web/json>.
- [37] *Working with Microsoft Excel in Java* [online]. Baeldung, 2022/02/02. [cit. 2023/02/18]. Dostupné z: <https://www.baeldung.com/java-microsoft-excel>.

- [38] *POI-HSSF and POI-XSSF/SXSSF - Java API To Access Microsoft Excel Format Files* [online]. The Apache, 2023/02/11. [cit. 2023/02/18].
Dostupné z: <https://poi.apache.org/components/spreadsheet/>.
- [39] *Apache POI - Legal Stuff* [online]. The Apache, 2023/02/11.
[cit. 2023/02/18]. Dostupné z: <https://poi.apache.org/legal.html>.
- [40] [online]. Github. [cit. 2023/02/18]. Dostupné z:
<https://github.com/dhatim/fastexcel>.
- [41] *What is DOCX4J?* [online]. Fileformat. [cit. 2023/02/08]. Dostupné z:
<https://products.fileformat.com/spreadsheet/java/docx4j/>.
- [42] FRYE, M.-K. *What is an API?* [online]. MuleSoft. [cit. 2023/04/12].
Dostupné z:
<https://www.mulesoft.com/resources/api/what-is-an-api>.
- [43] BREWSTER, C. *7 Examples of APIs in Use Today* [online]. Trio.
[cit. 2023/04/12]. Dostupné z:
<https://www.trio.dev/blog/api-examples>.
- [44] IBM. *What is an API (application programming interface)?* [online]. IBM.
[cit. 2023/04/12]. Dostupné z: <https://www.ibm.com/topics/api>.
- [45] SYDLE. *What's an API? Benefits, Examples, and Types* [online]. Sydle,
2022/02/24. [cit. 2023/04/12]. Dostupné z:
<https://www.sydle.com/blog/api-6214f68876950e47761c40e7/>.
- [46] GOOGLE. *Introduction to Google Drive API* [online]. Google, 2023/04/13.
[cit. 2023/04/14]. Dostupné z:
<https://developers.google.com/drive/api/guides/about-sdk>.
- [47] GOOGLE. *Create access credentials* [online]. Google, 2023/04/13.
[cit. 2023/04/14]. Dostupné z: <https://developers.google.com/workspace/guides/create-credentials>.
- [48] GOOGLE. *The arcoring tool* [online]. Google, 2023/03/20.
[cit. 2023/04/05]. Dostupné z: <https://developers.google.com/ar/develop/augmented-images/arcoring>.
- [49] *Services overview* [online]. Google, 2022/08/31. [cit. 2023/04/19].
Dostupné z:
<https://developer.android.com/guide/components/services>.
- [50] GOOGLE. *Add dimension to images* [online]. Google, 2023/03/20.
[cit. 2023/04/16]. Dostupné z:
<https://developers.google.com/ar/develop/augmented-images>.

Uživatelská příručka

Mobilní aplikace

Po zapnutí aplikace musí uživatel vyčkat, než se mu aktualizují veškerá data. Během této aktualizace může uživatel aplikaci opustit, ale nesmí jí vypnout, protože by se aktualizace přerušila. Po aktualizaci se uživateli objeví pohled kamery, viz obr. A.1 (vlevo), pomocí kterého může snímat obrazy. V pravém horním rohu aplikace má uživatel možnost si zobrazit dodatečné informace o aplikaci, viz obr. A.1 (vpravo).



Aplikace slouží k zpestření prohlídky budovy Fakulty aplikovaných věd v Plzni na adrese Technická 8.

Pomocí této aplikace lze snímat obrazy v clip rámech nacházejících se po celé budově.

Projekt vznikl v rámci bakalářské práce a za jeho vývojem stojí Vladimír Holý.



Obrázek A.1: Snímky obrazovky z aplikace Virtual Tour

Administrátorská aplikace

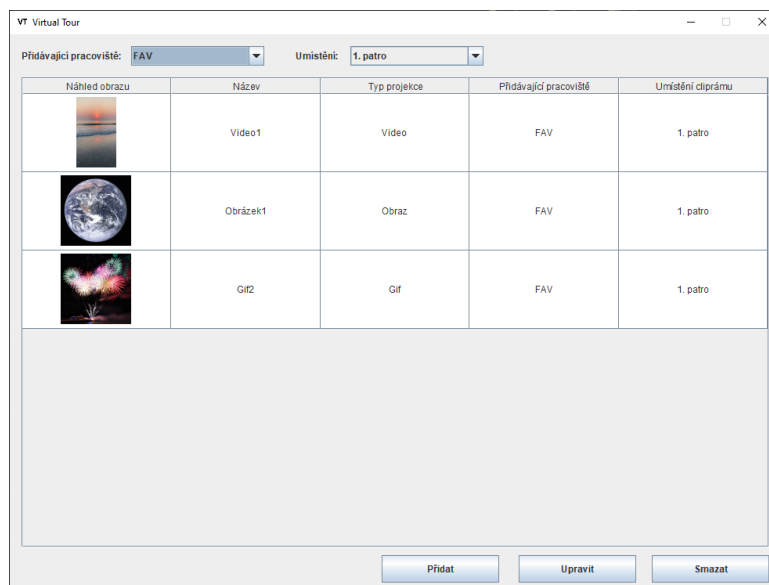
Přidání uživatele




Aby se uživatel mohl stát administrátorem, musí mu správce umožnit přístup do sdíleného disku. Správce musí přidat uživatele do členů sdíleného disku

a nastavit mu práva na Správce obsahu. V tomto momentu již uživatel má přístup ke všem datům a aplikace bude bez problémů fungovat.

Spuštění aplikace

Po spuštění aplikace musí administrátor počkat, než se aktualizují veškerá data. Pokud se jedná o první spuštění, bude uživatel nucen z důvodu vytvoření identifikačních údajů k přihlášení do svého Google účtu, na kterém má již požadovaná práva od správce. Po aktualizaci se administrátorovi zobrazí tabulka se všemi záznamy, viz obr. A.2. Pomocí rozbalovacích seznamů umístěných nad tabulkou může administrátor filtrovat záznamy v tabulce podle přidávajícího pracoviště a patra. Pomocí tlačítek umístěných pod tabulkou může administrátor záznamy vytvářet, mazat nebo upravovat. Pro upravení a mazání je nutné mít vybraný záznam z tabulky pomocí kliknutí na něj.



Náhled obrazu	Název	Typ projekce	Přidávající pracoviště	Umístění cliprámu
	Video1	Video	FAV	1. patro
	Obrázek1	Obraz	FAV	1. patro
	Gif2	Gif	FAV	1. patro

Obrázek A.2: Snímek okna se záznamy

Pro vytvoření a upravení záznamu se administrátorovi otevře speciální okno, viz obr. A.3, kde může upravovat parametry záznamu. Podle kvality obrazu administrátor pozná, jak kvalitně se bude zvolený obraz rozpoznávat a následně jak kvalitně se do něj bude mapovat objekt. Doporučená kvalita od společnosti Google je minimálně 75 [48]. I obrazy s kvalitou 0 lze ale rozpoznat a mapovat. Pro lepší rozpoznání a mapování je důležité zvolit přibližnou velikost obrazu. V případě, že administrátor zadá například velikost A0 v momentě, kdy bude skutečný obraz velikosti A4, rozpoznání a mapování bude stále možné. V případě zvolení modelu ve formátu OBJ

se administrátorovi umožní zvolit texturu k modelu. Textura je povinná. Při vytváření OBJ modelu představuje osa x šířku snímaného obrazu a osa y jeho výšku. Střed os je položen do středu snímaného obrazu. Aby 3D model přesně seděl na obrazu, musí mít model shodné rozměry s obrazem, konkrétně s jeho A formátem zvoleném při vytváření záznamu a nikoli s jeho skutečnými rozměry. Například při zvolení velikosti obrazu A4 musí mít model šířku 210 mm a výšku 297 mm, jinak by buďto přesahoval, nebo by nezaplnil celou plochu obrazu. V případě zvolení obrazu, gifu nebo videa, které nebude přesně sedět s obrazem co se poměru stran týče, se zmenší tak, aby se do obrazu vešlo. Poměr stran zůstane stejný.

The screenshot shows a dialog box titled "VT" with a close button in the top right corner. It contains the following elements:

- Název:** An empty text input field.
- Pracoviště:** A dropdown menu with "FAV" selected.
- Patro:** A dropdown menu with "1. patro" selected.
- Obraz:** A label followed by a "Vybrat obraz" button.
- Kvalita obrazu:** A label.
- Přibližná velikost obrazu:** A section with five radio buttons: "A4" (selected), "A3", "A2", "A1", and "A0".
- Model:** A label followed by a "Vybrat model" button.
- Textura 3D modelu:** A label followed by a "Vybrat texturu" button.
- At the bottom, there are two buttons: "Odejít" and "Vytvořit".

Obrázek A.3: Snímek okna pro vytvoření a upravení záznamu