

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Mobilní editor virtuálních stezek v systému EduARd

David Tyemnyák

Vedoucí: Ing. Ivo Malý, Ph.D.
Obor: Softwarové inženýrství a technologie
Květen 2022

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Tyemnýák** Jméno: **David** Osobní číslo: **465803**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávací katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Mobilní editor virtuálních stezek v systému EduARd

Název bakalářské práce anglicky:

Mobile editor of virtual trails for EduARd system

Pokyny pro vypracování:

Analýzujte strukturu systému EduARd, zejména komunikaci serverové části (backend) a existujícího webového editoru virtuálních stezek. Dále nastudujte mechanismus správy virtuálních stezek a jednotlivé role uživatelů systému. Na základě analýzy navrhnete mobilní editor virtuálních stezek, který bude podporovat vhodné případy užití typické pro mobilní prostředí. Důraz bude kladen na zakotvení jednotlivých bodů stezky v rámci souřadného systému, zakotvení virtuálních objektů pro scény s rozšířenou realitou (AR) a tvorbu multimediálních dat. Aplikaci implementujte pomocí technologie Flutter s využitím vhodných knihoven. Při vývoji postupujte v souladu s metodikou User-Centered Design (UCD) s využitím technik testování bez uživatelů. Aplikaci otestujte pomocí softwarových testů a ověřte její funkčnost na alespoň 2 různých stezkách.

Seznam doporučené literatury:

1. L. Šimon, Mobilní Flutter aplikace pro řešení výukových úloh v systému EduARd, Bakalářská práce, České vysoké učení technické v Praze, 2021.
2. J. Skála, Systém pro správu uživatelů projektu EduARd, Bakalářská práce, České vysoké učení technické v Praze, 2020.
3. Immersal SDK, <https://immersal.com/>
4. S. Aukstakalnis, Practical Augmented Reality, Addison Wesley, 2017.
5. D. Schmalstieg, T. Hollerer, Augmented Reality: Principles and Practice (Usability), Addison Wesley, 2016.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Ivo Malý, Ph.D. katedra počítačové grafiky a interakce FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **06.12.2021**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Ivo Malý, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

_____ Datum převzetí zadání

_____ Podpis studenta

Poděkování

Děkuji Ing. Ivovi Malému, Ph.D. nejen za vedení práce, ale rovněž za velmi věcné rady a konzultace, díky kterým mohla tato práce vůbec vzniknout. Dále děkuji své rodině, přítelkyni a v neposlední řadě také přátelům za jejich dlouhodobou podporu nejen na poli studijním.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně, a že jsem uvedl veškerou použitou literaturu.

V Praze, 20. května 2022

Abstrakt

Mobilní editor virtuálních stezek vzniká zejména pro usnadnění tvůrcům učebnic v systému EduARd, kteří teď mohou upravovat nebo vytvářet učebnice přímo na místě určení a zkvalitnit tak jejich obsah. Cílem práce je vytvořit editor učebnic v systému EduARd, který se zaměřuje na práci s multimediálním obsahem, zejména pak vyzkoušet propojení s rozšířenou realitou. Důležité je zapojení do celého ekosystému již existujících aplikací. Po uvedení do problematiky následuje analýza stávajícího systému a jeho dílčích částí, existujících aplikací a datové struktury systému. Na základě analýzy je pak navrženo řešení dle metodiky User-Centered Design. Návrh probíhá v souznění s již existujícím řešením webového editoru. Návrh následuje implementace mobilní aplikace ve frameworku Flutter ve spojení s vloženou aplikací v Unity pro prohlížení a ukotvování 3D objektů. Vše završuje automatizované a uživatelské testování, které ověří správnou funkcionalitu a uživatelskou přívětivost řešení.

Klíčová slova: Systém EduARd, mobilní aplikace, Flutter, Unity, Immersal

Vedoucí: Ing. Ivo Malý, Ph.D.
Katedra počítačové grafiky a interakce
FEL

Abstract

The mobile virtual path editor is created mainly to facilitate the creators of textbooks in the EduARd system, who can now edit or create books directly at the destination and thus improve their content. The work aims to create a textbook editor in the EduARd system, which focuses on working with multimedia content, especially to test the connection with augmented reality. It is essential to involve existing applications in the whole ecosystem. After the introduction, the analysis of the current system and its parts, existing applications, and data structure of the system follows. A solution according to the User-Centered Design methodology is then proposed based on the analysis. The design is in line with the existing web editor solution. The proposal follows implementing a mobile application in the Flutter framework in conjunction with an embedded application in Unity for viewing and anchoring 3D objects. Everything is completed by automated user testing, which verifies the correct functionality and user-friendliness of the solution.

Keywords: System EduARd, Mobile applications, Flutter, Unity, Immersal

Title translation: Mobile editor of virtual trails for EduARd system

Obsah

1 Úvod	1	2.3.1 SignIn response (JSON)	8
1.1 Představení	1	2.3.2 BookList response (JSON)	9
1.2 Motivace	2	2.3.3 BookDetail response (JSON)	10
1.3 Cíle práce	2	2.3.4 Assets response (JSON)	10
1.3.1 Podporované platformy	3	2.3.5 QuestionSet response (XML)	10
1.3.2 Uživatelské rozhraní	3	2.3.6 Task (XML)	11
1.3.3 Práce s audiovizuálním obsahem	3	2.3.7 Questionslide (XML)	12
1.3.4 Práce s rozšířenou realitou (AR)	3	2.3.8 Question (XML)	12
1.3.5 Menší úpravy textů a úkolů	4	2.4 Systémové požadavky	13
1.3.6 Kompatibilita editorů	4	2.4.1 Funkční požadavky	13
2 Analýza	5	2.4.2 Nefunkční požadavky	14
2.1 Stávající systém	5	2.5 Propojení s AR	14
2.2 Aplikační rozhraní	7	2.5.1 Wikitude	15
2.2.1 JSON formát	7	2.5.2 Immersal SDK	15
2.2.2 XML formát	8	3 Návrh řešení	17
2.3 Datová struktura	8	3.1 Flow obrazovek	17
		3.2 Návrh wireframe	18
		3.2.1 Přihlášení	19

3.2.2 Seznam učebnic	20	5.2.2 Testovací scénáře	35
3.2.3 Detail učebnice	21	5.2.3 Zjištěné nálezy z testování . .	38
3.2.4 Vytvoření a editace učebnice	22	6 Závěr	39
3.2.5 Editace úkolu v učebnici	23	6.1 Splněné cíle	39
3.2.6 Editace obrazovky úkolu	24	6.2 Rozvoj aplikace do budoucna . . .	40
3.2.7 Přidání objektu do AR scény	24	A Literatura	41
4 Implementace	27	B Návod pro testování	45
4.1 Flutter aplikace	27		
4.1.1 Architektura aplikace	28		
4.1.2 Použité rozšiřující balíčky . . .	29		
4.2 Unity aplikace	30		
4.3 Flutter - Unity bridge	31		
5 Testování aplikace	33		
5.1 Unit testy	33		
5.1.1 Pokrytí aplikace	33		
5.2 Uživatelské testování	34		
5.2.1 Participanti testování	34		

Obrázky a ukázky kódu

2.1 Schéma fungování systému EduARd.....	6
2.2 SignIn response	9
2.3 BookList response	9
2.4 Assets response	10
2.5 QuestionSet entita	11
2.6 Task entita	11
2.7 Questionslide entita	12
2.8 Question entita	12
3.1 Flow obrazovek.....	18
3.2 Přihlašovací obrazovka	19
3.3 Obrazovka se seznamem učebnic	20
3.4 Obrazovka detailu učebnice	21
3.5 Obrazovky pro vytváření a editaci učebnice	23
3.6 Obrazovky pro vytváření a editaci učebnice - detailu úkolu	23
3.7 Obrazovky pro vytváření a editaci učebnice - detail obrazovky	24

3.8 Obrazovky pro vytváření a editaci učebnice - úprava pozice 3D objektu ve scéně	25
4.1 Komunikace z Flutter aplikace do Unity	32
B.1 Obecné informace o učebnici ..	46
B.2 Úvodní obrazovky učebnice ...	47
B.3 Mapování scény	48
B.4 Vytvoření nové mapy	49
B.5 Přehled všech mých map	50
B.6 Umístění modelu do AR prostoru	51

Tabulky

5.1 Odhalené chyby z uživatelského testování.....	38
---	----

Kapitola 1

Úvod

Spolu s rozmachem a dostupností levných mobilních zařízení s nepřebornou funkcionalitou vyvstala možnost jejich využití nejen pro účely komunikace, práce a zábavy, ale rovněž také pro podporu vzdělávání. Ovládání mobilních zařízení se zdá býti natolik jednoduché, že většinou nedělá problém ani menším dětem. Tyto aspekty přispěly k tomu, že na půdě ČVUT, konkrétně na Fakultě elektrotechnické, díky znalostem a dlouhodobým zkušenostem v oblasti vývoje software začal před několika lety vývoj sady učebních nástrojů obecně známých pod názvem „EduARd“ [1].

1.1 Představení

Název v sobě skrývá slovní hříčku v podobě spojení slov „Education“ a známé technologické zkratky „AR“ (Augmented Reality) - rozšířená realita. Systém EduARd umožňuje vytvářet virtuální stezky v přírodě nebo ve městech, které je možné procházet za využití rozličných senzorů, vstupů a výstupů mobilních zařízení. Zejména GPS (Global Position System), přehrávání audia a videa, zobrazování fotografií a obrázků. Pro každé stanoviště je možné přidat úkoly různého zaměření - od klasických kvízů po porovnávání obrázků.

V době tvorby práce již existuje několik součástí, respektive dílčích aplikací systému EduARd - konkrétně se jedná o hlavní opěrný bod, kterým je aplikační server [2] vystavující API (Aplikační rozhraní) pro přístup jednotlivých klientů. Aplikační server zajišťuje manipulaci dat a propojení s databází, přihlašování

uživatelů, získávání potřebných médií a mnohé další. Mezi uživatele, které server obsluhuje řadíme nejen žáky samotné, ale i učitele a administrátory spravující obsah učebnic. Kromě webových verzí, mezi které patří primárně editor [3] existuje také nepřeberné množství mobilních aplikací. Konkrétně se jedná o prohlížečku 3D scén [4], aplikace pro průchod stezkou, např. ve frameworku Flutter [5], v jazyce Java [6] nebo ve frameworku React Native [7] apod. Další informace a bližší popis je dostupný na webu¹ nebo v následujících kapitolách.

1.2 Motivace

Jak již bylo řečeno, EduARd již má svou webovou verzi a hned několik mobilních aplikací. Jedná se primárně o aplikace postavené na procesech průchodu stezkou uživateli. Editor stezek jako takový existuje pouze jako webová verze. Pro většinu případů užití je webový editor zcela postačující, ale v poslední době spolu s rostoucími požadavky a možnostmi mobilních platforem se nabízí možnost vytvořit editor stezek jakožto mobilní aplikaci.

Mobilní aplikace přináší značné benefity v situacích, kdy si tvůrce stezky chce ověřit na místě, zda jsou všechny úkoly dobře rozmístěné, přidaná média aktuální a 3D objekty správně ukotvené. Takové úkony se z webové aplikace vykonávají poměrně obtížně a ne zcela přesně. Mobilní zařízení naopak není příliš vhodné pro psaní delších souvislých textů, kde jej nahradí pohodlnější psaní na klávesnici.

1.3 Cíle práce

Pro potřeby úpravy učebnic samotných, ale zejména jejich audiovizuálního obsahu přímo na daném místě při průchodu stezkou, je potřeba konkrétně zaměřit dané scény, umístit nové fotografie apod. Tyto úkony se především hodí pro mobilní zařízení, které je vybaveno hned několika druhy senzorů - jmenovitě například gyroskop, akcelerometr a další [8].

¹Bližší popis systému dostupný z <https://fel.cvut.cz/cz/vz/produkty/eduard>

■ 1.3.1 Podporované platformy

Je potřeba multiplatformní mobilní aplikace pro vytváření, úpravy a mazání virtuálních stezek v systému EduARd. V tomto případě rozumíme slovem multiplatformní dvě klíčové mobilní platformy, kterými jsou operačním systémem iOS společnosti Apple a operační systém Android společnosti Google.

■ 1.3.2 Uživatelské rozhraní

Velký důraz je kladen na snadno použitelné uživatelské rozhraní. Mobilní zařízení disponují oproti počítačům omezenou velikostí pracovní plochy a možnostmi interakcí s aplikací samotnou. Neméně důležitou částí je práce s rozšířenou realitou a dalším audiovizuálním obsahem, která bude upřednostněná oproti práci s texty. Spolu s těmito požadavky ale platí poučka, že pokud již existuje podobná aplikace, není dobré uživatele učit nové postupy a zvyklosti, které by byly v rozporu se stávajícím řešením.

■ 1.3.3 Práce s audiovizuálním obsahem

Mobilní zařízení díky dostupnému hardware nabízí možnosti pořizování fotografií, videa či zvukových záznamů. Takové druhy médií jsou důležitými podpůrnými pilíři pro jednotlivá zastavení na virtuálních stezkách. Konkrétně se jedná o nahrávání pořízeného audiovizuálního obsahu, jeho přiřazování k jednotlivým učebnicím, potažmo jednotlivým úkolům. Dále pak mazání nebo případná výměna obsahu.

■ 1.3.4 Práce s rozšířenou realitou (AR)

Rozšířená realita sice může být zahrnována mezi audiovizuální obsah, ale má zde své vlastní umístění. Důvodem je zcela odlišná práce s tímto typem média. Aplikace by měla umožňovat ukotvení 3D modelů do prostoru a jejich prohlížení v prostoru. Aby byla pro uživatele práce pohodlná a jednoduchá, měla by být možnost prohlížení a ukotvování objektů neoddelitelnou součástí editační aplikace. To zároveň usnadňuje budoucí vývoj, jelikož není potřeba spravovat více aplikací najednou.

■ 1.3.5 Menší úpravy textů a úkolů

Mobilní editor by měl nabízet možnost upravit texty obsažené v jednotlivých učebnicích. Vzhledem k omezením mobilních zařízení na tyto možnosti není kladen takový důraz. Během průchodu stezkou bude její autor převážně tvořit audiovizuální obsah. Co se textů týče, zde půjde patrně o kratší poznámky, které poté doplní ve webovém editoru.

■ 1.3.6 Kompatibilita editorů

V neposlední řadě je potřeba také dodržet zpětnou kompatibilitu mobilního editoru s editorem webovým a dalšími mobilními aplikacemi. To by mělo být zajištěno dodržáním správné struktury ukládaných a čtených dat dle specifikace aplikačního rozhraní a validací povinných polí ve formulářích.

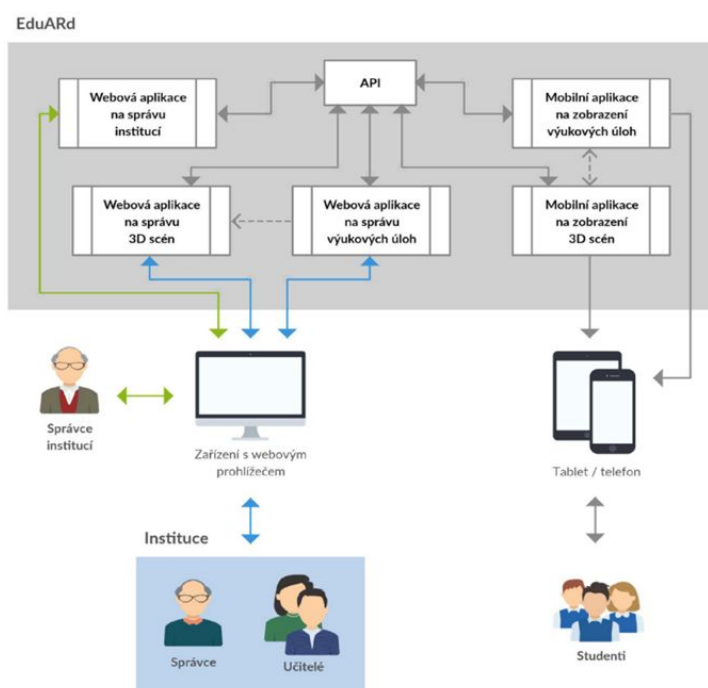
Kapitola 2

Analýza

Proces tvorby aplikace započne analýzou stávajícího řešení - zejména webového editoru [3], jakožto předchůdce a výchozího řešení pro tvorbu a úpravu učebnic. Dále je třeba nastudovat dokumentaci API [9], zjistit jakou strukturu dat lze očekávat v rámci jednotlivých koncových bodů, a jak s nimi pracovat [2]. Vše bude probíhat se zaměřením na splnění klíčových požadavků, které jsou vybrány ze slovního zadání v úvodní kapitole a blíže rozepsány v sekci Systémové požadavky. Dále pak flow obrazovek aplikace (přechod a navigaci mezi nimi), aby na sebe jednotlivé funkční části logicky navazovaly. Následně je potřeba navrhnout pomocí tzv. „wireframes“ [10] rozmístění interakčních prvků a dalších důležitých elementů rozhraní.

2.1 Stávající systém

Systém EduARd zahrnuje vícero samostatných aplikací operujících v režimu klient, nebo server. Rovněž z hlediska funkcionality se aplikace dělí na několik rozdílných. Existují verze úzce zaměřené na jednotlivé platformy, ale rovněž aplikace webové. Pro lepší orientaci uvádím jednoduché schéma (Obrázek 2.1) převzaté z bakalářské práce Anastasie Surikove [3] a níže pak i jejich seznam.



Obrázek 2.1: Schéma fungování systému EduARd [3]

1. Klient

- a. **Editor učebnic** - jak z názvu vypovídá, slouží tvůrcům učebnic pro vytváření, rychlý náhled a úpravy učebnic. Napsaná ve frameworku React [3]
- b. **Prohlížeč učebnic** - Prohlížeček učebnic existuje hned několik, jsou napsané ve frameworku Flutter [5] nebo v React Native [7] a historicky taktéž v Javě [6]
- c. **Prohlížečka 3D modelů** - Slouží pro prohlížení 3D modelů v X3D, napsaná v Unity [4]
- d. **Web AR** - Aplikace řešící správu 3D modelů, je součástí webového editoru, napsaná v javascriptu [11]
- e. **Web administration** - Aplikace pro správu institucí, napsaná ve Vue.js jež byla součástí aplikačního serveru [2]

2. Server

- a. **Aplikační server** - Server poskytující API rozhraní, napsaný v ASP.Net [2]
- b. **Marketplace server** - Pomocný server, který obsluhuje Marketplace, napsaný opět v ASP.Net [12]

V rámci prací na mobilním editoru je více než vhodné prozkoumat stávající aplikace, aby nová příliš nevybočovala ze zavedeného standardu, na který

již uživatelé byli naučeni, a který již znají. Tomuto postupu se v návrhu uživatelského rozhraní říká „Strive for consistency“, doslovně „Snažte se o konzistenci“. Pokud by aplikace fungovala jinak než doposud, museli by se uživatelé učit nové postupy, což by vedlo ke zbytečným a nežádoucím zmatkům.

■ 2.2 Aplikační rozhraní

Vzhledem k tomu, že aplikační rozhraní (dále zkráceně API) je již navrženo, nebylo v současné době možné příliš měnit jeho strukturu. Již běží aplikace, které jsou na tuto datovou strukturu napojené, a jejich funkcionality by se tímto mohla porušit. Strukturu API je možné nastudovat v dokumentaci dostupné online na stránce api.eduard.fel.cvut.cz. Dle této dokumentace bylo zjištěno, že jednodušší informace o knihách jsou navraceny ve formátu JavaScript Object Notation (dále zkratka JSON) [13], ty složitější pak ve formátu Extensible Markup Language (dále zkratka XML) [14]. Jeho struktura se od struktury JSON liší hlavně zaobalením jednotlivých entit do značek podobných struktuře HTML dokumentů. Oba formáty mají své výhody, avšak v dnešní době se pro přístup Representational state transfer (dále zkratka REST) využívá spíše struktury dat ve formátu JSON. V poslední době pak kromě REST rozhraní začíná být stále více populární rozhraní GraphQL, které ovšem není předmětem této práce.

■ 2.2.1 JSON formát

Základní kostru aplikace poskytuje formát JSON, který je z obou přístupů používanější a současně pracovně přívětivější. Jmenovitě zajišťuje systém autentizace a autorizace uživatelů, sadu operací Create, Read, Update a Delete (dále zkráceně CRUD) pro jednotlivé audiovizuální soubory, sadu operací CRUD pro jednotlivé 3D scény a základní strukturu učebnic. Mimo tyto důležité sekce obsahuje JSON API ještě další součásti, které v současné době nejsou pro editor využívány.

■ 2.2.2 XML formát

Pro potřeby detailu jednotlivých učebnic existuje struktura ve formátu XML, která zajišťuje strukturu dokumentu učebnice, kde má každá sekce svou část. Učebnice je členěna do takzvaných „tasků“ lokalizovaných pomocí GPS souřadnic, tyto tasky v sobě obsahují otázky, které v sobě obsahují různé druhy médií. Speciální úlohu zde hraje task s atributem `name="intro"`, který v sobě nese sadu úvodních obrazovek - těch, které jsou zobrazené ještě před započítím plnění úkolů v učebnici. Pro tvorbu nebo úpravu učebnice je zapotřebí nahrát celou strukturu XML dokumentu. Bližší rozepsání entit je zmíněno v následující sekci.

■ 2.3 Datová struktura

Bližší popis jednotlivých entit, se kterými pracuje aplikační server, je rovněž jako dokumentace API koncových bodů (endpointů) dostupný z webu api.eduard.fel.cvut.cz, ve spodní části pod sekcí jednotlivých skupin koncových bodů (endpointů). Aby bylo v budoucnu možné strukturu dat jednoduše modifikovat, zmíním zde pro úplnost klíčové entity v době tvorby práce.

■ 2.3.1 SignIn response (JSON)

Pokud dojde k úspěšnému ověření uživatele, vrátí se z koncového bodu `accessToken` a `refreshToken`, jak je patrné z ukázky Kód 2.2. `accessToken` slouží, jak již z názvu vypovídá, k přístupu k jednotlivým datům za přihlášením a `refreshToken` slouží k obnovení `accessToken` v případě jeho vypršení. V případě neúspěšného přihlášení navrátí server status 401 a aplikace by se podle něj měla správně zachovat - například zobrazit uživateli chybovou hlášku.

```

{
  "accessToken": {
    "token": "string",
    "expiresIn": 0,
    "claims": [
      ... pro tuto ukázkou podstatné
    ],
    "jti": "string"
  },
  "refreshToken": "string"
}

```

Kód 2.2: SignIn response

■ 2.3.2 BookList response (JSON)

Po úspěšném přihlášení již můžeme vykonávat i další požadavky. Seznam všech knih (Kód 2.3) je vhodný pro zobrazení seznamu obdobně jako je tomu ve webovém editoru. Zde za zmínku stojí hlavně fakt, že atribut „isPublished“ je sice z aplikačního serveru navracen, avšak nevrací správnou hodnotu - správná informace o publikování knihy je dostupná až v jejím detailu ve struktuře XML. Dále pak atributy „description“ a „language“ nemusí být vždy totožné s jejich protějškem ve struktuře XML.

```

[
  {
    "id": 0,
    "name": "string",
    "description": "string",
    "numberOfAttempts": 0,
    "numberOfTasks": 0,
    "mD5": "string",
    "createdAt": "2022-05-08T09:03:13.262Z",
    "updatedAt": "2022-05-08T09:03:13.262Z",
    "state": 0,
    "isPublished": true,
    "assetThumbNailId": 0,
    "image": "string",
    "language": "string"
  }
]

```

Kód 2.3: BookList response

■ 2.3.3 BookDetail response (JSON)

Tento se dle mých průzkumů nikterak neliší od struktury dat vracené pro všechny knihy, nepokládám proto příliš velkou důležitost jeho popisu.

■ 2.3.4 Assets response (JSON)

Pro práci s audiovizuálním obsahem knihy slouží právě tyto takzvané „Assets“, struktura je co možná nejjednodušší. Jak je vidět v ukázce Kód 2.4, jedná se o různé druhy médií - od obrázků až po videa. Kromě samotného zobrazení je možné jednotlivá média přiřazovat k jednotlivým obrazovkám a úkolům, dále pak také jako úvodní obrázek učebnice. Jednotlivé multimediální soubory lze rovněž nahrávat a mazat.

```
[
  {
    "id": 0,
    "bookId": 0,
    "name": "string",
    "fetchUrl": "string"
  }
]
```

Kód 2.4: Assets response

■ 2.3.5 QuestionSet response (XML)

Detailní popis knihy s jednotlivými úkoly a obrazovkami (Kód 2.5). Struktura XML umožňuje pracovat s knihou jako celkem a uložit ji formou nahráním XML souboru. Učebnice je uvozena tagy „questionset“, což je definice učebnice samotné. Tento tag může mít další atributy, jimiž mohou být například „language“ a „description“, jak již bylo zmíněno výše. Popis dalších entit obsahují následující sekce.

```

<questionset>
  <task>
    <location>
      <latitude/>
      <longitude/>
    </location>
    <questionslide>
      <questions>
        <question/>
        <description/>
        <images/>
        <audio/>
        <video/>
      </questions>
    </questionslide>
  </task>
</questionset>

```

Kód 2.5: QuestionSet entita

■ 2.3.6 Task (XML)

Task slouží jako popis skupiny obrazovek. Jeho popis demonstruje Kód 2.6. Každý „task“ má svou GPS lokaci uvozenou do tagu „location“. Úvodní obrazovky jsou pak, jak již bylo v úvodu zmíněno, označeny názvem „intro“. Každý task pak v sobě obsahuje 0 – n entit typu „questionslide“, které v sobě noří jednotlivé otázky.

```

<task>
  <location>
    <latitude/>
    <longitude/>
  </location>
  <questionslide>
    ... další obsah vynechán
  </questionslide>
</task>

```

Kód 2.6: Task entita

2.3.7 Questionslide (XML)

Tato entita (Kód 2.7) představuje v rámci prohlížečí aplikace jednu obrazovku, která může obsahovat různé typy otázek - od obyčejné textové otázky až po multimediální. Atributy objektu mohou nabývat hodnot jako je jméno, titulek a druh zobrazení.

```
<questionslide>
  <questions>
    <question/>
    <description/>
    <images/>
    <audio/>
    <video/>
  </questions>
</questionslide>
```

Kód 2.7: Questionslide entita

2.3.8 Question (XML)

Entita představující otázku samotnou, jejíž strukturu ukazuje Kód 2.8. Může se jednat o různé typy otázek, jak již bylo zmíněno v předchozí sekci. Question v sobě nese atributy jako order (pořadí), typ otázky, zda by se měla otázka validovat oproti možnostem a atribut shuffle, který udává, zda se má pořadí možných odpovědí náhodně promíchat.

```
<question>
  <description/>
  <variant valid="true">
    cervena
  </variant>
  <variant valid="true">
    zelena
  </variant>
  <variant valid="false">
    zluta
  </variant>
</question>
```

Kód 2.8: Question entita

2.4 Systémové požadavky

Nejen ze samotného zadání práce, ale rovněž z prvotních konzultací vznikl slovní popis zadání, ze kterého je možné identifikovat pro tvorbu aplikace tolik důležité systémové požadavky. Tyto požadavky se pak dále dělí na požadavky nezbytné pro funkčnost systému - typicky požadavky typu „systém umožní uživateli upravit strukturu učebnice“ a požadavky neurčující faktickou funkčnost systému - typicky požadavky typu „aplikace bude dobře rozšiřitelná a udržitelná do budoucna“.

2.4.1 Funkční požadavky

1. **Přihlášení do aplikace** - Aplikace bude umožňovat uživateli se přihlásit pomocí uživatelského jména a hesla
2. **Zobrazení všech dostupných knih** - Aplikace bude umožňovat uživateli zobrazit všechny dostupné knihy ve stavu publikované i nepublikované
3. **CRUD operace pro knihu** - Aplikace bude umožňovat uživateli vytvořit knihu, zobrazit detail knihy, upravit obsah knihy a smazat knihu
4. **CRUD operace pro obrazovky v knize** - Aplikace bude umožňovat uživateli vytvořit, zobrazit, upravit a smazat jednotlivé obrazovky v knize
5. **CRUD operace pro úkoly v knize** - Aplikace bude umožňovat uživateli vytvořit, zobrazit, upravit a smazat jednotlivé úkoly a jejich otázky v knize
6. **Práce s audiovizuálním obsahem** - Aplikace bude umožňovat nahrávání audiovizuálního obsahu a tento obsah posléze přiřazovat k jednotlivým úkolům a obrazovkám. Aplikace bude rovněž umožňovat audiovizuální obsah mazat
7. **Propojení s AR** - Aplikace bude umožňovat propojení s rozšířenou realitou, jakožto i ukotvení objektů do scény

2.4.2 Nefunkční požadavky

1. **Rozšiřitelnost** - Aplikace bude psána tak, aby byla do budoucna snadno rozšiřitelná
2. **Uživatelské prostředí** - Uživatelské prostředí bude snadno ovladatelné, přizpůsobené cílové skupině uživatelů
3. **Požadované platformy** - Aplikace bude fungovat na operačním systému Android a na operačním systému iOS
4. **Výkon a stabilita** - Aplikace bude dostatečně výkonná, aby se nezasekávala a nebude padat

2.5 Propojení s AR

Jedním z důležitých požadavků je propojení s rozšířenou realitou, konkrétně ukotvení a zobrazení 3D objektů v prostoru. Protože se jedná o poměrně velkou část, je blíže rozepsána v samostatné části.

Pro propojení s rozšířenou realitou existuje na poli mobilních zařízení nativní řešení pro obě platformy (Android i iOS). Pro Android se tato platforma nazývá ARCore [15], pro iOS pak ARKit [16]. Obě platformy jsou již delší dobu v produkční verzi, jsou značně pokročilé a pokrývají velké množství funkcionality. Nevýhodou těchto řešení pro multiplatformní aplikace je, že by se pro každou z platforem muselo vytvořit zvláštní řešení a sdílení mezi nimi by bylo značně komplikované. ARCore totiž umožňuje ukládat pozici bodů pomocí tzv. „Cloud Anchors“ do Google Cloud řešení. Co se ARKit týče, tam je možnost vytvořit si soubor, který v sobě obsahuje data, a ten si pak uložit buď přímo na zařízení, nebo synchronizovat přes svůj vlastní server. Struktura dat pro obě platformy je však jiná a nešlo by vytvořit jednotný obsah bez dalších modifikací.

Pokusy s těmito službami již v minulosti byly řešeny jako zadání jiných bakalářských prací, ale prozatím se žádná z nich nejeví jako dlouhodobé řešení problému. Proto se zde nabízí možnost při tvorbě editoru rovněž prozkoumat stávající řešení jiných poskytovatelů, zda by se nenašla platforma, která bude pro tyto účely fungovat podstatně jednodušeji.

■ 2.5.1 Wikitude

Komerční řešení zabývající se tvorbou SDK pro AR. Mají k dispozici taktéž balíček pro Flutter. Nevýhodou je příliš vysoká cena a obtížné řešení kotvení objektů v prostoru. Více informací na <https://www.wikitude.com/>

■ 2.5.2 Immersal SDK

Komerční řešení, které nabízí i možnost využití zdarma pro menší počty uživatelů. Immersal poskytuje SDK pro Unity, které umožňuje ukotvit objekty do zmapované scény. Díky propojení s Unity funguje pro platformy Android i iOS. Více informací na <https://immersal.gitbook.io/sdk/>

Na základě průzkumu a konzultace s vedoucím práce bylo rozhodnuto, že pro účely práce bude využito SDK společnosti Immersal. Jiné možnosti se v době psaní práce nezdály býti vhodné pro řešení daných problémů.

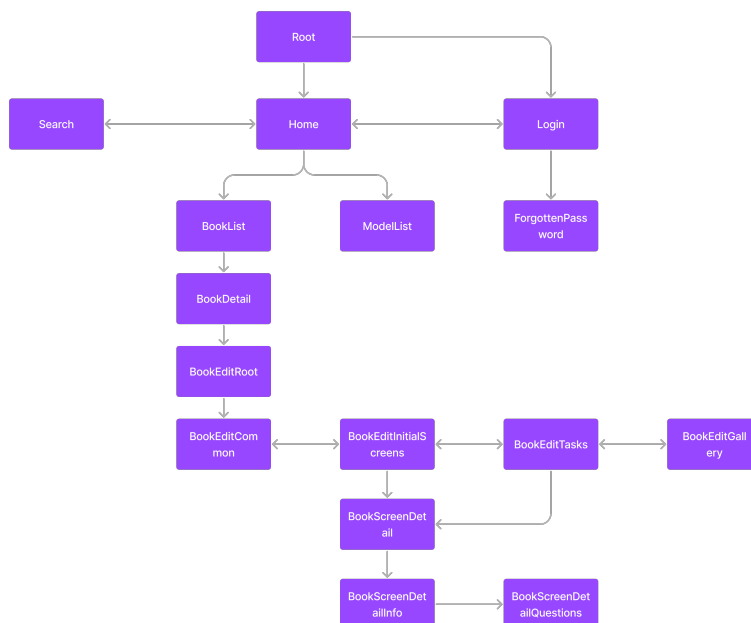
Kapitola 3

Návrh řešení

Dle úvodní analýzy, sběru požadavků a hlavně díky podnětu, že se má jednat o mobilní editor, bylo rozhodnuto pro ideální řešení ve formě multiplatformní mobilní aplikace. Práce s audiovizuálním obsahem je v mobilních aplikacích jednodušší. Rovněž je zde požadavek na propojení s rozšířenou realitou, pro což jsou mobilní aplikace vhodnější alternativou těch webových zejména díky snazšímu přístupu k hardware zařízení a tím pádem vyšší výkonnosti software.

3.1 Flow obrazovek

Než začne samotný návrh obrazovek, je dobré si rozmyslet, jaká bude jejich hierarchická struktura. Tento aspekt je důležitý hned z několika důvodů. Prvních z nich je uživatelská zkušenost a cíl na co největší jednoduchost rozhraní, dále pak také struktura a předávání dat mezi jednotlivými obrazovkami. Obrázek 3.1 ukazuje bližší rozpad této struktury.



Obrázek 3.1: Flow obrazovek

3.2 Návrh wireframe

Pro návrh uživatelského rozhraní je dobré začít takzvanými „wireframes“, tedy návrhem rozvržení ovládacích prvků na obrazovce pro který není důležité zaměřením na barvy, styly apod. V tomto případě bylo stylů využito, protože byly návrhy tvořeny za použití knihovny komponent pro software Figma [17]. Jelikož již existuje webová verze editoru, bylo k návrhu přistoupení se zaměřením na to využít co nejvíce prvků stejných nebo podobně řešených. Zejména se jedná o hlavní navigaci v rámci úpravy učebnice. Zde bylo využito takzvaných „tabů“ usnadňujících přepínání mezi kontextem obrazovky.

Část návrhu se může dle potřeb provádět opakovaně s různými variacemi uživatelského rozhraní, které budou následně vyzkoušeny během uživatelského testování a v případě potřeby budou upraveny dle nalezených poznatků - budeme postupovat v iteracích. Obecně bude probíhat návrh pomocí metodiky nazvané User-Centered Design (dále zkratka UCD), která se zaměřuje na návrh uživatelského rozhraní (dále zkratka UI) s úzkým zaměřením na cílové uživatele jako takové a jejich konkrétní potřeby a zvyklosti [18]. Tato metodika a postup budou dále popsány v následujících kapitolách, zejména pak v kapitole Návrh řešení.

3.2.1 Přihlášení

Výchozím a nezbytným bodem aplikace je přihlášení, bez něj by nebylo možné vykonávat v aplikaci žádné další úkony - viz Obrázek 3.2. V úvodní části je stručně popsáno o jakou aplikaci se jedná, pod popiskem pak formulář pro zadání přihlašovacího jména a hesla. Obzvláště na mobilních zařízeních může být někdy zadání hesla složitější, proto je zde navržena možnost zobrazit heslo pomocí ikonky oka v pravé části vstupu pro zadání hesla. Pod akčním tlačítkem pak tlačítko pro obnovu hesla v případě, že by uživatel své heslo zapomněl.


Obrazovka pro obnovu hesla není implementována, protože by byla zapotřebí nová logika na serveru, která by rozlišila zda se jedná o požadavek na resetování hesla z webu nebo z mobilního zařízení a vygenerovala by adekvátní odkaz buď pro otevření webové stránky nebo otevření aplikace. Nejedná se však o složitou funkcionalitu, tudíž může být dodána po úpravách aplikačního serveru. Prozatím je tlačítko přeměřováno na web, kde si uživatel heslo může bez problému změnit.

EduARd

Aplikace pro učitele pro tvorbu virtuálních naučných stezek v přírodě nebo ve městě.

Uživatelské jméno

Heslo

Přihlásit

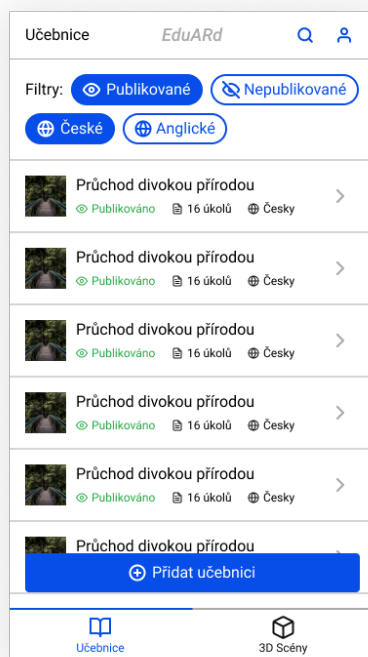
[Zapomenuté heslo](#)

Obrázek 3.2: Přihlašovací obrazovka

3.2.2 Seznam učebnic

Aplikační server na dotaz učebnic vrátí seznam všech učebnic, které jsou v dané organizaci vytvořené, můžeme vidět na obrázku 3.3. Aby došlo k usnadnění nalezení té právě hledané, obsahuje seznam filtry, které omezí počet výskytů výsledků. Filtry zabírají poměrně velkou část místa, proto bude dobré je při posouvání listu směrem dolů schovat a získat tak více prostoru pro seznam.

Jednotlivé karty seznamu obsahují úvodní obrázek učebnice, její název, stav, počet úkolů a jazyk. Obdobně je tomu tak i ve webové verzi editoru. Interakčním prvkem každé karty je pak takzvaný „tap“, což je krátký dotyk na kartu (klik myší), který otevře detail učebnice.

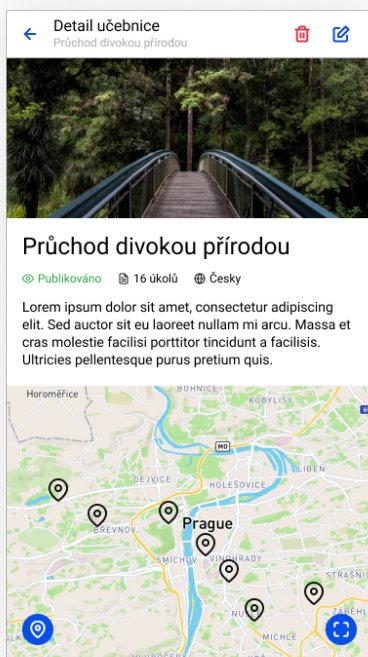


Obrázek 3.3: Obrazovka se seznamem učebnic

3.2.3 Detail učebnice

Obrázek 3.4 ukazuje takzvaný „Appbar“ v horní části obrazovky poskytuje zleva interakční tlačítko pro návrat zpět na výpis všech učebnic, dále pak popis, že se jedná o detail učebnice spolu s názvem učebnice samotné. Dále pak tlačítko pro smazání učebnice a tlačítko pro úpravu učebnice. Pod tímto barem se nachází úvodní obrázek učebnice, její detailní název, pak také klíčové atributy - stav, počet úkolů a jazyk. Pod tímto popisem se nachází mapa úkolů, které patří do dané učebnice. Mapu by mělo být možné zvětšit pomocí interakčních tlačítek.

Detail dále obsahuje popis, který může mít různou strukturu, jelikož je ve formátu HTML. Popis následuje mapa úkolů, které patří do dané učebnice. Mapu by mělo být možné zvětšit pomocí interakčních tlačítek.



Obrázek 3.4: Obrazovka detailu učebnice

3.2.4 Vytvoření a editace učebnice

Tvorba, ale rovněž i editace učebnic potřebuje pro své fungování stejný formulář, jen se změněným kontextem, jak naznačuje sada obrazovek na Obrázku 3.5. Buď se bude jednat o vytvoření zcela nové učebnice, nebo o modifikaci dat stávajících. Pro vytvoření se používají jiné koncové body API než pro úpravu, ale struktura dat je víceméně stejná s rozdílem, že při vytvoření musí nejprve dojít k vytvoření učebnice v JSON struktuře a až poté nahrání XML souboru.

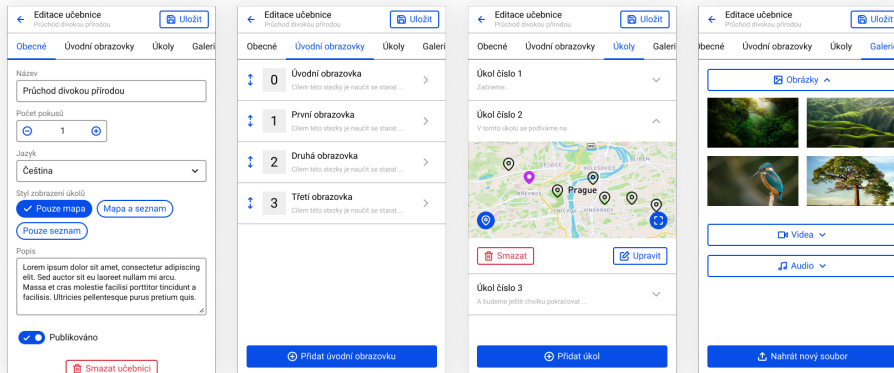
Editace učebnice probíhá na 4 základních obrazovkách - tzv. „Tabech“, mezi kterými lze přepínat v obou směrech. Inspirace pochází z webové verze, ale i z jiných aplikací a dává v tomto případě největší smysl. Ve vrchním „Appbaru“ je zleva interakční tlačítko pro návrat zpět, textové označení, zda se jedná o editaci, nebo vytváření spolu s názvem učebnice. Pravá část potom obsahuje tlačítko pro uložení dat celé učebnice. Vzhledem k povaze dat a zavedené funkcionalitě na webu by bylo nekonceptní k ukládání dat přistupovat jinak.

Obrazovka obecně obsahuje základní informace o učebnici. Za zmínku stojí vylepšené formulářové pole pro zadání počtu pokusů, které pro rychlejší manipulaci umožňuje zvyšovat či zmenšovat počet pokusů pomocí tlačítek plus a mínus.

Obrazovka úvodní obrazovky obsahuje seznam všech úvodních obrazovek učebnice, lze odtud přidat novou, upravit, smazat, nebo změnit pořadí obrazovek.

Obrazovka úkoly obsahuje seznam všech úkolů, které lze editovat a zároveň do nich přidávat nebo z nich odebírat jednotlivé obrazovky.

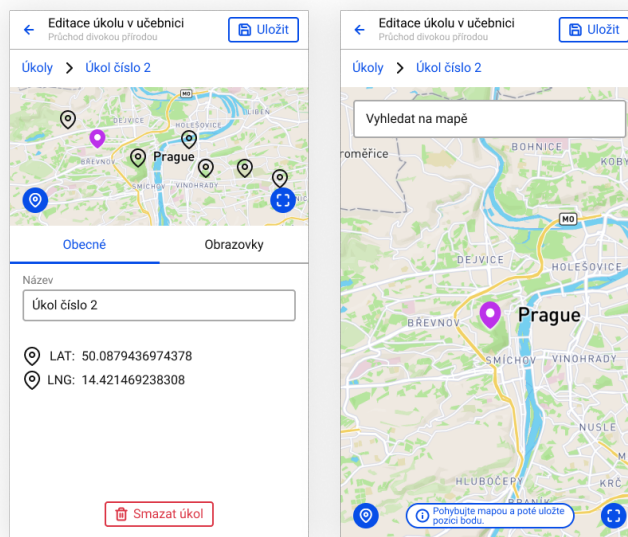
Obrazovka galerie obsahuje všechny druhy audiovizuálního obsahu, které uživatelé do dané učebnice nahráli. Jedná se primárně o přehled dat, které lze k jednotlivým obrazovkám přiřazovat. K rychlé orientaci budou sloužit filtry dle typu média - obrázek, video a audio.



Obrázek 3.5: Obrazovky pro vytváření a editaci učebnice

3.2.5 Editace úkolu v učebnici

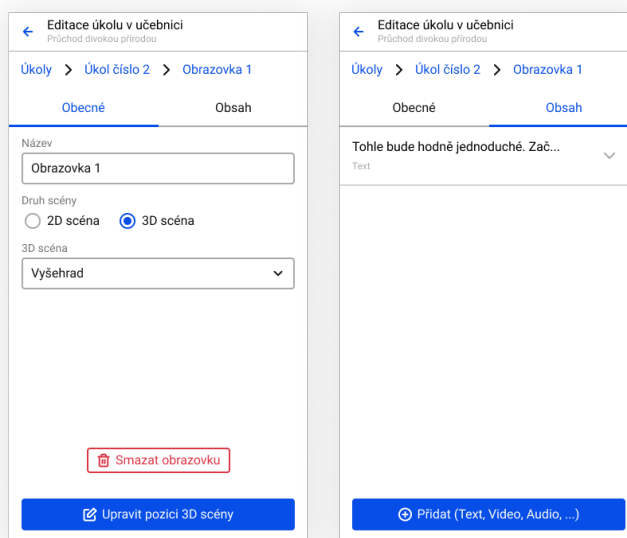
Každý úkol v učebnici je možné upravit, opět běžné operace pro editaci, smazání a vytvoření nového úkolu. Úkol má svůj název a pozici, kterou lze taktéž měnit, jak je patrné na Obrázku 3.6. K tomu by měla být využita na mobilním zařízení hlavně mapa, předpokládáme totiž, že upravovat pozici úkolu bude tvůrce stezky převážně na daném místě nebo v jeho okolí.



Obrázek 3.6: Obrazovky pro vytváření a editaci učebnice - detailu úkolu

3.2.6 Editace obrazovky úkolu

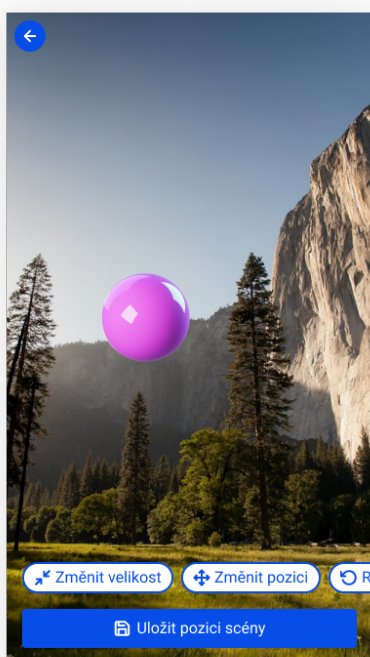
Kromě úkolu samotného lze upravit i jeho obrazovky (Obrázek 3.7). Konkrétně lze upravit název a druh scény. Pokud se jedná o 2D scénu, je možné přidat další obsah - otázky, porovnání obrázků a další audiovizuální obsah. Pokud se jedná o 3D obsah, je možné přidat k obrazovce 3D objekt.



Obrázek 3.7: Obrazovky pro vytváření a editaci učebnice - detail obrazovky

3.2.7 Přidání objektu do AR scény

Pro úpravu pozice 3D objektu v prostoru je nezbytné zanechat co největší možnou plochu pro zobrazení dat z kamery - obrazu okolí, na to bylo myšleno při návrhu obrazovky pro práci s AR, jak ukazuje Obrázek 3.8. Vybraný objekt pak už jen umístíme, nebo odstraníme pomocí tlačítka. Pokud to možnosti dovolí, bylo by dobré rovněž měnit velikost a rotaci objektu.



Obrázek 3.8: Obrazovky pro vytváření a editaci učebnice - úprava pozice 3D objektu ve scéně

Kapitola 4

Implementace

Pro práci byla hned z několika důvodů vybrána technologie Flutter [19] vyvinutá společností Google ve spojení s SDK v Unity od společnosti Immersal [20]. Bližší popis architektury a použitých technologií bude následovat v jednotlivých sekcích. Po implementaci dojde k uživatelskému testování, kde se ověří správná funkčnost aplikace a opraví případné chyby.

4.1 Flutter aplikace

Výchozím bodem celého řešení je multiplatformní mobilní aplikace ve frameworku Flutter - v jazyce Dart. Mezi důvody pro zvolení frameworku patří zejména jeho rychlost, protože se kompiluje do nativního kódu pro danou platformu [21], velká komunita, progresivní vývoj a v neposlední řadě taktéž jeho rostoucí popularita.

Flutter poskytuje nepřeborné množství připravených „Widgetů“ [22], které stačí pouze importovat z knihovny Flutter Material. Jak již z názvu napovídá, jedná se o elementy vytvořené dle specifikací a principů designového jazyka „Material Design“ rovněž od společnosti Google [23].

Před samotným programováním je potřeba rozmyslet architekturu a strukturu aplikace samotné, aby bylo možné udržet kód přehledný a snadno modifikovatelný. Existují obecně známé zkratky těchto programovacích paradigmat

a těmi jsou KISS (Keep it simple, stupid) a DRY (Do not repeat yourself). Pokud není zřejmé z anglického názvosloví, kód by měl být co nejjednodušší a programátor by neměl podobné části kódu neustále opakovat. Pro používání aplikace je potřeba správně navrhnout strukturu obrazovek a potřebnou datovou strukturu pro jejich případnou obsluhu. Struktura obrazovek již byla navržena v jedné z předchozích kapitol, nyní je možné postoupit dále.

■ 4.1.1 Architektura aplikace

Flutter aplikace se většinou označují za vrstvené, ostatně jako návrh frameworku samotného [24]. Pro potřeby práce s daty, modely a obrazovkami došlo k použití některých prvků architektury Model View View Model (MVVM), která odděluje datovou část od části UI, a přináší tak přehlednější a lépe čitelný kód. Některé vhodné prvky jsou využity z architektury Model View Controller (MVC), která navíc přidává ještě vrstvu controlleru, jenž se stará a složitější logiku aplikace [25]. Bližší popis pak nabízí následující výpis adresářové struktury, kde je vše blíže specifikováno.

Výchozím adresářem pro zdrojové kódu Flutter projektu je složka nesoucí název „lib“. Další strukturu pak framework sám o sobě nenutí a nechává volnou ruku programátorovi. V tomto případě byl zvolen systém členění následovně:

- **constants** - Sdílená neměnná data
- **models** - Třídy mapující modely z API do tříd v dartu a usnadňující tak použití v rámci aplikace
- **providers** - Poskytovatelé pro jednotlivé subčásti stavu dat
- **router** - Pomocné třídy sloužící pro navigaci aplikací
- **screens** - Hlavní Widgety obsahující jednotlivé obrazovky
- **services** - Pomocné třídy, které obsluhují převážně práci s API a složitější transformace dat
- **styles** - Nastavení barevných schémat, velikostí fontů apod.
- **utilities** - Pomocná sdílená funkcionality, která se přímo netýká konkrétních dat
- **widgets** - Sdílené Widgety napříč jednotlivými obrazovkami v rámci celé aplikace

■ 4.1.2 Použité rozšiřující balíčky

Pro usnadnění vývoje bylo využito hned několika rozšiřujících balíčků vytvořených buďto přímo tvůrci Flutteru nebo pak také komunitou. Ty z nich, které poskytují komplexní nebo nějakým způsobem zajímavou funkcionalitu, jsou níže popsány.

- **Dio**¹ - Http klient pro vykonávání dotazů na API. Dio je jedním z nej-používanějších balíčků, jeho hlavní výhodou je možnost přidání tzv. „Interceptorů“, jež mohou obohacovat každý request o data například v hlavičkách - skvěle se hodí pro přidávání přístupového tokenu. Do budoucna se pak nabízí i možnost přidat rozšíření, které je schopné vytvářet cache proběhlých dotazů, respektive navrácených dat
- **Shared preferences**² - Balíček sloužící pro ukládání perzistentních dat lokálního úložiště zařízení. V tomto případě je ukládán přístupový token, který je pak dostupný i po vypnutí a zapnutí aplikace
- **Flutter html**³ - Tento balíček zajišťuje zobrazení HTML obsahu ve formátované podobě - namísto HTML značek a stylů se zobrazí výsledný obsah. Využit je protože popis každé knihy může být díky WYSIWYG editoru ve formě HTML
- **Provider**⁴ - Provider zajišťuje takzvaný „State management“, neboli zajišťuje persistenci dat napříč Widgety, které jsou jeho potomky. Využívá přitom nativní Widgety Flutteru, takže nepřináší další komplexnost, naopak oprostuje programátora od psaní opakujícího se kódu a vše činí přehlednější
- **Xml**⁵ - Parser, který se stará o přečtení XML souborů stažených z API a po sléze i naopak vytváří XML strukturu pro upravená data z jednotlivých modelů
- **Mapbox**⁶ - Mapový poskytovatel, který jako jeden z mála nabízí možnost stažení offline map. Pro účely práce prozatím tato možnost nebyla využita, ale nejedná se o složitou funkcionalitu, lze tedy do budoucna snadno přidat i stahování offline mapových podkladů
- **File picker**⁷ - Jak již z názvu napovídá, jedná se o balíček, který umožňuje vybrat soubor ze systému zařízení a předat ho dále do aplikace

¹<https://pub.dev/packages/dio>

²https://pub.dev/packages/shared_preferences

³https://pub.dev/packages/flutter_html

⁴<https://pub.dev/packages/provider>

⁵<https://pub.dev/packages/xml>

⁶https://pub.dev/packages/mapbox_gl

⁷https://pub.dev/packages/file_picker

4.2 Unity aplikace

Klíčovou funkcionalitou pro aplikaci je umístění a následné zobrazení 3D objektů v prostoru pomocí rozšířené reality. Jak již bylo nastíněno v předchozích kapitolách, nejlepší volbou pro kotvení a dané případy užití se jeví využití SDK třetí strany a to finské společnosti Immersal. Jedná se o součást napsanou pro herní engine Unity, který však neslouží pouze pro vývoj her, ale rovněž i pro celou řadu dalších aplikací zaměřených na práci s prostorem.

Společnost nabízí kromě samotného SDK také ukázkou využití tohoto kitu v podobě jednotlivých dem. Pro účely zobrazení respektive ukotvení 3D objektů v prostoru existuje demo s příhodným názvem „Content placement demo“. Díky technologii kterou Immersal pro svůj kit využívá je však potřeba nejprve zmapovat okolí, do něž chceme objekt vložit. Mapování probíhá pomocí dema s názvem „Mapper“. Povaha demo projektů je taková, že jsou organizovány do několika scén v rámci jednoho projektu. Mezi scénami lze během běhu aplikace libovolně přepínat, což přináší skvělou možnost pro snadné používání.

Nyní zbývá ještě vyřešit, jak získat data modelů uložená v databázi. Kromě získání dat je potřeba rovněž vytvořit modely v Unity, aby je aplikace mohla zobrazit. Pro tyto účely již byla v minulosti aplikace napsána. Část funkcionality tedy pochází z dem přímo od společnosti Immersal a další část - stahování a parsování modelů pro Unity pochází z bakalářské práce Dominika Truonga, která se zabývala zobrazením 3D scén ve formátu X3D používaných v rámci systému EduARd [4]. Bylo potřeba spojení dema a části bakalářské práce, aby mohla vzniknout aplikace, která je schopna modely zobrazit a ukotvit v prostoru. Stažení a zpracování dat se sice podařilo, ale správné zobrazení za pomoci demo aplikace Immersal nikoliv. Zobrazení modelů z aplikačního serveru bylo ponecháno pro budoucí rozvoj.

Dvě zmíněné ukázkové aplikace byly zabaleny do jednoho Unity projektu, který nyní obsahuje dvě scény, mezi kterými lze přepínat pomocí tlačítka. Umísťování objektů vyžaduje, aby bylo okolí nejprve zmapováno a byla na serveru vytvořena mapa. Z tohoto důvodu byla dopsána ještě část kódu v Unity, která řeší stažení zpracovaných map ze serveru. Tyto mapy se posléze aplikace sama snaží napojit na místo, kde se uživatel právě nachází.

4.3 Flutter - Unity bridge

Nyní je zapotřebí obě aplikace vhodným způsobem spojit. Nabízí se možnost nechat je fungovat samostatně a pro prohlížení a ukotvování modelů používat pouze tu část napsanou v Unity. Existuje však podpůrný balíček pro Flutter, který přidává možnost ve Flutter aplikaci spustit tzv. „view“ s Unity aplikací a dále s ní pak komunikovat [26].

Integrace samotná vyžaduje několik nezbytných kroků pro obě mobilní platformy (Android i iOS), které jsou blíže popsány v návodu rozšiřujícího balíčku [26]. Nad rámec návodu musela být některá nastavení upravena, aby správně fungovala geolokace a zasílání zpráv mezi oběma aplikacemi. Tyto kroky jsou popsány v aplikačním kódu.

Komunikace s Unity aplikací probíhá pomocí controlleru - tzn. instance třídy `UnityWidgetController`, respektive `Widget` samotný přidává celou řadu callback metod, na které se dají navěsit vlastní metody. Například pomocí metody `postMessage(string gameObject, string method, dynamic data)` lze zavolat konkrétní metodu v Unity. Je potřeba, aby metoda byla `public` a aby třída využívala rozhraní `IEventSystemHandler`. Pro ukázkou zde uvádím část kódu (Kód 4.1) pro komunikaci s Unity.

```
// Komunikace z Flutteru do Unity
void setRotationSpeed(String speed) {
    _unityWidgetController.postMessage(
        'Cube',
        'SetRotationSpeed',
        speed,
    );
}

// Komunikace z Unity do Flutteru
void onUnityMessage(message) {
    print('Message from unity: ${message.toString()}');
}

// Callback metoda, která přiřadí obdržený controller
// na vytvořenou deklaraci
void onUnityCreated(controller) {
    _unityWidgetController = controller;
}

// Callback metoda, která dává informace o právě
// načtené Unity scéně zpět do Flutter aplikace
void onUnitySceneLoaded(SceneLoaded sceneInfo) {
    print('${sceneInfo.name}');
    print('${sceneInfo.buildIndex}');
}
```

Kód 4.1: Komunikace z Flutter aplikace do Unity

Pomocí těchto zpráv se do Unity aplikace při startu odešle identifikátor 3D scény, kterou aplikace načte z aplikačního serveru a posléze zpracuje data ve formátu X3D. Umístění objektu si pak SDK načte z interních dat - tato data jsou běžně uložena v souboru na konkrétním zařízení.

Kapitola 5

Testování aplikace

Nedílnou součástí vývoje každého software je testování. Existuje hned několik druhů testování, kde se každé zaměřuje na jinou problematiku a část aplikací. Důraz bude kladen na automatizované testování pomocí kódu přímo ve Flutteru, aplikace v Unity automatizovaně otestována nebude, neboť její hlavní část je ovlivněna využitím SDK a již vytvořené práce. Dále bude aplikace jako taková otestována uživatelským testováním na předem připravených scénářích průchodu aplikací.

5.1 Unit testy

Flutter nabízí vývojářům sadu testovacích nástrojů, mezi které patří unit testy, testy widgetů a taktéž integrační testy [27]. Vzhledem k tomu, že došlo k uživatelskému testování, nemá příliš smysl se zaměřovat na testy widgetů nebo integrační testování jednotlivých větších funkcionalit aplikace, a proto byl kladen důraz na vytvoření unit testů.

5.1.1 Pokrytí aplikace

Pokrytí testy není kompletní. Jak z praxe vypovídá, kvalita a zaměření testů je důležitějším aspektem nežli jejich množství. Pro testování bylo vybráno

načtení dat o učebnicích z formátu XML do modelů v jazyce Dart. Tyto modely se používají napříč celým editorem a jsou nezbytné pro správné fungování téměř všech dílčích částí aplikace. Dále musí fungovat i opačná transformace dat z modelů v Dartu do XML struktury. Tato transformace byla rovněž otestována. V neposlední řadě byly přidány testy pro `BookXmlService`, která řeší úpravy jednotlivých částí struktury učebnice - úkoly, obrazovky a otázky.

5.2 Uživatelské testování

Testy prováděné uživateli za reálných podmínek odhalují primárně chyby a nedořešené části z hlediska uživatelské přívětivosti grafického rozhraní. Mohou se objevit taktéž i chyby funkčnosti aplikace, ale těch by díky automatizovaným testům nemělo být příliš. Je dobré, aby participant ozřejmil své myšlenky nahlas, aby si tester mohl dělat poznámky a části rozhraní, které nejsou dostatečně srozumitelné nebo jsou matoucí tak mohl vylepšit.

5.2.1 Participanti testování

Pro testování byli vybráni 4 participanti, jejich věkové rozmezí se pohybovalo od 21 do 34 let. Jedná se o participanty se středním a vysokoškolským vzděláním. Participanti v tomto věkovém rozmezí a s tímto vzděláním byli vybráni záměrně s uvážením cílové skupiny, která by měla používat editor pro tvorbu učebnic.

Pro testování byla využita zařízení s operačním systémem Android, protože v době testování nebyla dořešena distribuce aplikace na zařízení s operačním systémem iOS. Jednalo se o různá zařízení, největší zastoupení zde měla zařízení s operačními systémy Android 11 a Android 12.

Jednotlivé průchody byly zaměřeny na menší úpravy textů, práci s audio-vizuálním obsahem a práci s rozšířenou realitou. Pro průchodu všemi scénáři měli participanti volně procházet aplikací a odhalit případné další chyby.

■ 5.2.2 Testovací scénáře

Pro testování byly zvoleny testovací scénáře, které se týkají tížených běžných úkonů, které tvůrci učebnic budou v mobilní aplikaci tvořit. Jedná se o práci s audiovizuálním obsahem a drobné úpravy textů nebo pořadí obrazovek.

■ Scénář číslo 1 - Přidání nové fotografie k úkolu

Vytvořili jste stezku (učebnici), která prochází vašimi oblíbenými místy. Během procházení chcete přidat nějaké fotografie, aby lépe ilustrovaly jak to na stezce vypadá.

1. **Přihlášení do aplikace** - Přihlaste se pomocí přihlašovacího jména a hesla
2. **Výběr učebnice** - Vyberte učebnici s názvem „Editace učebnice“
3. **Přepnutí do editace** - Otevřete učebnici v editačním režimu, abyste mohli přidat vybranou fotografii
4. **Nalezení úkolu** - Nalezněte úkol, na kterém se právě nacházíte a v něm přidejte úvodní obrazovku. Vyplňte její název a obrazovce nastavte vybranou fotografii
5. **Uložení postupu** - Nyní je potřeba, abyste učebnici uložili a ostatních tak mohli vidět, co bylo vytvořeno. Na konci pak zkontrolujte, že došlo ke správnému přiřazení obrázku k obrazovce

■ Scénář číslo 2 - Přiřazení 3D scény k obrazovce a její ukotvení (zmapovaná scéna)

Našli jste model, který chcete na nějakém zastavení vaší stezkou zobrazit a odkázat tak například na zříceninu hradu.

1. **Přihlášení do aplikace** - Přihlaste se pomocí přihlašovacího jména a hesla

2. **Výběr učebnice** - Pomocí filtrů můžete vaši učebnici nalézt snáze - vyfiltrujte si pouze aktivní učebnice a najděte tu, kterou chcete upravit
3. **Editace učebnice** - Přejděte do editace učebnice, abyste mohli přidat AR model
4. **Editace obrazovky v úkolu** - Zvolte si úkol, který chcete upravit (nebo nějaký přidejte) a následně buď upravte nebo přidejte novou obrazovku. Pro přidání modelu zvolte možnost 3D scény, kterou následně vyberte a ukotvíte ji v prostoru
5. **Uložení učebnice** - V závěru svou práci uložte a opusťte režim úprav

■ Scénář číslo 3 - Vytvoření učebnice s úkoly

Zrovna se procházíte se svými žáky městem a napadá vás spousta možností jak další procházku udělat zajímavější. Co kdyby na místech, kde zrovna jsme byly nějaké úkoly? Připravte si základní kostru učebnice, kterou pak můžete dokončit na webu.

1. **Přihlášení do aplikace** - Přihlaste se pomocí přihlašovacího jména a hesla
2. **Vytvoření učebnice** - Přejděte k vytvoření učebnice
3. **Obecné informace** - Zadejte název učebnice, další pole vyplňte dle svého uvážení
4. **Uložte učebnici** - Uložte svou práci, abyste nepřišli o vytvořená data
5. **Přidejte úvodní obrazovky** - Do učebnice přidejte alespoň jednu úvodní obrazovku, jméno vyplňte dle svého uvážení a přidejte k ní úvodní obrázek
6. **Přidejte obsah úvodní obrazovky** - Úvodní obrazovce, kterou jste právě vytvořili přidejte například audio. Pokud neexistuje, vytvořte nějaké a posléze ho přidejte
7. **Vytvořte úkol** - Po dokončení úprav úvodní obrazovky můžete svou práci opět uložit. Pokračujte vytvořením nového úkolu, jemuž přiřadte libovolné jméno. V úkolu vytvořte obrazovku, které opět přidejte libovolné audio
8. **Uložte svou práci** - Jakmile budete mít dokončeno, uložte svou práci a zkontrolujte, že je vše jak má být

■ Scénář číslo 4 - Přiřazení 3D scény k obrazovce a její ukotvení (nezmapovaná scéna)

Obdobně jako je tomu ve scénáři číslo 2, chcete umístit model do AR scény, tentokrát však scénu nikdo před vámi nezmapoval. Bez namapované scény není možné model správně umístit.

1. **Přihlášení do aplikace** - Přihlaste se pomocí přihlašovacího jména a hesla
2. **Výběr učebnice** - Vyfiltrujte si pouze aktivní učebnice a jednu z nich si vyberte
3. **Editace učebnice** - Přejděte do editace učebnice
4. **Editace obrazovky v úkolu** - Zvolte si libovolný úkol (nebo nějaký přidejte) a následně buď upravte nebo přidejte novou obrazovku. Zvolte možnost 3D scény, kterou následně vyberte a ukotvíte ji v prostoru
5. **Mapování scény** - scéna není zmapovaná, objekt by nešlo zachovat, přepněte se prosím do mapovacího režimu
6. **Mapování scény - vytvoření mapy** - nyní vytvořte sadu fotografií svého okolí, pozor - fotografie musí navazovat. Pokud by fotografie nenažovaly, mapovací aplikace vás na to upozorní. Pokud máte pocit, že jste již vytvořili dostatečné množství fotografií, pokračujte do dalšího bodu
7. **Mapování scény - online zpracování mapy** - nyní je potřeba vytvořit mapu, to se dělá v nastavení, kde odešlete snímky pro konstrukci mapy (pozn. zde jsem raději trochu navedl, protože to nemusí být zcela intuitivní). V seznamu map se posléze objeví vaše mapa, vyčkejte na její dokončení. Jakmile bude dokončena, přejděte k následujícímu kroku
8. **Přepnutí do kotvicí scény** - Nyní, když přepnete na kotvení scény, aplikace by měla detekovat vaši pozici a namapovat ji na aktuálně vytvořenou scénu (vlevo nahoře, zelený indikátor). Jakmile je toto splněno, můžete ukotvit objekt a učebnici uložit

5.2.3 Zjištěné nálezy z testování

Uživatelské testování odhalilo několik menších chyb, jejichž bližší popis je možné najít v tabulce 5.1. Víceméně bylo potřeba některé prvky zvýraznit nebo vyměnit. Největší problémy měli uživatelé s pochopením funkce AR části, která pro účely práce byla upravena zejména po technické stránce a její grafické rozhraní nebylo měněno. Testování odhalilo i několik menších technických chyb, které byly opraveny.

Nález	Priorita nálezu	Řešení
Jsem trochu zmaten z toho zanoření obrazovek a úkolů, v jisté chvíli už pomalu ztrácím přehled o tom kde jsem	Střední	Došlo ke zvětšení prvku tzv. „Breadcrumbs“, který ukazuje celou strukturu navigace (kde se uživatel právě nachází) na detailech obrazovek. Dále také k barevnému odlišení aktivního prvku
Mapování scény mi připadá složité, bez návodu bych to asi nedokázal	Střední	Do budoucna vynechat z grafického prostředí části pro AR ovládací prvky, které nejsou potřeba - nebylo řešeno, protože se jednalo o část v Unity
Přepínání v mapovací části by mě nenapadlo, je tam celkem hodně tlačítek	Střední	Celé grafické prostředí části pro AR by chtělo do budoucna zjednodušit, ale protože se jednalo hlavně o vyzkoušení funkcionality, nebude zapracováno
Vyděsila mě červená hláška hned při startu aplikace	Nízká	Jedná se o hlášku, která je zobrazena pokud nastane chyba při stahování aplikace (v tomto případě se jedná o porušená data uložená na serveru)
Ikona pro 2D obsah - tam mi spíš připadá, že se nějak špatně nahrála, než že by to mělo něco znamenat	Nízká	Ikona byla nahrazena jinou, více vystihující

Tabulka 5.1: Odhalené chyby z uživatelského testování

Kapitola 6

Závěr

V úvodu práce jsem se zabýval analýzou stávajícího řešení a možností, následoval návrh řešení v podobě prototypu mobilní aplikace. Po dokončení prototypu došlo k samotné implementaci, která měla tři části. První částí bylo vytvoření aplikace postavené na frameworku Flutter, další částí pak aplikace v Unity a nakonec pak tyto dvě aplikace propojit. Po implementaci přišlo na řadu testování, ze kterého vyplynulo několik menších chyb, které byly opraveny.

6.1 Splněné cíle

Cílem práce bylo vytvořit multiplatformní mobilní aplikaci pro editování učebnic zaměřenou na práci s multimediálním obsahem. Zvláštní část zde má 3D obsah, pro nějž bylo zapotřebí vymyslet, jak se bude ukotvovat v prostoru a následně zobrazovat. Mobilní aplikace by se neměla příliš lišit od její webové verze, protože by to mohlo působit uživatelům značnou míru zmatení a frustrace. Formát dat musel zůstat kompatibilní napříč různými aplikacemi v systému EduARd, čehož bylo docíleno zejména dodržením struktury XML a ověřováním správného zobrazení dat jednotlivých učebnic ve webové verzi.

Výsledkem práce je aplikace dostupná jak pro Android, tak pro iOS, která umožňuje učebnice nejen editovat, ale i vytvářet nové. Pro iOS nebyla aplikace sestavena, protože to nebylo potřebné, ale v průběhu vývoje byla funkcionality ověřena na simulátoru. Mobilní editor se zaměřuje na práci s audiovizuálním

obsahem a menší úpravy vhodné pro mobilní zařízení. Tato část aplikace má až na drobné výjimky obdobnou funkcionalitu jako její webová verze.

Prohlížeč a kotvící aplikace pro 3D modely je napsaná v Unity, ale díky rozšiřujícímu balíčku pro Flutter je možné tyto aplikace propojit dohromady a vytvořit tak ze 2 aplikací aplikaci jednu. Aplikace v Unity se spouští v rámci dílčích částí aplikace ve Flutteru. K napojení aplikace v Unity bylo přistupováno tak, že se jedná o prototyp ověřující funkčnost a možnosti takového řešení jako celku. To je ostatně jedním z důvodů, proč nebyla aplikace v Unity příliš upravena a byla ponechána v podobném duchu jako původní ukázkové aplikace. Úpravy v Unity se týkaly propojení s testovacím účtem, stahováním zmapovaných dat a možností jednoduše přepínat mezi scénami pro mapování a kotvení. Pokus o propojení s částí bakalářské práce zaměřené na prohlížení modelů [4] proběhl, ale nebyl dotažen k úspěšnému konci. Data se sice z aplikačního serveru stáhnou a zpracují, avšak k úspěšnému zobrazení za pomoci ukázkové aplikace Immersal se nepodařilo dojít. Někdy jsou důvodem chybějící média, jindy není zřejmé, jak objekt správně umístit do AR scény vytvořené kotvící ukázkovou aplikací společnosti Immersal. Základní funkcionalita se však jeví jako dostatečně stabilní pro budoucí rozvoj.

6.2 Rozvoj aplikace do budoucna

Motivací pro tvorbu mobilního editoru byla primárně možnost pracovat s multimediálními daty, což se podařilo. Nebylo však myšleno na kompletní pokrytí funkcionality webového editoru, což by mohlo být jedním z bodů budoucího rozvoje aplikace. Dále zůstávají drobné nedodělky kvůli chybějící funkcionalitě na straně aplikačního serveru - například funkce pro obnovu hesla uzpůsobená pro mobilní aplikaci.

Co se prohlížení a ukotvování 3D obsahu týče, funkcionalita je implementovaná částečně. Na funkcionalitě v Unity se dá do budoucna dále pracovat, například z výsledků uživatelského testování je patrné, že grafické rozhraní mapovací aplikace je pro dané účely příliš komplikované. Zpracování modelů z aplikačního serveru je dalším bodem, který stojí za vylepšení. Jelikož se však jedná o poměrně komplexní funkcionalitu, byla ponechána pro budoucí úpravy.

Co se vývoje frameworku Flutter týče, v posledních verzích byla přidána podpora pro web a také aplikace pro osobní počítače. Pro usnadnění vývoje by se tak dalo využít jedné „codebase“ pro editor na mobilních zařízeních a na osobních počítačích.

Příloha A

Literatura

- [1] Ing. Jiří Herinek. Eduard: Authoring výukových aplikací s ar + gps. <https://fel.cvut.cz/cz/vz/produkty/eduard>, 2022. [Online; cit. 09.05.2022].
- [2] Jan Skála. *Systém pro správu uživatelů projektu EduARd*. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2019. Bakalářská práce.
- [3] Anastasia Surikova. *Webový klient pro správu mobilních výukových úloh systému EduARd*. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2019. Bakalářská práce.
- [4] Dominik Truong. *Georeferencovaná rozšířená realita*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačové grafiky a interakce, 2020. Bakalářská práce.
- [5] Lukáš Šimon. *Mobilní Flutter aplikace pro řešení výukových úloh v systému EduARd*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačové grafiky a interakce, 2021. Bakalářská práce.
- [6] Michaela Kubišová. *Mobilní Android aplikace pro řešení výukových úloh v systému EduARd*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačové grafiky a interakce, 2020. Bakalářská práce.
- [7] Šimon Maňour. *Virtuální naučné stezky v React Native*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačové grafiky a interakce, 2019. Bakalářská práce.

- [8] Google developers. Sensor types. <https://source.android.com/devices/sensors/sensor-types>, 2022. [Online; cit. 09.05.2022].
- [9] Jan Skála. Eduard api. <https://api.eduard.fel.cvut.cz/>, 2019. [Online; cit. 18.05.2022].
- [10] Matthew J. Hamm. *Wireframing Essentials*. Packt Publishing; Illustrated edition, Leden 2014. ISBN: 978-1849698542.
- [11] Michal Mráz. *Webový portál pro přípravu podkladů pro rozšířenou realitu*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačové grafiky a interakce, 2020. Bakalářská práce.
- [12] Michaela Vlčková. *Sdílení obsahu v systému EduARD*. České vysoké učení technické v Praze, Fakulta elektrotechnická, Katedra počítačů, 2022. Diplomová práce.
- [13] Douglas Crockford and Chip Morningstar. *Standard ECMA-404 The JSON Data Interchange Syntax*, volume 2. Ecma International, Prosinec 2017. DOI: 10.13140/RG.2.2.28181.14560.
- [14] John Cowan (Google), Andrew Fang (PTC-Arbortext), Paul Grosso (PTC-Arbortext Co-Chair), Konrad Lanz (A-SIT), Glenn Marcy (IBM), Henry Thompson (W3C Staff Contact), Richard Tobin (University of Edinburgh), Daniel Veillard, Norman Walsh (Mark Logic Co-Chair), and François Yergeau. Extensible markup language (xml) 1.0. <https://www.w3.org/TR/xml/>, 2008. [Online; cit. 11.05.2022].
- [15] Google Developers. Overview of arcore and supported development environments. <https://developers.google.com/ar/develop>, 2022. [Online; cit. 11.05.2022].
- [16] Apple Inc. Arkit. <https://developer.apple.com/documentation/arkit/>, 2022. [Online; cit. 11.05.2022].
- [17] Evan Wallace and Dylan Field. Figma. <https://www.figma.com/>, 2022. [Online; cit. 18.05.2022].
- [18] Lowdermilk Travis. *User-Centered Design: A Developer's Guide to Building User-Friendly Applications*. O'Reilly Media, Inc., Duben 2013. ISBN: 978-1449359805.
- [19] Flutter developers. What is flutter. <https://docs.flutter.dev/resources/faq>, 2022. [Online; cit. 09.05.2022].
- [20] Immersal developers. What is the immersal sdk. <https://immersal.gitbook.io/sdk/>, 2022. [Online; cit. 09.05.2022].
- [21] Flutter developers. Technology. <https://docs.flutter.dev/resources/faq#technology>, 2022. [Online; cit. 10.05.2022].

- [22] Flutter developers. Widget catalog. <https://docs.flutter.dev/development/ui/widgets>, 2022. [Online; cit. 09.05.2022].
- [23] Alphabet Inc. Material design. <https://material.io/design>, 2022. [Online; cit. 09.05.2022].
- [24] Flutter developers. Flutter architecture. <https://docs.flutter.dev/resources/architectural-overview>, 2022. [Online; cit. 09.05.2022].
- [25] Martin Robert C. *Clean architecture*, volume 1. Prentice Hall, London, England, 2018. ISBN: 978-0-13-449416-6.
- [26] Rex Isaac Raphael. Flutter unity 3d widget. https://pub.dev/packages/flutter_unity_widget, 2022. [Online; cit. 10.05.2022].
- [27] Flutter developers. Testing flutter apps. <https://docs.flutter.dev/testing>, 2022. [Online; cit. 10.05.2022].

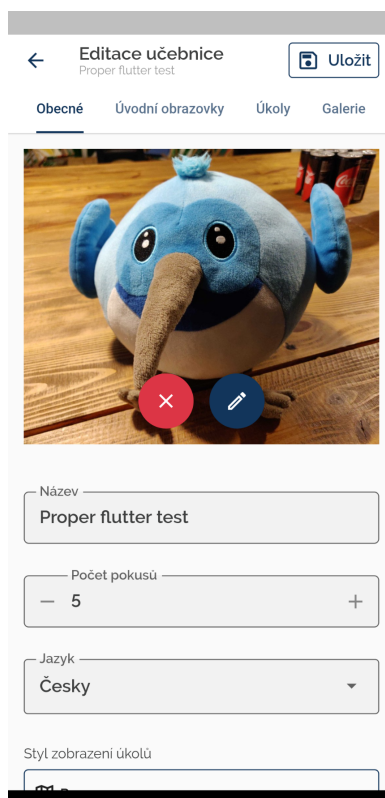


Příloha B

Návod pro testování

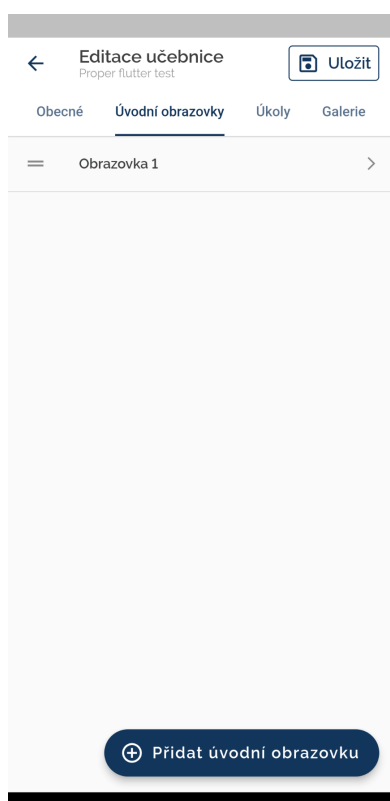
Pro lepší pochopení, zejména části pro práci s virtuální realitou je dobré se seznámit s tím, jak dema od společnosti Immersal fungují, respektive celý proces vytvoření mapy a následné umístění objektu do scény.

Než se dostaneme k práci s virtuální scénou, projdeme si jednotlivými kroky editace učebnice, které k ní vedou. Učebnici je potřeba nejprve vytvořit zadáním názvu a dalších obecných informací, jak ukazuje obrázek B.1.



Obrázek B.1: Obecné informace o učebnici

Po úspěšném vytvoření můžeme přidat několik úvodních obrazovek, které budou zobrazeny při otevření detailu učebnice v režimu procházení - viz obrázek B.2. Každá taková obrazovka může mít různý multimediální obsah.



Obrázek B.2: Úvodní obrazovky učebnice

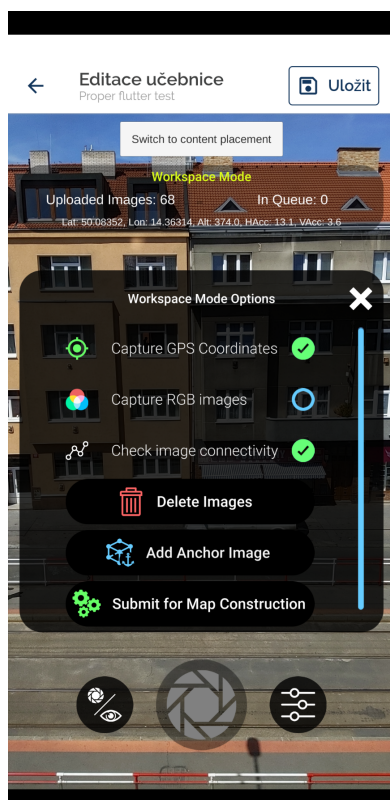
Po vytvoření úvodních obrazovek můžeme postupovat dále, na sekci s úkoly. Úkoly jsou lokalizovány pomocí GPS souřadnic a rovněž mají své jméno. Navíc oproti úvodním obrazovkám má každý úkol v sobě ještě několik dalších obrazovek. Obrazovky mohou kromě multimediálního obsahu mít referenci na 3D objekt umístěný v AR scéně.

Umístění je v tomto případě podmíněno existencí vytvořené mapy v mapovací části, která může být buďto již vytvořena někým jiným, nebo musí být pro konkrétní místo nově vytvořena. Vytvoření probíhá za pomoci GPS, dalších sensorických dat a také pomocí fotografií daného místa - ukázkou můžeme vidět na obrázku B.3. Je potřeba vytvořit několik fotografií, které na sebe budou navazovat (aplikace v případě úspěchu i neúspěchu zobrazí adekvátní hlášku). Vytvoření fotografií probíhá pomocí prostředního tlačítka ve spodní části - sensorická data si aplikace sama doplní.



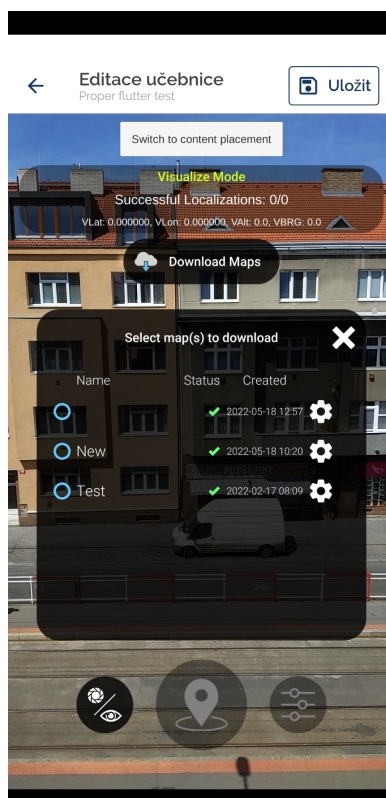
Obrázek B.3: Mapování scény

Jakmile je pořízeno několik fotografií, je potřeba odeslat data ke zpracování na servery společnosti Immersal. K tomu slouží tlačítko napravo vedle tlačítka pro pořízení fotografií. Zde vybereme možnost „Submit for Map Construction“, jak je vidět na obrázku B.4.



Obrázek B.4: Vytvoření nové mapy

Stav zpracování mapy můžeme sledovat přímo v aplikaci, jak ukazuje obrázek B.5. Tlačítkem nalevo od tlačítka pro pořizování fotografií se v mapovací části přepneme do vizualizačního režimu. Ve vizualizačním režimu si uprostřed v horní části obrazovky můžeme všimnout tlačítka „Download Maps“, které zobrazí přehled všech map a jejich stav.



Obrázek B.5: Přehled všech mých map

Když je mapa úspěšně zpracována (ukazuje se u ní zelená ikonka), můžeme se přepnout do části pro umístování objektů (obrázek B.6) pomocí tlačítka „Switch to content placement“, kde dojde ke stažení úspěšně zpracovaných map a následně se aplikace pokusí lokalizovat uživatele za pomoci interních mechanismů (GPS souřadnice a dalších senzorická data + námi vytvořená mapa). Pokud dojde k úspěšné lokalizaci, měla by se ikonka v levé horní části obrazovky obarvit zeleně a my víme, že nyní je možné objekt umístit do scény. Umístění je možné i dříve, avšak patrně nedojde k jeho správnému zachování na daném místě kvůli chybějícím geolokačním datům.



Obrázek B.6: Umístění modelu do AR prostoru

V současné době není dořešeno ukládání dat o pozici 3D objektu v prostoru na server, není tedy třeba pozici ukládat pomocí tlačítka „Uložit“. Pokud dojde ke správnému umístění v rámci zmapované scény, objekt by se měl uložit v rámci lokálního úložiště zařízení.