



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Mobilní Flutter aplikace pro řešení výukových úloh v systému EduARd

Lukáš Šimon

Vedoucí: Ing. Ivo Malý, Ph.D.
Obor: Softwarové inženýrství a technologie
Květen 2021

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Šimon** Jméno: **Lukáš** Osobní číslo: **483810**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Mobilní Flutter aplikace pro řešení výukových úloh v systému EduARd

Název bakalářské práce anglicky:

Mobile Flutter application for solving education tasks using EduARd system

Pokyny pro vypracování:

Provedte analýzu struktury projektu EduARd, zejména strukturu výukových aplikací a dále vazbu mobilní aplikace na ostatní části projektu (web, datové úložiště, editor úloh, aplikace v rozšířené realitě).

Na základě analýzy vytvořte seznam požadavků a navrhněte novou mobilní výukovou aplikaci, která bude nabízet uživatelům jednotlivé sady úloh a umožní jejich řešení. Zaměřte se zejména na přihlášení do systému a role uživatele, výběr a stahování jednotlivých sad úloh, a to jak pro vlastní aplikaci tak i pro AR aplikaci vyvinutou v rámci práce D. Truonga, Georeferencovaná rozšířená realita [5].

Maximalizujte možnosti aplikace v režimu bez přístupu k internetu.

Navržené řešení implementujte pomocí vývojové platformy Flutter pro platformu Android. Ověřte funkčnost aplikace pro různé velikosti displejů, tj. mobily i tablety na alespoň dvou výukových úlohách.

Seznam doporučené literatury:

1. M. Jones, G. Marsden, Mobile Interaction Design, Wiley, 2006
2. T. Lowdermilk, User-Centered Design, O'Reilly Media, 2013
3. M. Kubišová, Mobilní Android aplikace pro řešení výukových úloh v systému EduARd, Bakalářská práce, České vysoké učení technické v Praze, 2020.
4. J. Skála, Systém pro správu uživatelů projektu EduARd, Bakalářská práce, České vysoké učení technické v Praze, 2020.
5. D. Truong, Georeferencovaná rozšířená realita, Bakalářská práce, České vysoké učení technické v Praze, 2020.

Jméno a pracoviště vedoucí(ho) bakalářské práce:

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

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

Datum zadání bakalářské práce: **12.02.2021**

Termín odevzdání bakalářské práce: **21.05.2021**

Platnost zadání bakalářské práce: **30.09.2022**

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

Rád bych tímto poděkoval Ing. Ivovi Malému, Ph.D. za vedení práce a jeho profesionální asistenci. Chtěl bych také vzdát díky své rodině a přátelům za jejich podporu.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 20. května 2021

Abstrakt

Tato práce vzniká pro podporu moderního způsobu výuky 21. století. Mobilní aplikace bude multiplatformní při zachování nativní výkonnosti. Tato kombinace zabezpečí možnost masového rozšíření i mezi méně výkonné zařízení a je esenciální pro šanci na využití ve všech vzdělávacích institucích. V rámci této práce je aplikace analyzována a jsou vybrány konkrétní technologie pro implementaci. Rovněž je vytvořen design jednotlivých obrazovek. Konečně je práce implementována a následně otestována. Implementace je provedena ve frameworku Flutter, komunikace se serverem, který aplikaci dodává data, probíhá přes REST API. Aplikace je vytvořena v duchu material designu a ve snaze dosažení co nejvyšší uživatelské přívětivosti.

Klíčová slova: Flutter, Mobilní aplikace, Systém EduARd

Vedoucí: Ing. Ivo Malý, Ph.D.
Praha, Resslova 307/9

Abstract

This thesis is created to support a modern way of the 21—century education. The mobile app is going to be multiplatform with native performance. This combination secures the possibility of mass expansion even among less performant devices and is essential for the chance of usage in all the educational institutions. In this work, the application is analyzed, and particular technologies for implementation are selected. Complete design for each screen is also created. Finally, the application is implemented and also tested. The implementation is done using the Flutter framework, and communication with a server, which is supplying data, is accomplished by REST API. The application is created in a spirit of material design and with user-friendliness in mind.

Keywords: Flutter, Mobile applications, System EduARd

Obsah

1 Úvod	1	3.2.4 Profil	17
2 Analýza	3	3.2.5 Úvod učebnice	18
2.1 Systém	3	3.2.6 Úkoly	19
2.2 Entity	5	3.2.7 Rozložení	19
2.2.1 Questionset	5	4 Implementace	23
2.2.2 Task	6	4.1 Flutter	23
2.2.3 Questionslide	6	4.2 Architektura	24
2.3 Analýza existujících aplikací	7	4.3 Správa stavu	24
2.3.1 Původní aplikace	7	4.3.1 Provider	25
2.3.2 Lesní svět	9	4.3.2 Riverpod	25
2.3.3 Naučná stezka hajného Kázka	9	4.4 Směrování	26
2.3.4 Vyhodnocení	9	4.5 Data	26
2.4 Požadavky na nové řešení	9	4.5.1 JSON	26
3 Návrh	13	4.5.2 XML	27
3.1 Mobilní aplikace	13	4.5.3 Ukládání dat	27
3.2 Prototyp	13	4.6 Implementace uživatelského rozhraní	27
3.2.1 Přihlášení	14	4.6.1 Google fonts	28
3.2.2 Domovská obrazovka	15	4.6.2 MapBox	28
3.2.3 Stahování učebnice	16	4.6.3 Tmavý režim	28

4.7 Propojení s komponentou rozšířené reality	31
4.7.1 Komunikace se serverem	31
5 Testování	33
5.1 Automatické testování	33
5.1.1 Unit testy	33
5.2 Uživatelské testování	34
5.2.1 Distribuce účastníkům testování	34
5.2.2 Průchody aplikací	35
5.2.3 Výsledky uživatelského testování	37
6 Závěr	39
6.1 Budoucnost řešení	40
Bibliografie	41
A Návod pro instalaci	43

Obrázky

Tabulky

2.1 Interakce komponent systému a uživatelů [2]	4
2.2 Tmavý a světlý režim seznamu úloh.	8
3.1 Přihlášení	14
3.2 Domovská obrazovka	15
3.3 Stahování	16
3.4 Průběh stahování	16
3.5 Profil uživatele	17
3.6 Úvod učebnice	18
3.7 Ukázky úkolů	20
3.8 Ukázky rozložení	21
4.1 Tmavý a světlý režim domovské obrazovky.	29
4.2 Tmavý a světlý režim seznamu úloh.	30
4.3 Tmavý a světlý režim statistiky.	30

5.1 Zadání nového úkolu	37
-------------------------------	----



Kapitola 1

Úvod

Děti a mladiství jsou dnes vedeni k memorování a často nejsou schopni pochopit, co se učí. Abstraktnost předávané výuky následně vede ke ztrátě vůle se dále o témata zajímat. Toto a mnoho dalšího подрývá vzdělání mladých lidí a tím i přímo dopadá na vývoj celé země.

EduARd je systém umožňující propojení s virtuálním světem a skrz multi-mediální obsah je schopen zobrazit informace poutavě a názorně. V rámci systému jsou vytvářeny knihy (stezky), které obsahují nejen informace, ale rovněž mohou obsahovat i různé typy úkolů a přimět se nad vysvětlovanou látkou zamyslet. Systém je složen ze serverové části, webové aplikace a dvou mobilních klientských aplikací.

Cílem celé práce je implementovat mobilní klientskou aplikaci, která přebírá data vytvořena v aplikaci webové a uložené na serveru. Tato data (knihy) zobrazuje a v případě 3D obsahu je předává druhé mobilní aplikaci na zobrazení rozšířené reality.

Nejen ve školství je velký problém s nedostatkem financí, a proto nutnost kupovat zařízení pro žáky by vedlo k nerealizovatelnosti projektu. Tuto komplikaci řeší využití frameworku Flutter, který umožňuje vytvářet moderní aplikace běžící na operačním systému android i iOS. Mobilní aplikace systému EduARd tedy může běžet na zařízeních, která žáci většinou běžně mají.

Práce se nejdříve zaměřuje na analýzu systému a jeho komponent. Poté se věnuje datovému modelu a také existujícím řešením. V kapitole návrh ukazuje, jak byl vytvořen design, v implementační části popisuje tvorbu samotné aplikace a v kapitole o testování vysvětluje, jak bylo přistoupeno k testování aplikace.



Kapitola 2

Analýza

Tato kapitola se soustředí především na analýzu celého systému, původního řešení a dalších aplikací dostupných na trhu. Dále stručně popisuje datové entity v systému.



2.1 Systém

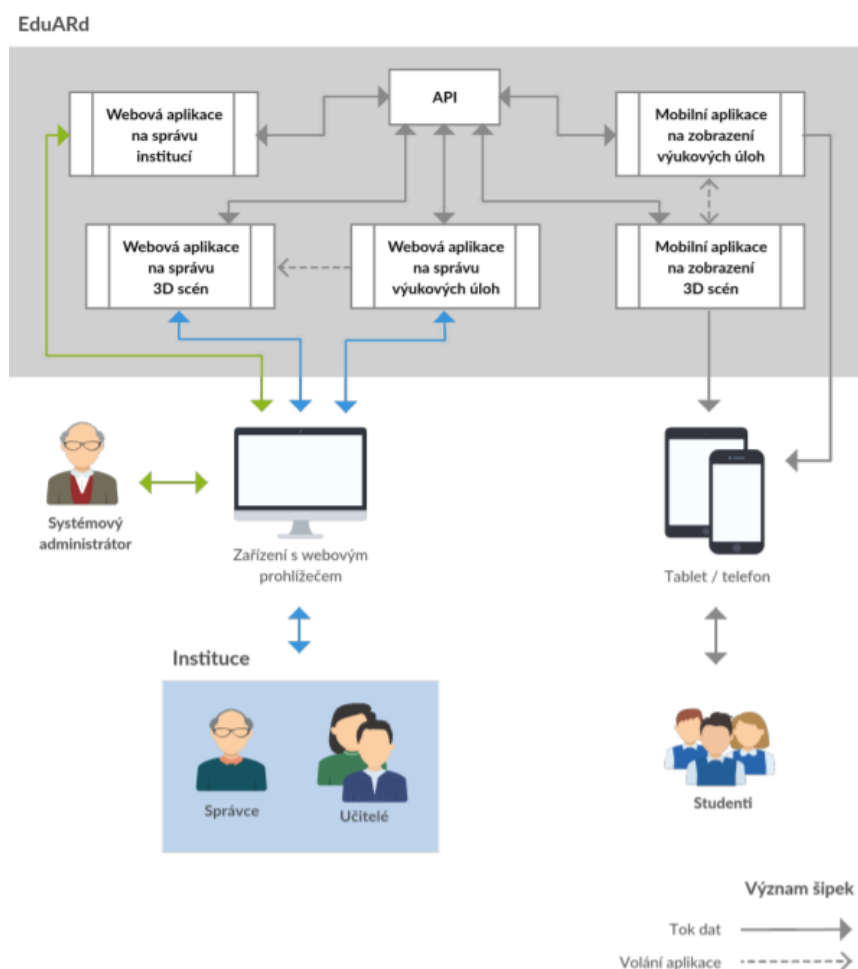
EduARd je komplexní systém postavený na mnoha technologiích a skládající se z několika esenciálních komponent uvedených na obrázku 2.1. Zajisté je třeba zmínit REST API [1], webového klienta [2], pak mobilní aplikaci [3], která bude přepsána na základě této práce a mobilní aplikaci pro zobrazení rozšířené reality.



REST API

Tato komponenta, jenž byla napsána v jazyce C#, obstarává veškerou logiku systému včetně správy uživatelů a zpracovává volání webového klienta i mobilní aplikace.

Dokumentace je k dispozici za využití technologie Swagger a popisuje jednotlivé koncové body API, kterých je několik desítek. S daty se pracuje ve formátu dnes velmi populárního JSON, jedinou výjimku tvoří učebnice, které přichází ve formátu XML.



Obrázek 2.1: Interakce komponent systému a uživatelů [2]

■ Webový klient

Webový klient slouží administrátorům pro správu systému a editorům jako nástroj tvorby obsahu. Vytváří se zde konkrétní učebnice, jejich obsah, rozložení a další. Umožňuje ovšem učebnice i mazat a upravovat, případně přidávat a odebírat jejich obsah.

Je zde také možnost přidat multimediální obsah, dále zobrazovaný v mobilní aplikaci uživatelům. Podpora je pro video soubory, audio soubory a samozřejmě také obrázky. Rovněž je zde možnost tvorby scén pro rozšířenou realitu, které jsou dále vloženy do knih jakožto obrazovka v úkolu.

■ Mobilní aplikace

Je psána přímo pro koncové uživatele a slouží jako konzument obsahu vytvořeného v rámci webového klienta. Hlavní účel této komponenty je přihlášení uživatelů, stažení a následná práce s učebnicemi.

Učebnice je možné procházet, zvolit si a otevřít. Dle typu učebnice, je možné si po otevření zobrazit úlohy na mapě a v seznamu, případně pouze v jedné z variant. Aplikace poskytuje také jednoduchou statistiku úspěšnosti a vyplněnosti učebnice.

Pro zobrazení 3D obsahu slouží další mobilní aplikace[4]. Ta dostává data i povely od řídicí aplikace vzniklé v rámci této práce.

■ 2.2 Entity

Pro běh aplikace a její účel zobrazit knihy je důležité analyzovat entity, které danou knihu reprezentují. Mimo knihu aplikace pracuje i s multimediálními soubory a 3D scénami. Dále stojí za zmínku i entita uživatele obsahující základní informace jako jméno, email a instituce.

■ 2.2.1 Questionset

Tento prvek je reprezentací samotné učebnice. Jeho atributy jsou maxTries a layout. Jak vyplývá z názvu atribut maxTries udává maximální počet pokusů povolených v rámci dané učebnice. Co se týká atributu layout, ten má definovány tři možnosti:

1. Seznam = Zobrazen je pouze seznam s úkoly.
2. Mapa = Zobrazena je mapa a na ní jsou jednotlivé úkoly.
3. Mapa se seznamem = Ve své podstatě jde o kombinace předchozích možností, zobrazena je mapa, která má i přepínač do seznamu s úkoly.

■ 2.2.2 Task

Jedna úloha v rámci učebnice. Atributy jsou name a title, které se zdají být duplicitní, avšak důvodem dvou atributů je, že jeden je určen pro programátora a druhý pro uživatele, tudíž k zobrazení. Dále obsahuje prvek location skládající se z prvků latitude a longitude, jak z názvu plyne, jde o GPS souřadnice. Nakonec je třeba zmínit atribut questionslide, jenž definuje jednu konkrétní obrazovku v rámci tasku. Jeden task může obsahovat více questionslide.

■ 2.2.3 Questionslide

Question slide, reprezentuje jednu obrazovku v rámci úkolu. Obsahuje znovu title i name, jejich opodstatnění je stejné. Velmi důležitý element je layout a ten má předem definované hodnoty, jež může mít přiřazené a to:

1. Předvolený
2. Obrázek vlevo od textu
3. Obrázek vpravo od textu
4. Obrázek pod textem
5. Obrázek nad textem
6. Otázka = Bude obsahovat otázku.

Dále může obsahovat description, images, questions. Questions se skládá z jednotlivých otázek, images obsahuje obrázky pro zobrazení a description je zadání otázky. Otázka má také další atributy a to:

1. Validate = (bool) Jestli se bude vyhodnocovat.
2. Shuffle = (bool) Jestli se budou odpovědi přehazovat.
3. CaseSensitive = (bool) Jestli bude odpověď brát ohled na velká a malá písmena.

4. Type = druh odpovědi

- Textová
- Číselná
- Pouze jedna správná
- Více správných
- Intervalová
- Seřadovací
- Přepínací

■ 2.3 Analýza existujících aplikací

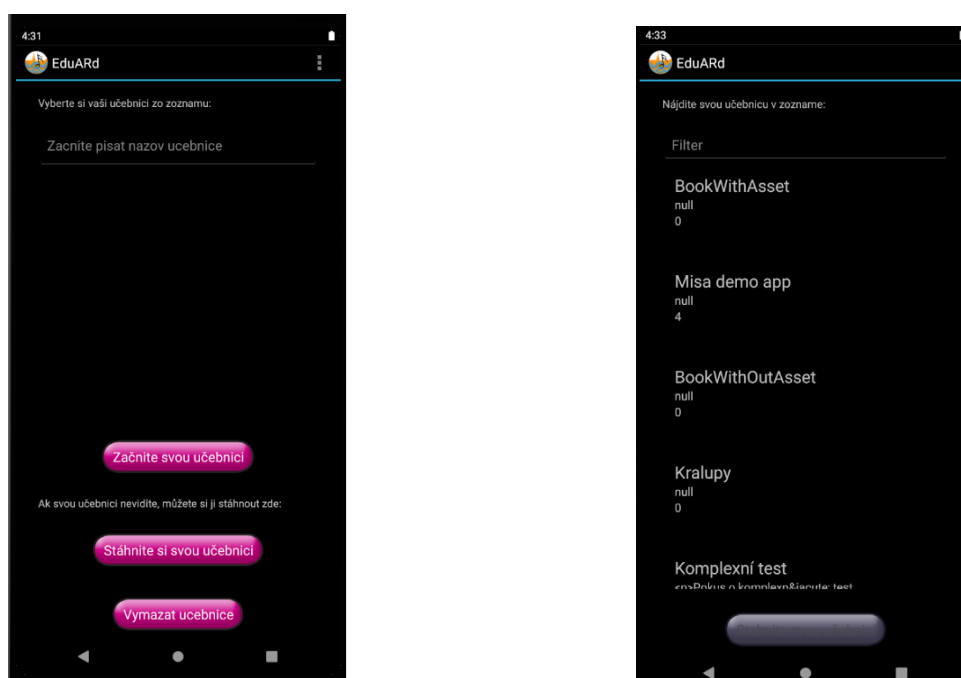
Tato aplikace je přímým nástupcem aplikace zpracovávané v rámci práce paní Kubišové [3]. Kromě té existuje několik více či méně podobných řešení napříč republikou, která jsou naprosto nezávislá na systému EduARd. Tato sekce se stručně věnuje právě těmto řešením.

■ 2.3.1 Původní aplikace

Původní aplikace se potýkala s několika nedostatky. Zejména design, který sám o sobě není špatný, ale je v dnešní době poněkud zastaralý. Dále je samozřejmě zcela zásadní problém, a to již zmíněná možnost běhu pouze na zařízeních s operačním systémem Android.

■ Zastaralý design

Původní aplikace, i když esteticky působí dobře, je po designové stránce dnes poměrně zastaralá a to v oblasti rozložení prvků na obrazovce, uživatelského zážitku a i konkrétních komponent.



(a) : Úvodní obrazovka

(b) : Seznam učebnic

Obrázek 2.2: Tmavý a světlý režim seznamu úloh.

Na obrázku 2.2a si můžeme povšimnout, že zde máme jednotlivé komponenty seřazené ve vertikální sekci a není snadno rozeznatelné, která část se čemu věnuje. V případě tlačítek ve spodní části vyvstává problém s náročností rozeznat, které je na co určeno a to z důvodu jejich designové totožnosti. Text je dost tmavý a spolu s černým pozadím je obtížnější jej přečíst.

Na obrazovce z obrázku 2.2b může být pro uživatele nepříjemná možnost vybrat více učebnic ke stažení, ale po kliknutí na tlačítko je upozorněn na možnost stahovat pouze jednu učebnici v danou chvíli. Tlačítko je na obrazovce i když není nic vybráno a text na něm je nečitelný, což může vést ke zmatení.

■ Nekompatibilita s dalšími operačními systémy

I když je dle statistik Android zcela dominantním operačním systémem na trhu, iOS má stále podíl, který je třeba brát v potaz a pro něj aplikaci také optimalizovat.

■ 2.3.2 Lesní svět

Obdobným řešením je například Lesní svět [11] fungující na systému naučných stezek a sloužící jako náhrada za tradiční naučné tabule a to primárně v přírodě. Rozdílem je pak využívání QR kódů, z kterých aplikace získává data, jež poté zobrazuje, což může být nevýhodou především vzhledem k vandalismu. Lesní svět necílí na vzdělávací instituce, ale především na instituce přímo spjaté s přírodou, jako například národní parky, rezervace a další.

■ 2.3.3 Naučná stezka hajného Kázka

Naučná stezka hajného Kázka [13] je aplikace rovněž založena na QR kódech a slouží jako rozšíření tradičních naučných tabulí. Umožňuje přidat různorodé fotografie, videa nebo zvukové nahrávky. Pro zdatnější publikum je zde také možnost odborných článků. Toto řešení znovu naráží na problém vandalismu.

■ 2.3.4 Vyhodnocení

Více či méně podobná řešení existují, ovšem jsou většinou zastaralé, nejsou multiplatformní a obecně závisí na dalších prvcích, jako například QR kódech a tudíž s sebou přináší mnohá úskalí. Tyto aplikace jsou rovněž mnohdy navrženy pro jednu konkrétní stezku a neumožňují korekci dat, či jakékoli rozšíření nebo úpravy. Původní aplikace trpí také mnohými nedostatky. Všechna řešení sdílí problematický design, který neodpovídá standardům a není jednotný. Z problémů, které byly nalezeny vznikly požadavky na nové řešení.

■ 2.4 Požadavky na nové řešení

Aplikace vznikající v rámci této bakalářské práce má za účel nahradit předchozí aplikaci navrženou pouze pro operační systém Android a hned z úvodu tedy vyplývají požadavky, jež předchozí aplikace vzhledem k zvolené technologii nebyla schopna zpracovat a to zejména multiplatformnost a design. Dále jsou zde další požadavky, které byly objeveny až během analýzy.

Zcela zásadní je jednoduchost a snadná rozšiřitelnost aplikace, aby mohla cílit na pokud možno co největší finální skupinu. Toto je jeden z hlavních

důvodu zvolení frameworku Flutter a napsání jedné aplikace, která poběží na jakémkoli zařízení.

Vzhledem k principu, na kterém má aplikace fungovat, je nutností, aby byla schopna fungovat off-line a proto je třeba materiály ke stezkám stahovat. S tímto ale nastává problém se správou uživatelů a jejich dat, jelikož bez připojení k internetu nelze kontrolovat validitu přihlašovacích údajů. Řešením by se zdánlivě mohlo jevit ukládání uživatelských údajů přímo v zařízení, z toho vyvstává bezpečnostní riziko, které by mohlo být zneužito a ve výsledku mohlo způsobit kolize hesel, pokud by se na serveru vyskytovalo jiné než v zařízení. Vybraným řešením je zpřístupnění přihlašování a odhlašování pouze při připojení zařízení k internetu, tudíž vznikají jakési dva režimy běhu aplikace, a to klasická on-line verze a off-line verze, která rovnou po zapnutí ukazuje učebnice pro naposledy přihlášeného uživatele.

Vzhledem k faktu, že systém budou s vysokou pravděpodobností využívat rovněž mladší děti, je třeba věnovat větší množství pozornosti uživatelskému prostředí, jeho jednoduchosti a intuitivnosti. Z tohoto důvodu byla při návrhu designu snaha minimalizovat množství komponent na jednotlivých obrazovkách pro zvýšení přehlednosti a čitelnosti. Z hlediska designu byl vybrán dnes velmi populární material design [12].

Konkrétní požadavky jsou následující:

■ Funkční

- Přihlášení
- Výběr učebnice
- Stažení učebnice
- Zobrazení průběhu stahování
- Zahájení učebnicí
- Zobrazení úvodní obrazovky učebnice
- Zobrazení úlohy
- Dokončení úlohy i celé učebnice
- Vyhodnocení úlohy
- Zobrazení statistiky
- Vymazání výsledků
- Smazání učebnice
- Předávání dat AR komponentě

■ Nefunkční

- Výkon
- Uživatelsky přívětivé rozhraní
- Multiplatformnost
- Off-line režim
- Bezpečnost
- Stabilita

Kapitola 3

Návrh

Kapitola Návrh se věnuje vytvořenému prototypu a přístupu k rozšířené realitě. Rovněž stručně zmiňuje přístup k návrhu mobilního řešení v rámci systému.

3.1 Mobilní aplikace

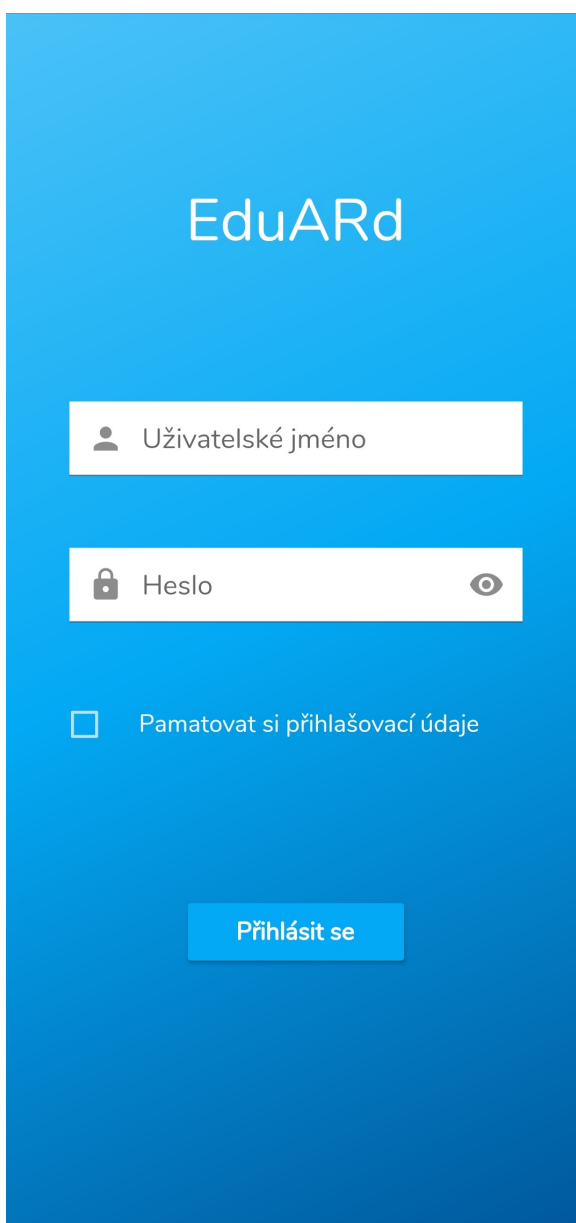
Celý koncept konzumace dat ze serverové části v mobilní aplikaci spočívá v zobrazování 2D obsahu a 3D obsahu. Google má snahu přidat možnosti rozšířené reality do Flutter frameworku, jak je popsáno v několika článcích jako například *Flutter and AR* [5]. Ovšem toto řešení stále není dostatečně funkční a proto bylo zachováno rozdělení do dvou celků a to řídicí aplikace vzniklé v rámci této práce, která přebírá data od serverové části a zobrazuje veškerý 2D obsah, jakožto obrázky, videa, text a další. 3D obsah je pak vykreslován pomocí aplikace vytvořené v rámci práce Dominika Truonga[4].

3.2 Prototyp

Klikatelný prototyp byl vytvořen v moderním designovém nástroji Figma [7]. Drží se definovaných požadavků na jednoduchost a přehlednost. Navrhován je primárně v duchu material designu.

3.2.1 Přihlášení

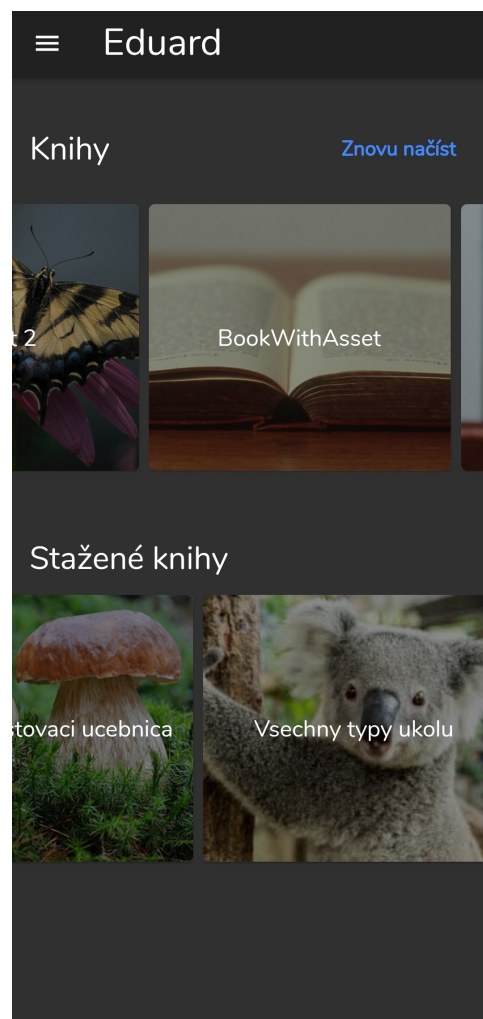
Pro autentizaci uživatele je třeba se přihlásit, a to tradičně pomocí uživatelského jména a hesla, jak je možné vidět na obrázku 3.1. Následně dojde k ověření identity na serveru. Tato možnost není dostupná v případě, že je zařízení bez přístupu k internetu. Přihlašování nevyužívá žádnou třetí stranu, ale všechna data jsou zpracovávána na serverové části.

The image shows a login interface for 'EduARd'. It features a blue gradient background. At the top center, the text 'EduARd' is displayed in white. Below this, there are two white input fields. The first field is labeled 'Uživatelské jméno' (Username) and has a person icon on the left. The second field is labeled 'Heslo' (Password) and has a lock icon on the left and an eye icon on the right. Below the password field, there is a checkbox labeled 'Pamatovat si přihlašovací údaje' (Remember login details). At the bottom center, there is a blue button with the text 'Přihlásit se' (Log in).

Obrázek 3.1: Přihlášení

3.2.2 Domovská obrazovka

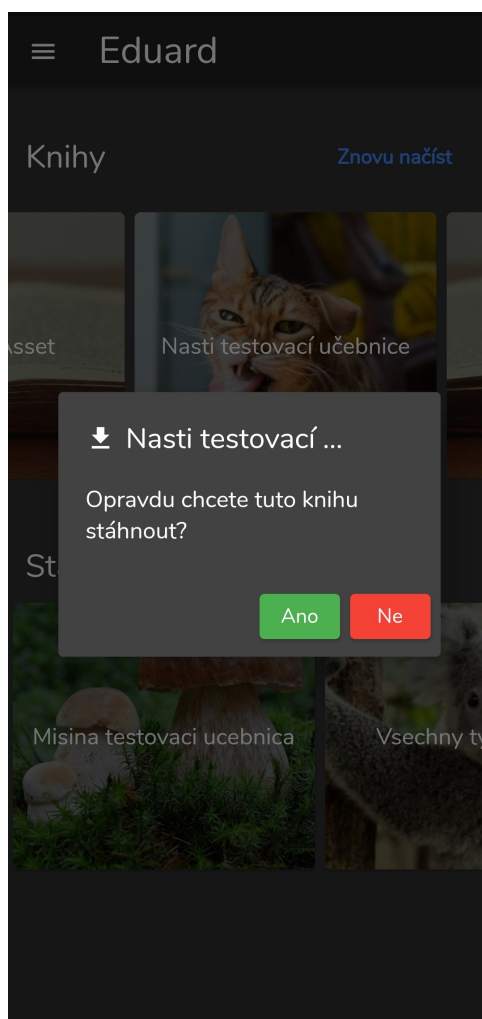
Domovská obrazovka z obrázku 4.1 slouží jako přehled učebnic, a to jak těch stažených, tak i těch, které jsou dostupné ke stažení. Jednotlivé knihy jsou klikatelné karty v horizontálně posuvné sekci. Po kliknutí na kartu se otevře úvodní stránka dané učebnice.



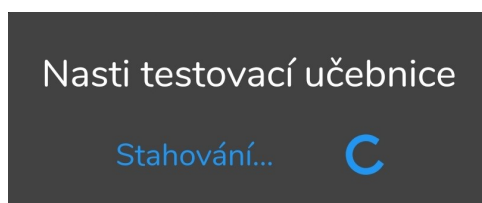
Obrázek 3.2: Domovská obrazovka

3.2.3 Stahování učebnice

Po kliknutí na kartu učebnice následuje zobrazení dialogu s potvrzením snahy o stažení, jak je vidět na obrázku 3.3. Následuje zobrazení upozornění na průběh stahování viz obrázek 3.4.



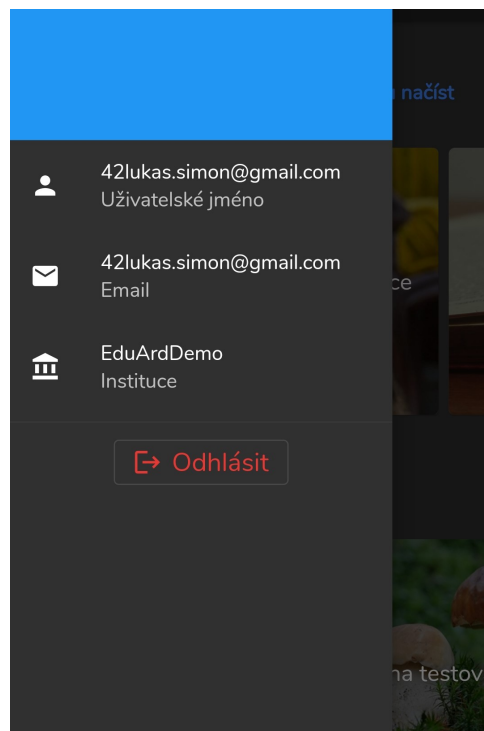
Obrázek 3.3: Stahování



Obrázek 3.4: Průběh stahování

3.2.4 Profil

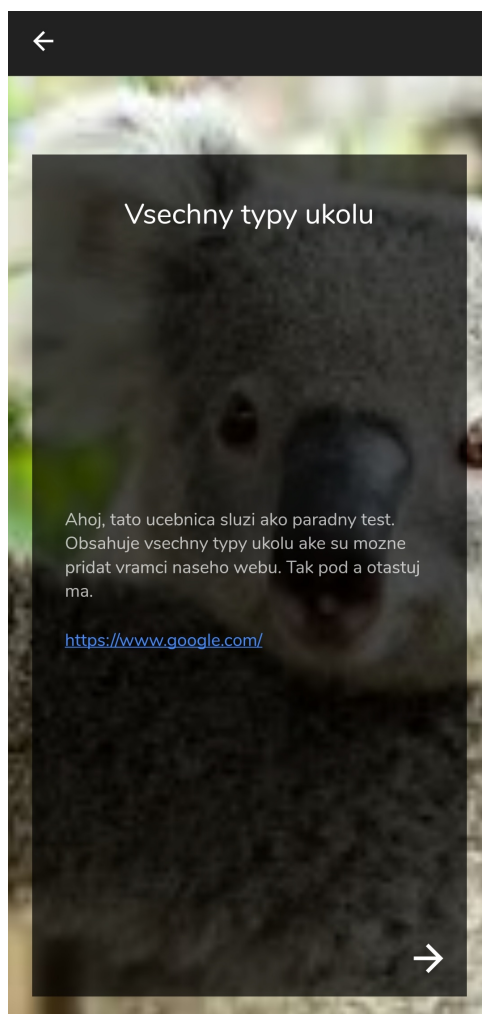
Obrázek 3.5 zobrazuje profil, který je udělán jako výsuvný panel na domovské obrazovce. Nachází se na něm informace o uživateli a to emailová adresa, instituce, role uživatele a jeho jméno. Jednotlivé informace jsou zobrazeny jako položky v horizontální sekci.



Obrázek 3.5: Profil uživatele

■ 3.2.5 Úvod učebnice

Po otevření učebnice se zobrazí úvodní stránka s obrázkem, názvem a případným popiskem. Z této obrazovky už se přesouváme přímo do učebnice. Design se nachází na obrázku 3.6. Pro jednoduchou čitelnost je před obrázek na pozadí umístěn šedý obdélník, na kterém jsou umístěny zmíněné informace a tlačítko pro vstup do učebnice



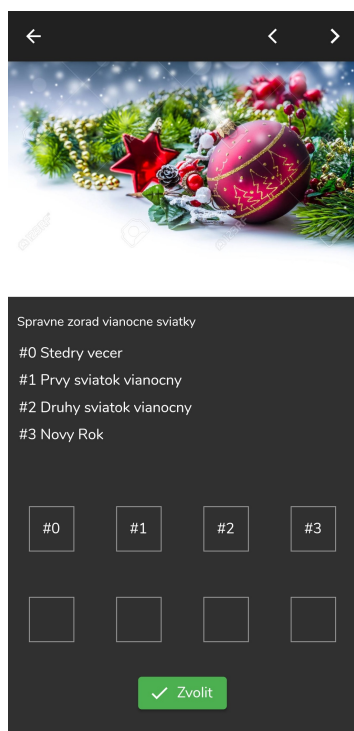
Obrázek 3.6: Úvod učebnice

■ 3.2.6 Úkoly

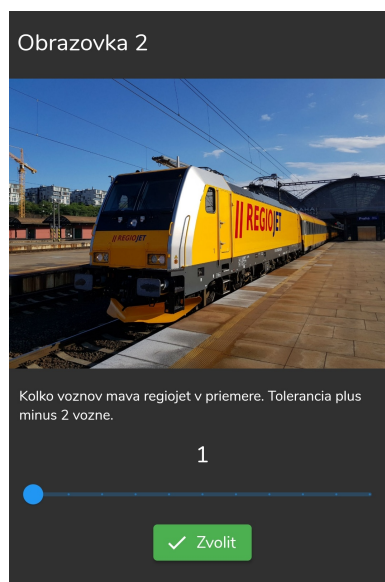
Systém EduARd poskytuje několik typů úkolů pro otestování znalostí. Například se jedná o výběr z výčtu možností, intervalový výběr, textovou odpověď a další viz 2.2.3. Příklady jsou dále na obrázku 3.7. Pro návrat z konkrétního úkolu je v komponentě appbar umístěna šipka zpět. Dále se na obrazovce nachází název úkolu, jeho popis a konečně také část pro odpověď.

■ 3.2.7 Rozložení

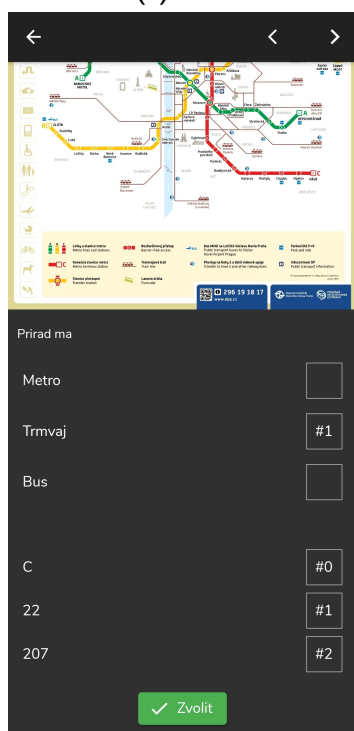
Pro zobrazení všech úkolů v učebnici jsou využity různé typy rozložení dle preference tvůrce viz obrázky 3.8. Pro příjemný uživatelský zážitek je pro přechod mezi sekcemi s mapou a seznamem úkolů použita komponenta tabs, která horizontálně řadí prvky za sebe a umožňuje jejich přepínání po kliknutí na šipku.



(a) : Řazení



(b) : Intervalový výběr



(c) : Spojování

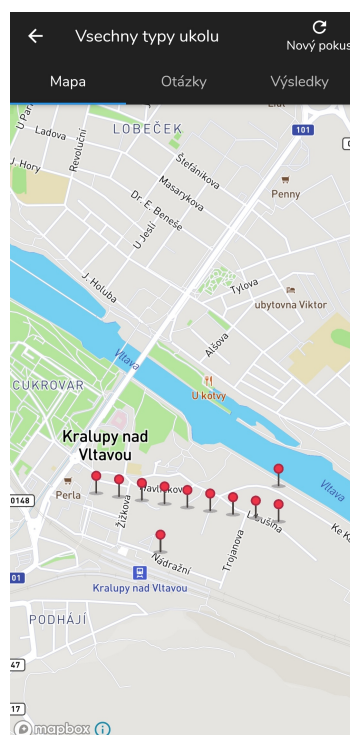


(d) : Textová odpověď

Obrázek 3.7: Ukázky úkolů

Úkol	Úspěšnost
intro	0/0
task1	1/2
task2	0/1
task3	1/1
task4	0/1
task5	0/1
task6	0/1
task7	1/1
task8	0/1
task9	0/1
task10	0/0

(a) : Seznam



(b) : Mapa



(c) : Statistika

Obrázek 3.8: Ukázky rozložení

Kapitola 4

Implementace

Tato kapitola se věnuje především popisu frameworku Flutter, architektuře aplikace, načítání a zpracování dat a také implementaci uživatelského rozhraní. Několik sekcí je věnováno správě stavu, směřování a komunikaci se serverem.

4.1 Flutter

Flutter je moderní framework pro tvorbu skutečně multiplatformních aplikací, které jsou však výkonem na úrovni aplikací nativních.[8] Tohoto dosahuje díky kompilaci do nativního ARM kódu a v případě webových stránek do JavaScriptu. Tyto aplikace mají možnost využít ohromného množství vizuálních komponent, které už od základů působí moderně a esteticky příjemně. Mnoho z nich umožňuje přidání animací bez většího množství dalšího kódu.

Kvalitně vypadající a svižné aplikace nejsou jedinou výhodou frameworku. Flutter používá jazyk Dart. Ten byl vyvinutý společností Google a dostává neustále nové funkcionality. Pro navrhování uživatelských rozhraní byl Dart přímo vytvořen. Od verze 2.12 rovněž poskytuje podporu pro tzv. null safety, jež umožňuje kontrolu nedefinovaných odkazů na objekty.

Vyvstává zde mnoho výhod i pro příjemný a rychlý vývoj. Za zmínku jistě stojí funkce pro překreslení uživatelské rozhraní bez ztráty stavu, či nutnosti znovu celý kód zdlouhavě kompilovat. Flutter rovněž, díky své aktivní komunitě, nabízí nepřeberné množství knihoven, ať už od jednotlivých vývojářů, tak i od velkých korporací jako jsou Google, Microsoft, Canonical, BMW a mnoho dalších.

4.2 Architektura

Momentálně se v rámci Flutteru hojně používají různé typy architektur, avšak každá aktuálně přináší určitá úskalí. Z tohoto důvodu byla snaha nedržet se striktně žádného paradigmatu, ale naopak přizpůsobit se potřebám tohoto řešení. Částečně se aplikace drží vrstevné architektury a určité principy přebírá rovněž z architektury Model-View-Controller (MVC).

Vytvořeny byly prvky uživatelského rozhraní obsažené v modulu *UI*, tento modul je oddělen od získávání dat ze serveru a využívá pouze metody jednotlivých kontrolérů. Vrstva kontrolérů obstarává logiku potřebnou pro předávání a ukládání dat, volání jednotlivých metod na API vrstvě a konečně také přepínání obrazovek. Pro volání metod na serveru slouží API vrstva obsažená v modulu *API*, velká část kódu pro tuto vrstvu byla vygenerována díky využití knihovny Retrofit.

Pro snížení duplikace kódu byly využity tzv. extension metody [6], které jazyk Dart poskytuje a které umožňují přidat funkcionalitu na již existující třídy. Příkladem by mohlo být využití na postupnou animaci prvků na obrazovce pomocí extension metody na třídě *Widget* obsažené v modulu *extensions*.

4.3 Správa stavu

Správa stavu (State Management) je v komunitě Flutteru bez konkurence jedno z nejvášnivěji probíraných témat a způsobů, jak jej uchopit, je opravdu velké množství. Pro webové vývojáře známý Redux, případně Fish-Redux, dále BLoC, MobX, GetIt a mnoho dalších. Největší popularitě se ale těší Provider a Riverpod, oba od stejného tvůrce. Těm se bude práce dále věnovat. Riverpod je rovněž řešení využité v rámci této práce.

Správou stavu je myšleno ukládání dat při běhu aplikace a možnost tato data sdílet mezi nezávislými komponentami a obrazovkami. Například pokud uživatel klikne na knihu, tak obrazovka s knihou potřebuje vědět, na kterou učebnici uživatel klikl.

4.3.1 Provider

Provider [15] jakožto doporučený přístup si získal popularitu zejména svou jednoduchostí užití, čitelností kódu a nepotřebností psát nadbytečný kód, jako tomu je u mnohých výše zmíněných řešení. Velkou výhodou je, že nevychází z principů převzatých z jiných frameworků či jazyků, ale byl navržen přímo pro Flutter a ve své podstatě jen rozšiřuje funkcionalitu nativních komponent frameworku.

Velkou nevýhodou Provideru je jeho závislost na třídě `BuildContext`, která dělá práci s ním mimo komponenty uživatelského rozhraní nepřijemnou. Provider také není tzv. `compile safe` a snadno se tedy může stát, že aplikace přestane fungovat i přesto, že kód byl zkompilován bez potíží.

4.3.2 Riverpod

Riverpod [16] pochází od stejného tvůrce jako Provider a v mnohém jsou si oba přístupy podobné, avšak Riverpod přichází s výraznými vylepšeními. Předně není závislý na žádné jiné knihovně ani Flutteru samotném, což efektivně řeší problém s třídou `BuildContext`. Riverpod je také `compile safe`, důsledkem čehož dává vývojáři jistotu, že jakmile je kód zkompilován, tak jistě nepřestane fungovat přičiněním Riverpodu.

V rámci této práce byl z knihovny Riverpod použit *State Notifier Provider*, který umožňuje příjemnou práci s daty, například se seznamem textových řetězců. Při jakékoliv změně je automaticky každý vizuální prvek závisící na tomto datovém zdroji upozorněn na změnu. Na straně uživatelského rozhraní se o upozornění stará třída *Consumer*, která předává data dále a vizuální prvky překreslí. Pro třídy starající se o stav aplikace byl vytvořen modul *providers*, který obsahuje jednak obecné providery, jako například *stream-ConnectivityProvider*, který všechny poslouchající komponenty upozorňuje na změnu připojení k internetu. Uvnitř modulu jsou také jednotlivé repozitáře, které obsahují datové zdroje jako stažené knihy, aktuálně dostupné knihy a další.

4.4 Směrování

Směrování (Routing) slouží pro pohyb mezi obrazovkami v aplikaci a to použitím zásobníku, na který se obrazovky ukládají a po vrácení zpět se z něj rovněž odebírají.

Pro směrování v aplikaci je možné využít řešení frameworku. To poskytuje třídu Navigator, která umožňuje dostatek funkcionality, ovšem vede k opakování kódu, což je obecně bráno jako špatný přístup. Flutter později přišel s druhou verzí třídy Navigator, ta se soustředí primárně na lepší podporu pro web a opakování kódu je stále problém.

Jak je u Flutteru zvykem i pro navigaci je zde několik knihoven, ovšem pro snazší úpravy a snížení závislosti na knihovnách byl v rámci práce navržen odlehčený přístup, který veškerou navigaci přesouvá do jedné implementované třídy *Routes*. Umožňuje tak jednodušší správu a zamezuje zbytečnému opakování stejného kódu.

Třída *Routes* požaduje pouze dvojici navigovatelné stránky a jejího názvu, zbytek práce automaticky dělá sama. Animace přesunu na další stranu je nastavena na posun z boku, avšak dá se jednoduše změnit v metodě *navigate*, tento fakt zaručuje jednotnost aplikace a dopomáhá její celkové konzistenci. Rovněž je zadefinován widget, který je zobrazen v případě, že vývojář zapomene stránku registrovat, ale pokusí se na ni přejít. Třída *Routes* se nachází v modulu *utils* a je dostupná pro všechny komponenty.

4.5 Data

Serverová část systému Eduard poskytuje většinu dat ve formátu JSON. Ten má díky své popularitě mnoho způsobů, jak data převést na třídy v jazyce Dart, již připravených v rámci frameworku. Jednotlivé učebnice jsou však v již méně podporovaném formátu XML, jehož zpracování bylo poněkud obtížnější.

4.5.1 JSON

JSON je formát, který má ve frameworku Flutter kvalitní podporu skrze funkce `jsonEncode` a `jsonDecode`, efektivně jej přeměňující na páry klíč a

hodnota. Dart poté poskytuje vývojářům factory metody, které více usnadňují přetvoření dat z formátu JSON do tříd definovaných vývojářem. Tato metoda přijme parametry a předá konkrétní atributy konstruktoru, načež vrátí konkrétní instanci.

■ 4.5.2 XML

XML formát je ve Flutteru o poznání méně podporován a pro jeho parsování byla využita knihovna `xml`[17]. Pro převod celého dokumentu na třídy byly znovu využity factory metody, zpřehledňující kód.

■ 4.5.3 Ukládání dat

Pro ukládání dat byla zvolena knihovna Hive, která poskytuje Key-value databázi napsanou přímo v programovacím jazyce Dart. Důvodem jejího zvolení je skvělý výkon, intuitivnost a jednoduchost. Hive nemá žádné závislosti na dalších knihovnách a má v sobě zabudované šifrování dat. Konkurenty by mohly být Shared preferences případně SQLite ovšem Hive je nejen rychlejší, jak můžeme vidět na benchmarku dostupném v dokumentaci[10], ale je i jednodušší na použití. Hive je využit při ukládání dat o knize a to v provideru `bookSaveProvider`. Kniha je ukládána jako třídy v jazyce Dart, které po označení anotací `@HiveType` a označení jednotlivých atributů `@HiveField`, jsou schopny generace adaptérů. Ty dokáží třídy serializovat.

■ 4.6 Implementace uživatelského rozhraní

Implementace uživatelského rozhraní proběhla primárně pomocí základních vizuálních prvků Flutteru, avšak pro zobrazení mapy bylo třeba využít knihovnu `MapBox` 4.6.2.

■ 4.6.1 Google fonts

Flutter sám poskytuje množství fontů hned po instalaci, ale Google pro vývojáře udržuje knihovnu Google Fonts [9], která obsahuje vysoký počet kvalitních fontů a jejich použití v aplikaci je velmi intuitivní a dává další možnosti. Rovněž snižuje velikost instalačního souboru oproti přímému stažení fontů. V aplikaci byl vybrán font s názvem Nunito a byl nastaven v třídě *styles*, která dodává styly nejen pro text, ale i barvy, ohraničení elementů a mnoho dalšího.

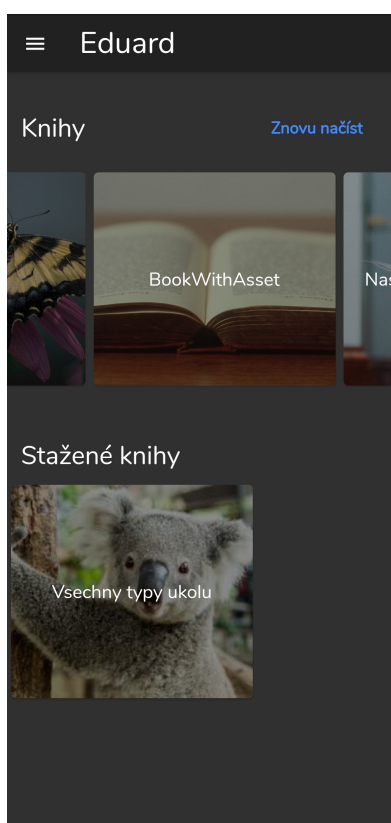
■ 4.6.2 MapBox

Pro práci s mapou existuje mnoho řešení. Mezi ty velmi populární se dají zařadit Google maps, které poskytují velké množství funkcionalit. Mají však přinejmenším jednu nepříjemnost, a to žádnou podporu pro off-line užití. Jelikož je jedním z požadavků běh aplikace bez internetu, byla využita knihovna MapBox. Její integrace ve Flutteru funguje velmi solidně a poskytuje všechnu nezbytnou funkcionalitu pro správné fungování aplikace, a to zobrazení jednotlivých bodů zájmu na mapě, zoomování pomocí gest a přesun po mapě. Mapbox byl zakomponován do aplikace jako jedna z obrazovek při otevření knihy, a to v třídě *QuestionMap*. Ta se stará o vykreslení mapy a přidání značek. Když uživatel klikne na značku, otevře se obrazovka s úkoly. Pozice značky je definována u každého úkolu, který je zobrazitelný na mapě.

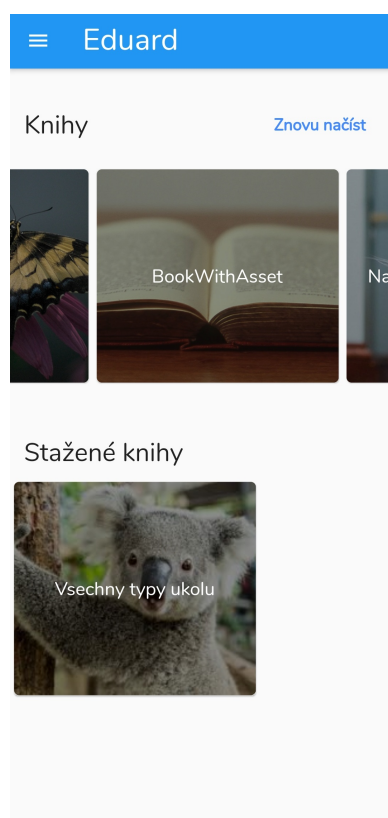
■ 4.6.3 Tmavý režim

I když práce s tmavým a světlým režimem nebyla přímo zmíněna v zadání, pro kvalitní moderní aplikace je jejich přítomnost vhodná. Flutter poskytuje příjemnou funkcionalitu zjištění režimu nastaveného v systému. Když je mu poskytnutý jak světlý, tak i tmavý režim, automaticky mezi nimi přepíná dle preferencí uživatele.

Při tvorbě aplikace bylo dbáno na jednoduchost uživatelského rozhraní a i v této problematice byla snaha se držet material designu.



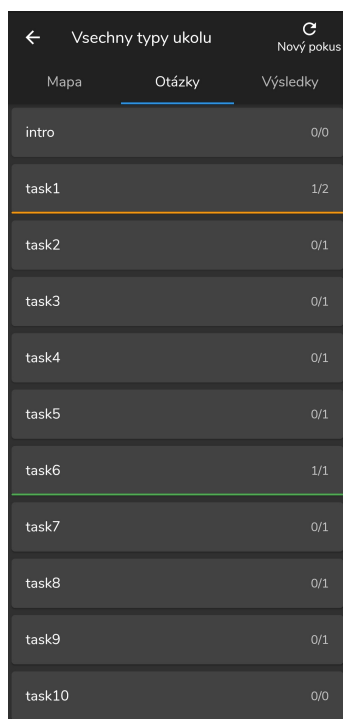
(a) : Tmavý režim



(b) : Světlý režim

Obrázek 4.1: Tmavý a světlý režim domovské obrazovky.

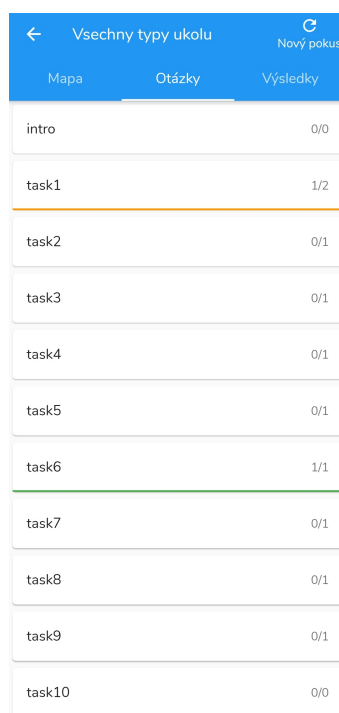
4. Implementace



Dark mode screenshot of the task list. The header is dark grey with a back arrow, the text 'Všechny typy ukolu', and a refresh icon labeled 'Nový pokus'. The task list is on a dark background with white text. The 'Otázky' tab is selected.

Task	Progress
intro	0/0
task1	1/2
task2	0/1
task3	0/1
task4	0/1
task5	0/1
task6	1/1
task7	0/1
task8	0/1
task9	0/1
task10	0/0

(a) : Tmavý režim



Light mode screenshot of the task list. The header is light blue with a back arrow, the text 'Všechny typy ukolu', and a refresh icon labeled 'Nový pokus'. The task list is on a white background with blue text. The 'Otázky' tab is selected.

Task	Progress
intro	0/0
task1	1/2
task2	0/1
task3	0/1
task4	0/1
task5	0/1
task6	1/1
task7	0/1
task8	0/1
task9	0/1
task10	0/0

(b) : Světlý režim

Obrázek 4.2: Tmavý a světlý režim seznamu úloh.



(a) : Tmavý režim



(b) : Světlý režim

Obrázek 4.3: Tmavý a světlý režim statistiky.

4.7 Propojení s komponentou rozšířené reality

Pro komunikaci s komponentou rozšířené reality bylo původně zamýšleno využít knihovny pro spouštění dalších aplikací, ovšem v průběhu implementace bylo zjištěno, že tato knihovna neumožňuje předávat parametry, což je nezbytné pro správné fungování.

Nad rámec zadání byl tedy vytvořen modul pro provolávání nativních metod a metody samotné, které se starají o spuštění aplikace rozšířené reality a předání potřebných argumentů spolu s funkcí, kterou je nutno provést.

Aplikace vytvořená v rámci této práce tedy dostane informace od serverové části, ty vyhodnotí a posílá vytvořeným kanálem do nativní části, kde se dle dostupných parametrů spustí aplikace rozšířené reality. Dále se jí zadá povel pro vykonání konkrétní funkce a předají se jí parametry pro to potřebné.

4.7.1 Komunikace se serverem

Komunikace se serverem je zjednodušena knihovnou Retrofit. Ta jako stejnojmenná knihovna pro vývoj v systému android, pomáhá mapovat jednotlivé koncové body API (API Endpoints). Tato knihovna poskytuje generování kódu a je potřeba pouze zdefinovat rozhraní s návratovými typy, parametry a URL. Konkrétní implementace metod je následně automaticky vygenerována.

Toto je využito v modulu *API*, kde jsou jednotlivé metody jen definovány a následně vytvořeny, jak je popsáno výše. Pro autentizaci je použit token, který je přidáván k jednotlivým voláním pomocí třídy *RequestInterceptor*. Ta zachytí jednotlivé požadavky a token k nim přidá. Pro získání dat zavolá kontrolér potřebnou metodu na třídě *RestClient*, která předá parametry vygenerované třídě a ta vykoná konečný požadavek na server.

Kapitola 5

Testování

Tato kapitola se věnuje testování, které je důležitou součástí vývoje systému. Framework Flutter umožňuje několik způsobů automatického testování, avšak pro kontrolu celkového konceptu aplikace je značně výhodné provést testování uživatelské. Vzhledem k tomuto byla vybrána kombinace unit testů pro důležité vnitřní funkcionality a uživatelských testů pro kontrolu celkové stability a rovněž kontrolu, že uživatelské rozhraní je dostatečně srozumitelné.

5.1 Automatické testování

Automatické testování bývá u mobilních aplikací často vynecháno, a to mnohdy z důvodů pouhého konzumování dat ze serveru. Aplikace, jež byla vytvořena s touto prací je rovněž primárně pouze konzument dat, ovšem pro větší komplexnost řešení bylo automatické testování také zahrnuto.

5.1.1 Unit testy

Unit testy jsou obecně užívány pro testování malých celků, jako například jednotlivých funkcí, metod či tříd krajními hodnotami a mohou efektivně dopomoci k odchytní hraničních případů. Jsou také vhodné pro kontrolu nových funkcionalit, které by jinak mohly poškodit již funkční komponenty. Pro Flutter je typické mít testy v sekci oddělené od ostatního kódu. V rámci práce bylo vytvořeno několik sad unit testů pokrývajících většinu výpočtů a v

průběhu vývoje byly pravidelně spouštěny pro kontrolu. Žádné chyby však v rámci unit testování nalezeny nebyly.

Flutter má pro unit testy knihovnu, která poskytuje mnoho užitečných funkcionalit.

- `test()` - funkce pro jeden konkrétní test
- `group()` - funkce pro seskupení několika testů do jedné skupiny
- `expect()` - funkce pro potvrzení výsledku testu
- `matcher` - typ funkcí pro komplexnější porovnání, například `isNot()`, `endsWith()` a podobné
- `throwsA()` - funkce na testování výjimek

5.2 Uživatelské testování

Testy provedené přímo uživateli jsou velmi efektivní cestou odhalení chyb v systému a jsou populární zejména ve vývoji mobilních aplikací. Zařízení se dnes dají simulovat, ovšem test na fyzickém přístroji má často k realitě blíže. Rovněž testování uživateli, pokud je provedeno korektně, bude s vysokou pravděpodobností více objektivní než automatické testy napsané vývojářem daného softwaru.

Pro tuto práci bylo vybráno vzhledem k stále trvající pandemii 5 testerů v rozmezí od 17 do 24 let s 5 odlišnými zařízeními, technickou zdatností, velikostí obrazovek a verzemi operačního systému. Všem testerům bylo poskytnuto několik požadovaných průchodů aplikací, dále popsanych v sekci průchody aplikací 5.2.2.

5.2.1 Distribuce účastníkům testování

Pro distribuci účastníkům byla využita platforma Firebase a její služba App Distribution. Tato služba umožňuje nahrát instalační soubor a rozeslat jej uživatelům. Pro tyto účely existuje i mobilní aplikace, která po přijetí pozvánky dává možnost reinstalovat jednotlivé verze. Uživatelé mohou být dále rozděleni do skupin dle preferencí správce.

Celý proces není obtížný, ovšem pro úsporu času byl i přesto automatizován. Pro tento účel byl vytvořen jednoduchý skript, který projekt zkompile, nahraje jej do služby App Distribution a vybere uživatele, případně skupiny, kterým instalační soubor poslat. Je zde rovněž možnost přidat poznámky k nové verzi a zmíněný skript této skutečnosti rovněž využívá.

■ 5.2.2 Průchody aplikací

Pro nasměrování uživatelů na využití co nejvíce funkcionalit bylo vytvořeno několik průchodů aplikací. Ty se liší složitostí a byly navrženy, aby na sebe lineárně navazovaly a i pro velmi málo technicky zdatné uživatele byly snadno pochopitelné.

■ První průchod

První průchod byl navržen pro seznámení testera s uživatelským rozhraním a otestováním základní funkcionality.

1. Přihlášení - Přihlaste se do aplikace pomocí poskytnutých údajů.
2. Výběr učebnice - Vyberte učebnici a stáhněte ji do zařízení.
3. Spuštění učebnice - Vyberte učebnici ze stažených učebnic, spusťte ji a projděte skrz úvodní stránku do učebnice.
4. Spuštění úkolu - Vyberte úkol a spusťte ho.
5. Prolistování úkolu - Projděte všechny obrazovky úkolu a vraťte se zpět do učebnice.
6. Návrat a smazání učebnice - Vraťte se zpět na domovskou obrazovku a smažte staženou učebnici.
7. Zobrazení profilu a odhlášení - Zobrazte profilové informace a odhlaste se.

■ Druhý průchod

Druhý průchod byl navržen pro kontrolu úkolů. Pro vyšší pokrytí byla vybrána učebnice obsahující všechny typy úkolů.

1. Přihlášení - Přihlaste se do aplikace pomocí poskytnutých údajů.
2. Výběr učebnice - Vyberte zadanou učebnici a stáhněte ji do zařízení.
3. Spuštění učebnice - Vyberte staženou učebnici, spusťte ji a projděte skrz úvodní stránku do učebnice.
4. Průchod úkoly - Přejděte na seznam úkolů, postupně zvolte 3 a dle nejlepšího uvážení vyplňte otázky.
5. Průchod úkoly - Dle bodu 4 vyberte další 3 úkoly odlišné od zvolených v bodě 4 a dle znalostí se pokuste zodpovědět záměrně nesprávně.
6. Kontrola stavu - Vraťte se zpět na domovskou obrazovku a poté znovu otevřete učebnici.
7. Návrat a odhlášení - Vraťte se zpět na domovskou obrazovku a odhlaste se.

Třetí průchod

Třetí průchod byl navržen pro kontrolu ukládání dat. Pro vyšší pokrytí byla vybrána učebnice obsahující všechny typy úkolů.

1. Přihlášení - Přihlaste se do aplikace pomocí poskytnutých údajů.
2. Výběr učebnice - Vyberte zadanou učebnici a stáhněte ji do zařízení.
3. Spuštění učebnice - Vyberte staženou učebnici, spusťte ji a projděte skrz úvodní stránku do učebnice.
4. Průchod úkoly - Přejděte na seznam úkolů a postupně zvolte 3 a vyplňte otázky.
5. Návrat a odhlášení - Vraťte se zpět na domovskou obrazovku a odhlaste se.
6. Vypnutí aplikace - Vypněte aplikaci a ujistěte se, že neběží ani na pozadí.
7. Zapnutí a návrat do učebnice - Zapněte znovu aplikaci a dle bodu 1 se přihlaste a znovu otevřete učebnici z bodu 2.
8. Kontrola dat - Přejděte na seznam úkolů a zkontrolujte, že je zelené zvýraznění stále přítomno u vyplněných úkolů.
9. Resetování učebnice - Zahajte nový pokus učebnice a zkontrolujte na obrazovce vyhodnocení, že žádné otázky nejsou zodpovězeny a počet pokusů byl navýšen.
10. Návrat a odhlášení - Vraťte se zpět na domovskou obrazovku a odhlaste se.

■ Volný průchod aplikací

Po otestování průchodů dostali testeři možnost aplikaci testovat dle vlastního uvážení a případně se pokusit ji i záměrně učinit nefunkční či ji jakýmkoliv způsobem poškodit.

■ 5.2.3 Výsledky uživatelského testování

Testy proběhly dle očekávání a bylo odhaleno několik chyb a nepříjemností, které byly následně vyřešeny. Mimo problém s odhlášením bez připojení k internetu se nepodařilo žádnému testerovi aplikaci učinit nefunkční a vše fungovalo správně. I když ostatní chyby a nepříjemnosti neměly výraznější vliv na použitelnost aplikace, byly samozřejmě rovněž odstraněny.

■ Odhalené chyby

Problém	Řešení
Uživatel se i bez připojení k internetu mohl odhlásit bez obtíží	Nyní je uživatel upozorněn, že to není bez připojení k internetu možné
Bez připojení k internetu aplikace přestala fungovat po kliknutí na přihlášení	Vyřešeno zamezením odhlášení bez připojení k internetu, chyba způsobena null hodnotou, přidán blok pro odchyčení
Uživatel očekával potvrzení stažení pro kliknutí omylem	Potvrzovací dialog přidán před začátkem stahování
Uživatelé zaznamenali hlášení aplikace s problémem stažení jedné z učebnic a tuto skutečnost nahlásili jako chybu	Toto ve skutečnosti není chyba v aplikaci, ale správné hlášení, že server odmítl poskytnout k učebnici data, byla změněna barva pro zdůraznění
Při zadání špatných uživatelských údajů bylo chybové hlášení málo čitelné	Změna barvy na více kontrastní odstín

Tabulka 5.1: Zadání nového úkolu



Kapitola 6

Závěr

V rámci první části práce byla provedena kompletní analýza systému, jeho jednotlivých komponent a celkové struktury. Zvýšená pozornost byla v analytické sekci věnována datovému modelu a jeho částem. Dále byl vytvořen interaktivní prototyp pro demonstraci designu a uživatelsky orientovaného ovládání. Z předchozí aplikace a zadání vyplynuly požadavky, které byly rovněž zahrnuty v návrhu prototypu.

Vývojová fáze začala implementací požadovaných funkcionalit, a to zejména přihlášení uživatelů a jejich striktní navázání na instituci, a tudíž knih pro ně dostupných. Následně zobrazení knih, jejich stažení, spuštění a případné smazání. Samozřejmě je rovněž zobrazení jednotlivých úloh a jejich průchod včetně vyhodnocování případných otázek. V neposlední řadě bylo třeba vyvinout modul pro propojení aplikace s aplikací rozšířené reality[4]. Flutter ani jeho knihovny tuto funkcionalitu v dostatečném provedení neposkytují, tudíž nad rámec zadání byla vytvořena vrstva integrace nativních metod. Rovněž byla pro větší komplexnost a pro uživatelsky příjemnější zážitek implementována možnost tmavého režimu.

V rámci poslední fáze byla aplikace uživatelsky otestována na několika zařízeních a pro vyšší jistotu stability byla částečně pokryta unit testy.

■ 6.1 Budoucnost řešení

Aplikace byla vytvořena se vši požadovanou funkcionalitou a dokonce částmi navíc, ovšem během vývoje byla do Flutteru přidána podpora pro *null safety* [14] posouvající vývoj na novou úroveň. Bohužel v době psaní této práce není migrace možná vzhledem ke knihovnám užitým při implementaci, které momentálně migrovány nejsou.



Bibliografie

- [1] Jan Skála. “Systém pro správu uživatelů projektu EduARd”. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2019.
- [2] Anastasia Surikova. “Webový klient pro správu mobilních výukových úloh systému EduARd”. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2019.
- [3] Michaela Kubišová. “Mobilní Android aplikace pro řešení výukových úloh v systému EduARd”. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2020.
- [4] Dominik Truong. “Georeferencovaná rozšířená realita”. Bakalářská práce. České vysoké učení technické v Praze, Fakulta elektrotechnická, katedra počítačů, 2020.
- [5] *AR*. URL: <https://medium.com/codechai/flutter-and-ar-a65bb202c5d8> (cit. 26.04.2021).
- [6] *extension-methods*. URL: <https://dart.dev/guides/language/extension-methods> (cit. 26.04.2021).
- [7] *Figma*. URL: <https://www.figma.com/> (cit. 04.01.2021).
- [8] *Flutter-performance*. URL: <https://dart.dev/overview#platform> (cit. 26.04.2021).
- [9] *google-fonts*. URL: https://pub.dev/documentation/google_fonts/latest/ (cit. 26.04.2021).
- [10] *Hive*. URL: <https://pub.dev/packages/hive> (cit. 26.04.2021).
- [11] *Lesní svět*. URL: <https://www.lesnisvet.cz/> (cit. 04.01.2021).
- [12] *material-design*. URL: <https://material.io/design> (cit. 26.04.2021).
- [13] *Naučná stezka hajného Kázka*. URL: <https://www.infopanely.cz/aplikace/> (cit. 04.01.2021).

- [14] *null-safety*. URL: <https://dart.dev/null-safety> (cit. 26.04.2021).
- [15] *provider*. URL: <https://pub.dev/documentation/provider/latest/> (cit. 26.04.2021).
- [16] *riverpod*. URL: https://pub.dev/documentation/flutter_riverpod/latest/ (cit. 26.04.2021).
- [17] *xml*. URL: <https://pub.dev/documentation/xml/latest/> (cit. 26.04.2021).



Příloha A

Návod pro instalaci

Pro spuštění aplikace je třeba vytvořit instalační soubor. Pro ten je nezbytný Flutter SDK, pokud na zařízení není dostupný je možné ho snadno získat pomocí postupu dostupného v oficiální dokumentaci¹.

Jakmile je zdrojový kód v zařízení, je třeba v projektové složce spustit terminal případně windows terminal a zadat příkaz *flutter packages pub get*, pro získání všech potřebných knihoven.

Dále stačí zadat příkaz *flutter build apk*, který vytvoří instalační soubory, které jsou poté k nalezení */eduard/build/app/outputs/*.

Poté stačí jen vybrat soubor *app-release.apk*, nainstalovat jej a spustit.

¹Jak nainstalovat flutter sdk - <https://flutter.dev/docs/get-started/install>