



Zadání bakalářské práce

Název:	Frontend webové aplikace na učení se angličtiny
Student:	Jan Jeníček
Vedoucí:	Ing. Jiří Hunka
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2024/2025

Pokyny pro vypracování

Cílem práce je ve spolupráci s Janem Hlaváčem realizovat webovou aplikaci na učení se angličtiny. Z důvodu velkého rozsahu a také aby byla zajištěna forma izolace od práce Jana Hlaváče je předmětem této konkrétní práce frontendová část aplikace a mock koncových bodů REST API.

Postupujte v následujících krocích:

1. Proveďte analýzu konkurenčních řešení s důrazem na frontendovou část, dále také formou, která bude využitelná pro tvorbu funkčních požadavků v práci Jana Hlaváče. Důraz kladte také na analýzu použitelnosti.
2. Na základě analýzy proveďte návrh, včetně vhodné formy uživatelského rozhraní. Dle výsledného návrhu proveďte vhodnou volbu technologií.
3. Implementujte frontend část aplikace a vytvořte jednoduchý mock koncových bodů REST API.
4. Navrhněte a realizujte vhodné formy testů frontend části aplikace.
5. Nasadte svou část práce do produkčního prostředí.
6. Zhodnoťte výsledné řešení, navrhněte úpravy do budoucna.

Bakalářská práce

FRONTEND WEBOVÉ APLIKACE NA UČENÍ SE ANGLIČTINY

Jan Jeníček

Fakulta informačních technologií
Katedra softwarového inženýrství
Vedoucí: Ing. Jiří Hunka
16. května 2024

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2024 Jan Jeníček. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení, je nezbytný souhlas autora.

Odkaz na tuto práci: Jeníček Jan. *Frontend webové aplikace na učení se angličtiny*. Bakalářská práce. České vysoké učení technické v Praze, Fakulta informačních technologií, 2024.

Obsah

Poděkování	v
Prohlášení	vi
Abstrakt	vii
Seznam zkratek	viii
Úvod	1
1 Analýza	2
1.1 Analýza existujících řešení	2
1.1.1 Duolingo	2
1.1.2 Mondly	6
1.1.3 Busuu	8
1.1.4 Babbel	10
1.1.5 Závěr analýzy	13
1.2 Analýza požadavků	15
1.2.1 Funkční požadavky	15
1.2.2 Nefunkční požadavky	15
2 Návrh	17
2.1 Případy užití	17
2.1.1 Popis aktérů	17
2.1.2 Seznam případů užití	18
2.2 Návrh uživatelského rozhraní	38
2.3 Výběr technologií	39
2.3.1 Framework a jazyky	40
2.3.2 Knihovny a další programy	44
3 Implementace a nasazení	47
3.1 Postup při realizaci projektu	47
3.1.1 Příprava realizace projektu	47
3.1.2 Postup při implementaci frontendu	48
3.1.3 Postup při spolupráci s backendem	53
3.2 Nasazení aplikace	54
3.2.1 Vývojové prostředí frontendu	54
3.2.2 Produkční prostředí frontendu	54
4 Testování	56
4.1 Volba vhodných forem testování	56
4.2 Manuální testování	56
4.3 Uživatelské testování	57
4.3.1 Plán testování	57

4.3.2	Testovací scénáře	57
4.3.3	Průběh testování	60
4.3.4	Výsledky testování	60
4.4	Zhodnocení aplikace a návrh úprav do budoucna	61
4.4.1	Zhodnocení aplikace	61
4.4.2	Návrh úprav do budoucna	62
	Závěr	63
	A Výsledky dotazníku před uživatelským testováním	64
	B Výsledky dotazníku SUS	66
	C Výstupy uživatelského testování	68
	Slovník pojmů	75
	Obsah příloh	76

Seznam obrázků

2.1	Hlavní stránka kurzu	39
2.2	Stránka zápisu do kurzu	39
2.3	Poslechové cvičení	40
2.4	Stránka s volbou témat kurzu	40
3.1	Návrh přehledu lekce mluvení	49

Seznam tabulek

2.1	Licence frontend frameworků	41
2.2	Podpora vývoje pro jiné platformy u frontend frameworků	41
2.3	Míra používání frontend frameworků	42
2.4	Počet dostupných knihoven frontend frameworků	42
2.5	Oblíbenost frontend frameworků	43
2.6	Složitost osvojení frameworků	43
2.7	Počet vývojářů ve frameworkích	43
2.8	Celkové bodové ohodnocení frontend frameworků	44
4.1	Vybrání testeři pro uživatelské testování	58
B.1	Odpovědi účastníků dotazníku SUS	67

Seznam výpisů kódu

3.1	Definice odpovědi koncového bodu pro získání dané lekce	50
3.2	Ukázka služby LessonsAPI	51
3.3	Ukázka implementace stránky s přehledem lekce	52

Tímto bych chtěl poděkovat vedoucímu práce Ing. Jiřímu Hunkovi, zejména za čas, který mi věnoval a za velice cennou zpětnou vazbu, kterou mi poskytoval v průběhu celé mé práce. Dále bych chtěl poděkovat kolegovi Janu Hlaváčovi za jeho trpělivost a jeho výbornou práci, díky které se nám podařilo vytvořit funkční aplikaci. Poděkovat bych chtěl také všem mým přátelům, kteří mě podporují a motivují mě k podávání lepších výsledků při studiu. Na závěr bych také rád poděkoval mé rodině, která mě neustále podporuje v průběhu mých studií.

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.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 16. května 2024

Abstrakt

Hlavním cílem této práce bylo vytvořit plně funkční frontend webové aplikace na učení se angličtiny. Práce zahrnuje kompletní proces vývoje softwaru včetně analýzy, návrhu, implementace, testování a následného nasazení aplikace do produkčního prostředí. Jako součást práce byl vytvořen mock REST API, který aplikace využívá ve vývojovém prostředí. V produkčním prostředí pak aplikace využívá reálné REST API.

Nejprve byla provedena analýza se zaměřením na existující aplikace. Následně byl v programu Figma vytvořen návrh uživatelského rozhraní. V rámci implementace pak byly využity technologie React, Next.js a TypeScript. Ve fázi testování bylo provedeno zejména uživatelské testování použitelnosti, které odhalilo několik příležitostí pro zlepšení aplikace. Celkově je výsledkem práce nasazený a funkční frontend aplikace. Uživatelé se tak mohou díky lekcím a cvičením v aplikaci zdokonalovat ve většině aspektech anglického jazyka včetně gramatiky, slovní zásoby, výslovnosti, poslechu a čtení.

Klíčová slova webová aplikace, frontend, učení se angličtiny, React, Next.js, TypeScript

Abstract

The main goal of this thesis was to create a fully functional frontend of a web application for learning English. The thesis includes the complete software development process, including analysis, design, implementation, testing and subsequent deployment of the application into a production environment. As part of the work, a mock REST API was created, which the application uses in the development environment. In the production environment, the application utilizes a real REST API.

Firstly, an analysis was conducted focusing on existing applications. Subsequently, a user interface design was created in the program Figma. During the implementation phase, React, Next.js, and TypeScript technologies were utilized. In the testing phase, usability testing was mainly performed, which revealed several opportunities for improving the application. Overall, the result of the work is a deployed and functional frontend of the application. Through the lessons and exercises in the application, users can improve in most aspects of the English language, including grammar, vocabulary, pronunciation, listening, and reading.

Keywords web application, frontend, English learning, React, Next.js, TypeScript

Seznam zkratek

API	Application Programming Interface
REST	Representational State Transfer
SEO	Search Engine Optimization
SUS	System Usability Scale
SWR	Stale While Revalidate
UI	User Interface
XP	Experience Points

Úvod

Na trhu existuje řada aplikací a softwarových programů na výuku angličtiny. I přes velký pokrok a vývoj v této oblasti má stále většina těchto programů značné množství nedostatků. Některé aplikace poskytují zábavnou formu výuky za cenu nižší efektivity učení. Jiné se naopak soustředí na vysokou efektivitu učení, ale neposkytují tak kvalitní uživatelskou zkušenost. Většina těchto aplikací dovede uživatele pouze na středně pokročilou úroveň angličtiny a neumožňuje mu si přizpůsobit výuku svým potřebám.

Výše uvedené důvody tak vedly ke vzniku projektu, jehož cílem je vytvořit a nasadit funkční aplikaci, která bude balancovat efektivitu, zábavnost a přizpůsobitelnost učení. Aplikace tak bude umožňovat samoukům dosáhnout pokročilých úrovní angličtiny a neustále se zdokonalovat v anglickém jazyce.

Pro dosažení tohoto cíle byl projekt rozdělen do dvou bakalářských prací. Tato bakalářská práce se bude zabývat analýzou existujících řešení pro výuku angličtiny a dále návrhem, implementací, testováním a nasazením frontendové části aplikace. Součástí této práce bude také jednoduchý mock, který bude simulovat chování reálného REST API. Druhá bakalářská práce Backend webové aplikace na učení se angličtiny, kterou se zabývá kolega Jan Hlaváč, se pak zaměřuje převážně na analýzu požadavků a dále na návrh, implementaci, testování a nasazení backendové části aplikace. Tyto dvě části aplikace budou komunikovat přes REST API, které bude navrženo a implementováno jako součást backendové části projektu.

Hlavním cílem této bakalářské práce je tedy vytvořit a nasadit funkční frontend aplikace na učení se angličtiny. Z toho plyne řada dílčích cílů, na které se zaměřují jednotlivé kapitoly tohoto textu. Prvním dílčím cílem bude provést analýzu existujících řešení pro výuku angličtiny, která umožní odhalit silné a slabé stránky stávajících aplikací. Druhým cílem pak bude na základě této analýzy, a také na základě analýzy požadavků kolegy Jana Hlaváče, vytvořit návrh frontendové části aplikace. Součástí toho bude také vytvoření návrhu uživatelského rozhraní a volba technologií, které budou v rámci vývoje využívány. Dalším cílem bude provést implementaci samotné aplikace, včetně vytvoření jednoduchého mocku koncových bodů REST API. Čtvrtým cílem bude navrhnout a realizovat vhodné formy testování frontendové části aplikace. Následujícím cílem bude nasadit frontend aplikace do produkčního prostředí. Posledním cílem pak bude zhodnotit výsledné řešení a navrhnout možné úpravy do budoucna. Tato práce je zaměřena výhradně na vývoj aplikace, nikoliv na vytváření jejího obsahu. Součástí práce proto nebude například vytváření konkrétních kurzů, lekcí a cvičení.

Výsledek práce bude prospěšný pro jednotlivce, kteří se chtějí zdokonalit v anglickém jazyce. Aplikace bude užitečná zejména pro uživatele, kteří usilují o větší kontrolu nad tématy, která se učí. Program tak budou moci využít například samouci nebo studenti středních a vysokých škol.

Kapitola 1

Analýza

Jako první krok v rámci vývoje softwaru provedu analýzu. Nejprve se zaměřím na analýzu již existujících aplikací, která pomůže odhalit jejich silné a slabé stránky a další podstatné informace, které pomohou sestavit požadavky tohoto projektu.

Součástí analýzy je dále také analýza požadavků. Tu provádí kolega Jan Hlaváč v rámci jeho bakalářské práce, a proto zde tuto analýzu popíšu jen stručně. Zaměřím se zejména na nefunkční požadavky, které jsou specifické pro frontendovou část aplikace.

1.1 Analýza existujících řešení

Na trhu existuje celá řada aplikací na výuku jazyků. Některé se zaměřují čistě na výuku jednoho jazyka, zatímco jiné se snaží pokrýt jazyků co nejvíce. Tyto aplikace lze také rozdělit podle rozsahu výuky. Některé z nich se soustředí jen na určité aspekty osvojování jazyka, jako například na slovní zásobu nebo na správnou výslovnost. Jiné se naopak snaží pokrýt kompletní výuku daného jazyka. Dále se aplikace liší způsobem výuky. Některé slouží jako pomůcka pro učitele ve školách a jiné jsou naopak určeny pro samouky.

K analýze jsem tedy zvolil čtyři nejpoužívanější aplikace na učení se jazyků [1], které jsou určeny převážně pro samouky a snaží se pokrýt kompletní výuku anglického jazyka. Tato řešení jsem vybral na základě počtu aktivních uživatelů a počtu instalací.

Na tomto místě je vhodné poznamenat, že existuje i celá řada aplikací, které se zaměřují čistě na výuku anglického jazyka. Ty mají ovšem výrazně nižší počet aktivních uživatelů, což může mít za následek menší rozpočet pro vývoj a uživatelské testování a v důsledku i nižší kvalitu výsledné aplikace. Proto jsem se rozhodl analyzovat existující řešení, která vyučují více jazyků a jsou používanější.

V analýze se zaměřím na frontendovou část aplikací, zejména na jejich použitelnost, silné stránky a příležitosti pro zlepšení. Dále se zaměřím na klíčové vlastnosti těchto aplikací, kterými jsou způsoby výuky anglického jazyka a způsoby motivace uživatelů. Získané informace budou sloužit jako podklad k vytvoření funkčních požadavků a následnému sestavení případů užití.

1.1.1 Duolingo

Duolingo¹ je světově nejpoužívanější aplikace na učení se jazyků [1]. Je k dispozici jako webová aplikace, ale i jako mobilní aplikace pro Android a iOS. Celkem Duolingo vyučuje více než 40 jazyků [2].

¹Dostupné na: <https://www.duolingo.com/>

Uživatelské rozhraní se mírně liší podle toho, zda uživatel používá mobilní, nebo webovou aplikaci. Dále také závisí na konkrétním kurzu, který se uživatel učí. Pro analýzu proto použiji mobilní aplikaci s kurzem vyučujícím angličtinu v češtině. Pro jednoduchost budu dále v textu označovat jazyk, který se uživatel v rámci kurzu učí, jako L2. Jako L1 budu označovat primární jazyk, pomocí kterého se v kurzu uživatel učí jazyk L2.

1.1.1.1 Analýza použitelnosti

Duolingo se svým designem značně liší od ostatních aplikací. Při prvním otevření aplikace je okamžitě zachycena pozornost uživatele díky velkému množství kreslených obrázků a velice pestré paletě barev. Duolingo používá v rámci celé aplikace kreslené obrázky s konzistentním stylem [3] a žádné fotografie. Tím se odlišuje od ostatních analyzovaných aplikací, které často dávají přednost fotografiím, které na uživatele mohou působit méně konzistentním dojmem.

S tím souvisí používání kreslených postav. V aplikaci Duolingo je uživatel v průběhu učení doprovázen řadou animovaných postav [4], díky kterým mohou být lekce výrazně zábavnější ve srovnání s lekcemi v ostatních aplikacích. Kromě postav Duolingo také používá značné množství méně výrazných animací a efektů, například při správné odpovědi v průběhu cvičení. V aplikaci lze, kromě grafických efektů, nalézt také celou řadu zvukových efektů. Ty opět přispívají k celkově pozitivní uživatelské zkušenosti.

Po registraci a zapsání se do kurzu je uživateli ihned zobrazena první lekce s jednoduchými cvičeními. V rámci lekce jsou zobrazovány i malé nápovědy, díky kterým se uživatel jednodušeji naučí aplikaci ovládat a procházet cvičení samostatně.

Samotná navigace je v aplikaci snadná a přehledná. Na hlavní stránce je k dispozici přehled všech lekcí kurzu. Lekce jsou uspořádány do tvaru připomínající cestu, vedle které se nachází dříve zmíněné animované postavy.

Co se uživatelské zkušenosti týče, působí Duolingo velice pozitivním dojmem. Důvodem může být značné množství testů [5], které Duolingo provádí za účelem zdokonalení celé aplikace.

1.1.1.2 Způsoby učení

Hlavní stránka aplikace znázorňuje cestu učení s jednotlivými lekcemi. Plněním lekcí uživatel postupně odemyká následující lekce a posouvá se tak dál v kurzu.

Aplikace zahrnuje celou řadu cvičení pokrývajících většinu aspektů učení se jazyka. Při studiu dané lekce jsou uživateli zobrazovány různé druhy cvičení. Díky tomu učení působí méně repetitivně, než u jiných aplikací.

Při učení lze nalézt například následující typy cvičení.

- Překlad věty z jednoho jazyka do druhého pomocí nabízených slov. Uživateli je umožněno místo vybírání ze slov napsat celou větu ručně.
- Poslech věty nebo fráze v L2 a sestavení této věty pomocí nabízených slov v L2. Opět je uživateli umožněno místo vybírání slov napsat celý text ručně.
- Spojení slov v L1 a odpovídajících slovíček v L2. Jednotlivá slovíčka jsou zobrazena buď v textové formě, nebo ve formě audiozáznamu, který si uživatel může kliknutím přehrát.
- Doplnění slovíčka z nabídky do věty na základě obrázku, který odpovídá danému slovíčku.
- Doplnění chybějících slov do tabulky. Toto cvičení se například používá při časování sloves.

Při procházení cvičení jsem upozoroval několik detailů, které uživateli usnadňují učení. Příkladem toho je možnost přeskočit některá cvičení, nebo možnost dočasně vypnout poslechová cvičení, pokud si uživatel zrovna nemůže audionahrávku poslechnout. Podstatnou funkcí je také možnost libovolně cvičení nahlásit. Duolingo tak může díky zpětné vazbě uživatelů podchytit chyby v obsahu cvičení.

Duolingo se od ostatních aplikací liší také tím, že má uživatel omezené množství chyb, které může při studiu udělat. Po pěti chybách se učení na několik hodin zablokuje, pokud uživatel nemá koupenou placenou verzi aplikace.

Před koncem každého učení se uživateli znovu zobrazí cvičení, ve kterých chyboval. Pokud uživatel v dané sekvenci cvičení neudělal žádnou chybu, je mu místo opakování zobrazeno několik těžších cvičení.

Po dokončení dané sekvence cvičení se uživateli zobrazí jeho odměna ve formě XP bodů, které znázorňují jeho pokrok v aplikaci. Uživatel má také možnost si zobrazit přehled všech jeho odpovědí. Na konci učení Duolingo také zobrazuje reklamu.

1.1.1.3 Způsoby motivace uživatele

Motivace uživatelů je klíčová při osvojování jazyka. Duolingo tak zahrnuje řadu mechanismů, které se snaží motivovat uživatele v průběhu procházení kurzů. Jedním z těchto mechanismů je gamifikace, kterou lze definovat jako používání prvků herního designu v mimoherních kontextech [6] (překlad autora). V aplikaci Duolingo se vyskytují následující prvky gamifikace.

1.1.1.3.1 Denní cíl a řada Při registraci si uživatel nastaví svůj denní cíl učení, který se bude snažit splnit. Každý den, kdy se uživatel něco naučil, se mu přičte bod k tzv. denní řadě. Cílem uživatele je nepřerušit tuto denní řadu a dosáhnout tak co nejvyššího skóre.

1.1.1.3.2 XP body Dokončováním lekcí uživatel získává body XP, které se zobrazují na jeho veřejném profilu. Tyto body reprezentují pokrok uživatele v aplikaci.

1.1.1.3.3 Drahokamy Učením uživatel získává drahokamy. Jedná se o virtuální měnu, pomocí které lze v aplikaci například koupit pozastavení denní řady. Uživatel tak může učení na několik dnů vynechat, aniž by svoji denní řadu ztratil.

1.1.1.3.4 Ranní a večerní truhly Každý den má uživatel možnost získat tzv. ranní a večerní truhlu s odměnou tím, že v odpovídající denní dobu dokončí alespoň jednu lekci.

1.1.1.3.5 Žebříčky uživatelů Uživatel svým učením může s ostatními uživateli soutěžit o postup do různých úrovní v žebříčku uživatelů. Tato úroveň v žebříčku se pak zobrazuje na profilu daného uživatele.

1.1.1.3.6 Denní úkoly a výzvy Každý den aplikace zadá uživateli několik jednoduchých úkolů, které může splnit. Přibližně jednou za měsíc je také zveřejněna výzva, která funguje obdobně, ale je obtížnější na splnění.

1.1.1.3.7 Ocenění Pokud uživatel splní určitý cíl, získá ocenění, které se zobrazí na jeho profilu. Příkladem takového ocenění je naučení se 2000 slov v daném kurzu nebo absolvování 50 lekcí bez chyb.

1.1.1.3.8 Vytváření si vlastního avatara Duolingo umožňuje uživatelům si vytvořit vlastní postavu do svého profilového obrázku.

Duolingo také umožňuje uživatelům si vytvářet přátele a sledovat pokrok ostatních. V kombinaci s výše zmíněnými formami gamifikace tak aplikace vytváří kompetitivní, ale přátelské prostředí, ve kterém je uživatel motivován se učit vybrané jazyky.

1.1.1.4 Silné stránky

V aplikaci jsem identifikoval následující silné stránky.

Duolingo obsahuje kreslené obrázky s konzistentním stylem, čímž se odlišuje od konkurentů.

V aplikaci lze nalézt rozsáhlé, ale vybalancované použití pestrých barev, efektů a animací.

Klíčovou výhodou této aplikace je velice silná gamifikace. Aplikace místy působí spíše jako zábavná hra než jako aplikace na učení se jazyků.

Aplikace obsahuje propracované a uživatelsky přívětivé učení s mnoha různými cvičeními. Díky tomu učení probíhá hladce a není repetitivní.

Duolingo také obsahuje tzv. stories neboli krátké příběhy, díky kterým si uživatel může procvičit například slovní zásobu nebo gramatiku v různých situacích.

1.1.1.5 Příležitosti pro zlepšení

Při analýze aplikace jsem dále identifikoval následující příležitosti pro zlepšení.

Většina vyučovaných kurzů naučí uživatele pouze základní nebo středně pokročilou úroveň daného jazyka. Aplikace například neobsahuje pokročilou slovní zásobu nebo gramatiku, které jsou klíčové pro dosažení pokročilých úrovní jazyka.

Duolingo obsahuje přizpůsobené učení, ve kterém si uživatel může procvičovat aspekty jazyka, ve kterých často chybuje. Tato funkce je ovšem dostupná pouze pro uživatele s aktivním předplatným.

Aplikace obsahuje vysvětlení gramatických jevů. Tato vysvětlení ovšem bývají často krátká a nedostačující pro pochopení dané látky.

Uživatel si nemůže přizpůsobit, která témata se chce učit. Je tak často nucen se učit slovní zásobu, kterou již zná, nebo kterou nepotřebuje umět.

Duolingo má u každého kurzu striktně danou cestu podle které musí uživatel postupovat. Uživatel si tak nemůže prohlížet lekce, které si ještě neodemknul, a nemůže se je učit. Aplikace umožňuje přeskočit řadu lekcí splněním testu. Tím se ovšem může uživatel potenciálně připravit o látku, kterou neovládá tak dobře.

Aplikace také neumožňuje se učit například pouze gramatiku. Každá lekce obsahuje různé druhy cvičení a uživatel nemá možnost si vybrat, jestli se chce například zaměřit pouze na slovní zásobu, výslovnost nebo gramatiku.

Potenciální nevýhodou může být příliš agresivní gamifikace. Uživatel mnohdy musí v průběhu učení projít řadu stránek, které mu jen zobrazují informace týkající se jeho denního cíle, ocenění a úspěchů. To může být pro uživatele potenciálně rozptylující.

1.1.1.6 Další poznatky

V rámci analýzy jsem dále narazil na následující poznatky.

Duolingo umožňuje uživateli používat aplikaci i bez nutnosti registrace.

Při registraci Duolingo umožňuje vyplnění úvodního testu, na základě kterého je odhadnuta úroveň uživatele.

Při učení je většina akcí doprovázena zvukem. Při kliknutí na libovolné slovíčko se například přehraje zvuk daného slova. Díky tomu může uživatel trénovat poslech, aniž by cíleně dělal poslechová cvičení.

Pokud uživatel při učení nerozumí některému slovíčku, může na něj kliknout a zobrazí se mu nápověda s překladem.

Při používání webové aplikace lze učení jednoduše ovládat pomocí klávesnice. Učení je tak výrazně rychlejší a plynulejší.

V aplikaci Duolingo existuje stránka s novinkami, kde se zobrazují různé články o učení se jazyků, ale také informace o postupech a dosažených ocenění přátel daného uživatele. Na tyto zprávy může uživatel reagovat pomocí emotikonů.

1.1.2 Mondly

Druhou nejpoužívanější [1] aplikací na učení se jazyků je Mondly². Podobně jako Duolingo je Mondly k dispozici jako mobilní i jako webová aplikace.

1.1.2.1 Analýza použitelnosti

Na první pohled působí design Mondly méně konzistentně než u analyzované aplikace Duolingo. Celá aplikace používá jako barvu pozadí modrou s přechodem do fialové. Společně s těmito barvami Mondly využívá oranžovou, červenou, světle zelenou a růžovou barvu. Tato paleta místy působí velice agresivně, obzvláště v kombinaci se zmíněným modrofialovým pozadím. To může být pro uživatele rozptylující, zejména při učení.

Aplikace dále používá kombinaci fotografií a kreslených obrázků. To vnáší do uživatelské zkušenosti další nekonzistenci, které si nelze nepovšimnout.

V Mondly lze, podobně jako v aplikaci Duolingo, najít značné množství animací a efektů. Animované jsou v Mondly ovšem pouze jednoduché prvky grafického rozhraní. Aplikace tak neobsahuje komplexnější animace postav nebo ikon, na rozdíl od dříve analyzované aplikace Duolingo.

Po registraci a zapsání kurzu aplikace uživateli nijak nevysvětluje, jak má při učení postupovat. Jednotlivá cvičení jsou ovšem intuitivně navržena a není těžké je splnit i bez nápovědy. V průběhu učení jsem si povšiml poněkud zvláštního rozložení jednotlivých tlačítek. Některá tlačítka byla také, kvůli dříve zmíněné paletě barev, velice špatně viditelná. Některé prvky na obrazovce na sebe naopak upozorňovaly výraznou barvou, přestože byly při cvičení naprosto nepodstatné.

Navigace v aplikaci je přehledná. Podobně jako v aplikaci Duolingo jsou zde lekce vyobrazeny na hlavní stránce ve tvaru připomínajícím cestu na mapě. Mondly také umožňuje uživateli si zobrazit lekce v podobě seznamu. Dále umožňuje mezi lekcemi vyhledávat, což značně usnadňuje práci, když chce uživatel najít konkrétní lekci.

Celkově Mondly působí místy zvláštním dojmem, zejména co se uživatelské zkušenosti týče. K tomu přispívá již zmíněná paleta barev, výběr obrázků a místy neintuitivní pozice prvků na obrazovce. Některé složitější funkce jsou v aplikaci nedostatečně vysvětlené, což může vést ke zmatení uživatele.

1.1.2.2 Způsoby učení

Podobně jako Duolingo tato aplikace zobrazuje na hlavní stránce cestu uživatele s danými lekcemi. Na rozdíl od aplikace Duolingo Mondly dává uživateli více volnosti, co se procházení lekcí týče. Uživatel si například může procvičovat lekce dopředu bez nutnosti dokončení všech předchozích lekcí.

Dále v Mondly existují tzv. denní lekce, které jsou každý den obměňovány. Podobným způsobem fungují týdenní kvízy, kde si může uživatel otestovat své znalosti, pokud v daném týdnu dokončil všechny denní lekce. Když uživatel splní alespoň 3 týdenní kvízy, může se pokusit o splnění tzv. měsíční výzvy.

Aplikace nabízí velice podobná cvičení jako Duolingo. Příkladem takových cvičení jsou překlady vět, doplňování slov do vět a přiřazování slov obrázkům. Unikátním cvičením je zde sestavování slov z nabízených písmen.

Dále aplikace obsahuje jednoduchého chatbota, který simuluje konverzace s uživatelem v různých prostředích, jako například v restauracích nebo v hotelech. Celá konverzace je zobrazována ve formě chatu. Všechny věty v konverzaci jsou zobrazeny v obou jazycích. S chatbotem lze komunikovat pomocí mikrofону nebo klávesnice a uživatel má při komunikaci na výběr několik doporučených odpovědí. Podle mého testování byl chatbot schopen přijímat i některé odpovědi,

²Dostupné na: <https://www.mondly.com/>

kteřé nebyly v nabídce dostupné. Také bylo možné se ptát na velice jednoduché otázky, které byly mimo hlavní téma daného cvičení. Když chatbot odpověď neznal, jednoduše konstatoval, že dané otázky nebo věty nerozumí.

1.1.2.3 Způsoby motivace uživatele

Podobně jako Duolingo obsahuje Mondly určité formy gamifikace. Ta se zde ovšem nachází v mnohem menší míře. V Mondly je možné nalézt například XP body, žebříčky uživatelů a denní řadu. Na tyto mechaniky ovšem nebývá příliš upozorňováno a působí tak velice nevýrazně. Dále v této aplikaci není možné získávat virtuální měnu nebo ocenění. Mondly také uživateli průběžně zobrazuje, kolik slov a frází se již naučil, což může být pro některé uživatele motivující.

1.1.2.4 Silné stránky

Asi nejvýraznější silnou stránkou této aplikace je výše zmíněný chatbot s konverzacemi zaměřenými na konkrétní situace. Díky tomu si uživatel může vyzkoušet slovní zásobu a fráze, které by v daných situacích použil.

Silnou stránkou je zde také dostupnost kurzů a pokročilých lekcí zaměřených na velice specifická témata. Příkladem toho jsou jazykové kurzy na trénování recepčních v hotelu, stevardů, zdravotnického personálu nebo prodavačů.

Další výhodou jsou podrobnější nápovědy v průběhu gramatických cvičení. Tyto nápovědy například vysvětlují správné skloňování daného slova ve větě. To může být pro uživatele užitečné, nicméně stále nejsou nápovědy tak podrobné, aby uživatel plně pochopil, proč udělal například v daném cvičení chybu.

Aplikace také obsahuje stránku s podrobnějšími statistikami o učení daného uživatele. Zde se zobrazují informace jako například počet naučených slov a frází, nebo množství času, které uživatel nad učením strávil.

Poslední výhodou je jednoduché vyhledávání obsahu aplikace. Lekce jsou přehledně seskupené podle témat a lze je prohledávat pomocí vyhledávacího pole. Díky tomu může uživatel velice jednoduše najít konkrétní lekci, kterou hledá.

1.1.2.5 Příležitosti pro zlepšení

Značnou nevýhodou může být pro uživatele fakt, že má tato aplikace většinu svého obsahu placenou. Zdarma je k dispozici pouze několik úvodních lekcí každého kurzu.

Dále aplikace obsahuje výrazně méně druhů cvičení než například Duolingo. Kvůli tomu se může stát učení po delší době monotónní. Některá cvičení jsou také poněkud nešťastně navržena, protože umožňují dokončit celé cvičení, aniž by uživatel jakkoliv použil jazyk, který se učí. Příkladem toho je přiřazování slov v L1 k obrázkům se slovy v L2. V tomto cvičení může uživatel jednoduše přiřazovat slova v L1 k obrázkům, aniž by jakkoliv použil jazyk L2, který se učí.

Další nevýhodou Mondly je nedostatečné vysvětlení gramatických jevů. Kvůli tomu může být pro uživatele mnohdy velice složité gramatiku pochopit, obzvláště u pokročilejších lekcí.

Při učení také může být pro uživatele nepříjemné, že jsou lekce seskupeny podle daných témat. Když pak uživatel při učení postupuje chronologicky podle kurzu, musí se například naučit osm lekcí slovní zásoby zaměřených pouze na jedno téma, než se přesune na další skupinu lekcí. Učení pak může na uživatele z nedostatku různorodosti působit příliš zdlouhavě.

V některých částech uživatelského rozhraní byly také nepřesné, a místy i špatné, překlady. To může na uživatele působit neprofesionálním dojmem, obzvláště když se jedná o aplikaci na výuku jazyků.

Poslední nevýhodou je množství reklam. Mondly často zobrazuje uživateli reklamy na placenou verzi této aplikace. Při průchodu aplikací jsem dokonce narazil na stav, kdy se zobrazovaly dvě reklamy přes sebe a uživatelské rozhraní tak působilo rozbitě.

1.1.2.6 Další poznatky

Při analýze jsem narazil na několik dalších poznatků.

Při učení uživatel může kliknout na neznámé slovíčko a zobrazit si tak nápovědu. Ta bývá výrazně podrobnější než v ostatních aplikacích.

Stejně jako Duolingo tato aplikace umožňuje uživateli se začít učit bez nutnosti registrace.

V průběhu učení se při kliknutí na libovolné slovo přehraje audionahrávka správné výslovnosti. To může opět přispět k lepší uživatelské zkušenosti při učení.

1.1.3 Busuu

Busuu³ je třetí nejpoužívanější [1] aplikací na učení se jazyků. Tato aplikace se vyjímá zejména svým přehledným uživatelským rozhraním, používáním videí při učení a aktivní komunitou, v níž si uživatelé mohou navzájem dávat zpětnou vazbu ke splněným cvičením. Busuu je k dispozici jako mobilní i jako webová aplikace.

1.1.3.1 Analýza použitelnosti

Uživatelské rozhraní Busuu působí oproti předchozím aplikacím čistším a přehlednějším dojmem. Aplikace využívá velice jednoduchou paletu barev. Pro označení většiny tlačítek využívá modrou barvu a všechny prvky uživatelského rozhraní jsou vyobrazeny na bílém pozadí. Dále Busuu používá červenou, oranžovou a zelenou barvu. Tyto barvy ovšem využívá v mnohem menší míře, než například Duolingo nebo Mondly.

Rozložení prvků v uživatelském rozhraní je velice přehledné, což s výše zmíněnou paletou barev tvoří velice pozitivní uživatelskou zkušenost, obzvláště v průběhu učení.

Busuu využívá v aplikaci jak fotografie, tak i kreslené obrázky. Kreslené obrázky jsou ovšem využívány velice zřídka a jdou ruku v ruce se vzhledem celé aplikace. Díky tomu nenarušují uživatelskou zkušenost a dobře doplňují výše zmíněné fotografie.

Významnou výhodou této aplikace jsou videa, která jsou uživateli zobrazována v průběhu učení. Video jsou velice krátká a vystupují v nich rodilí mluvčí, kteří například říkají větu, kterou má uživatel za úkol napsat. Díky tomu, že se jedná o videa, a nikoliv jen o audionahrávku, je učení poutavější a zábavnější. Tato videa jsou také často natáčena například na ulici nebo v restauracích, tedy v místech, kde by člověk přirozeně daný jazyk používal. To přináší uživatelům velice pozitivní uživatelskou zkušenost, ke které přispívá i vysoká kvalita samotných videí.

Navigace v Busuu je velice přehledná. V aplikaci je velice jednoduché se zorientovat, zejména v hlavních funkcích, kterými jsou například prohlížení lekcí nebo učení.

Jedinou výraznou nevýhodou, na kterou jsem při analýze použitelnosti narazil, je pomalé načítání hlavní stránky, na které se nachází přehled kurzu s jednotlivými lekcemi. Někdy načítání trvalo i tři sekundy, což může mít negativní dopad na uživatelskou zkušenost [7], obzvláště u stránky, kterou uživatel navštěvuje často.

Celkově aplikace působí velice pozitivním dojmem. Uživatelské rozhraní je přehledné, konzistentní a používání aplikace je velice snadné.

1.1.3.2 Způsoby učení

Na rozdíl od předchozích dvou aplikací nemá Busuu na hlavní stránce vyobrazenou cestu učení. Místo toho jsou lekce zobrazeny ve formě jednoduchého seznamu.

Jednou z unikátních funkcí této aplikace je opravování cvičení ostatním uživatelům. Po studiu dané lekce může uživatel napsat nebo nadiktovat libovolnou zprávu v jazyce, který se učí. Tato zpráva je pak zobrazena uživatelům, kteří jsou rodilými mluvčími v tomto jazyce. Ti pak mohou

³Dostupné na: <https://www.busuu.com/>

na zprávu reagovat, ohodnotit ji a zanechat u ní zpětnou vazbu. Díky tomu mohou mezi sebou uživatelé komunikovat a získávat zpětnou vazbu od rodilých mluvčích.

Co se samotného učení týče, používá Busuu obdobná cvičení jako předchozí aplikace. V této aplikaci se ovšem také klade důraz na sledování videí a následné doplňování slov do vět. Překládání vět z jednoho jazyka do druhého zde není tak často jako v předchozích dvou aplikacích. Po dokončení dané lekce si může uživatel také zobrazit přehled naučených frází a slovíček.

Při průchodu kurzy jsem si všiml, že je mnoho lekcí vytvořeno tak, aby uživateli představily kulturu zemí, které mluví daným jazykem. V lekcích tak často byla zmiňována například známá místa v dané zemi nebo různé kulturní zvyky. Díky tomu se uživatel může při procházení lekcí naučit nejen samotný jazyk, ale i další podstatné informace o kultuře a zemích, které daným jazykem mluví.

Busuu také umožňuje uživatelům si sestavit vlastní plán učení. V tomto plánu si může uživatel zvolit, jaké úrovně chce v daném jazyce dosáhnout. Dále si může zvolit, které dny, v jaký čas a jak dlouho se chce daný jazyk učit. Busuu na základě zvolených parametrů sestaví plán přizpůsobený danému uživateli. Tato funkce je ovšem dostupná pouze pro uživatele, kteří mají zaplacené předplatné.

1.1.3.3 Způsoby motivace uživatele

V Busuu existuje, podobně jako v předchozích aplikacích, několik forem gamifikace. Podobně jako v Mondly zde není na gamifikaci kladen příliš vysoký důraz. V Busuu najdeme denní řady a týdenní žebříčky uživatelů, které jsou velice podobné žebříčkům v aplikaci Duolingo.

Busuu se od ostatních aplikací liší svými týdenními výzvami. V nich má uživatel za úkol odpovědět na danou otázku formou psaní nebo mluvení. Odpovědi na otázky jsou potom hodnoceny ostatními uživateli. Díky výzvám si tak uživatel může v aplikaci vytvářet nové přátele, se kterými si může posílat jednotlivá cvičení a získávat zpětnou vazbu.

V aplikaci má každý uživatel svůj profil, na kterém se zobrazují statistiky o jeho učení. Mezi nimi najdeme například počet naučených slov, denní řadu nebo počet cvičení, které uživatel ostatním opravil. Tyto statistiky mohou uživatele motivovat, aby aplikaci nadále používal a snažil se zdokonalovat v daném jazyce.

1.1.3.4 Silné stránky

První bezpochybně silnou stránkou Busuu je samotné uživatelské rozhraní. Jeho jednoduchost a přehlednost umožňuje uživateli se jednoduše v aplikaci zorientovat a pochopit všechny její funkce.

Jak jsem již zmínil, v Busuu si mohou uživatelé navzájem dávat zpětnou vazbu k jednotlivým cvičením. Tuto funkci bych považoval za jednu z nejdůležitějších silných stránek této aplikace. V průběhu mé analýzy jsem zpozoroval, že je tato funkce často využívána a uživatelé mezi sebou velice často interagují. Pro uživatele může být vysoce motivující, když ho rodilý mluvčí pochválí nebo když mu dá zpětnou vazbu.

V aplikaci také existuje stránka na opakování slovní zásoby a gramatiky. Stránka je dále rozdělena do tří kategorií podle toho, jak dobře si uživatel pamatuje daná slovíčka a gramatické jevy. Každou kategorii je možné si zvlášť opakovat. Díky tomu se mohou uživatelé přesně zaměřit na slovní zásobu a gramatické jevy, které mu dělají největší problémy.

Další výhodou aplikace jsou kurzy zaměřené na specifická témata, kterými jsou například cestování, život v dané zemi, výslovnost a využití jazyku v pracovním prostředí. To umožňuje uživateli si zvolit kurz podle svých potřeb. Díky tomu se pak nemusí učit látku, kterou by nevyužil.

Placená verze aplikace umožňuje uživateli si nastavit vlastní plán učení, a naplánovat si tak lekce podle svých cílů.

Aplikace také umožňuje uživatelům s předplatným si stahovat jednotlivé lekce a učit se je i bez přístupu k internetu. Tím se Busuu liší od aplikací Duolingo a Mondly, které toto stahování neumožňují.

Poslední významnou výhodou jsou dříve zmíněná videa zobrazovaná v rámci učení. Tato videa dělají cvičení zábavnější a poutavější a umožňují uživateli se více vžít do situací, ve kterých se následně může v životě nacházet, když bude daný jazyk používat.

1.1.3.5 Příležitosti pro zlepšení

Jak jsem již zmínil, některé stránky v aplikaci se načítají i několik sekund, což může mít negativní dopad na uživatelskou zkušenost.

Dále, pokud uživatel používá neplacenou verzi aplikace, je nucen před každou lekcí zhlédnout jednu až dvě reklamy. Reklamy typicky trvají pět až třicet sekund. To může mít také potenciálně negativní dopad na uživatelskou zkušenost.

Aplikace má v základní neplacené verzi určité nevýhody. Uživatel například nemůže přeskakovat jednotlivé lekce a musí je plnit v daném kurzu postupně. Opakování již naučené látky není zahrnuto přímo v rámci učení a uživatel si musí sám látku opakovat na samostatné stránce. Na ní je ovšem opakování pro uživatele bez předplatného značně omezené.

Busuu při učení také nenabízí jednoduchou možnost vložení speciálních znaků daného jazyka do textových polí. Když chce uživatel například vložit apostrof do věty, musí ho najít na vlastní klávesnici. To může být pro některé uživatele velice složité a výrazně to zpomaluje vyplňování jednotlivých cvičení.

1.1.3.6 Další poznatky

Kromě výše zmíněných výhod a nevýhod jsem přišel na následující poznatky.

Busuu při učení uživateli vždy zobrazuje nová slovíčka na zvláštní stránce před tím, než ho z nich začne zkoušet. To se liší od přístupu ostatních aplikací, které uživateli rovnou zobrazí slovíčko v jednoduchém cvičení a uživatel tak musí sám odvodit, jak se slovíčko používá.

Dále Busuu klade větší důraz na poslechová cvičení než ostatní aplikace. Učení tak může být obtížné, pokud se uživatel zrovna nachází na místě, kde nemůže audionahrávky poslouchat.

1.1.4 Babel

Poslední aplikací této analýzy je Babel⁴. Podobně jako předchozí aplikace je Babel k dispozici jako mobilní i jako webová aplikace.

1.1.4.1 Analýza použitelnosti

Babel působí na první pohled minimalistickým dojmem. Aplikace využívá především světlejší barvy, kterými jsou například béžová, světle fialová, světle žlutá nebo světle oranžová. Tlačítka jsou většinou označena červenou barvou, která ovšem může místy působit až příliš agresivně. Všechny prvky uživatelského rozhraní jsou zobrazovány na bílém pozadí.

Prvky jsou na obrazovce rozloženy přehledně a intuitivně. Na rozdíl od aplikací Duolingo a Mondly, tato aplikace nezobrazuje žádné výrazné animace nebo efekty. Oproti zmíněným aplikacím tak Babel působí čistším a klidnějším dojmem.

Aplikace používá kombinaci fotografií, kreslených obrázků a ikon. Fotografie je možné najít zejména v rámci učení. Ikony a kreslené obrázky, které jsou často jen obrysy různých objektů, Babel využívá v rámci celého uživatelského rozhraní.

Babel používá pro navigaci jednoduchou navigační lištu, která obsahuje jen čtyři tlačítka. Aplikace také nemá žádný navigační panel, přestože má řadu různých funkcí, ke kterým může

⁴Dostupné na: <https://www.babel.com/>

uživatel chtít často přistupovat. Kvůli této zjednodušené navigaci pro mě bylo místy obtížné v průběhu analýzy najít některé funkce nebo stránky.

Celkově aplikace působí pozitivním dojmem. Uživatelské rozhraní je přehledné a aplikace se používá jednoduše. Jedinou potenciální slabou stránkou z hlediska použitelnosti je již zmíněná navigace.

1.1.4.2 Způsoby učení

Na hlavní stránce aplikace je zobrazen pouze přehled několika nadcházejících lekcí, které se bude uživatel učit. Tím se Babel liší od ostatních aplikací, které uživateli na hlavní stránce typicky zobrazují rozsáhlý přehled lekcí daného kurzu. Nicméně i v aplikaci Babel si může uživatel zobrazit seznam lekcí, které jsou v kurzu obsaženy.

Babel používá při učení obdobná cvičení jako předchozí aplikace. Stejně jako v ostatních aplikacích zde nalezneme například doplňování slov do vět z nabídky. Toto doplňování probíhá buď při překladu samostatných vět, nebo v simulovaných konverzacích. V těchto konverzacích je uživateli postupně formou chatu zobrazována konverzace několika osob a uživatel má za úkol doplňovat do vět chybějící slova z nabídky. Všechny věty v konverzaci jsou zobrazovány v obou jazycích. Díky tomu má uživatel příležitost vidět, jak jsou jednotlivé věty používány v konkrétních situacích.

Dalším unikátním cvičením v této aplikaci je sestavování slov z nabízených slabik. Uživateli je vždy zobrazeno slovo v jazyce L1 a jeho úkolem je toto slovo sestavit v jazyce L2 s pomocí nabízených slabik.

Po dokončení učení má uživatel možnost si znovu zopakovat cvičení a opravit tak chyby, které v dané lekci udělal.

Babel klade velký důraz na opakování již naučeného materiálu. Hned po první lekci je uživateli nabídnuta možnost si zopakovat nabyté znalosti. Opakování zde probíhá ve 4 formách.

1.1.4.2.1 Opakování formou poslechu První je uživateli nabízeno opakování formou poslechových cvičení. V nich je nejdříve přehrán zvuk slovíčka v L2. Následně má uživatel za úkol vybrat z nabídky text v L1, který odpovídá překladu přehrávaného audia.

1.1.4.2.2 Opakování formou kartiček Další formou opakování jsou kartičky, na nichž je zobrazeno z jedné strany slovo v jazyce L1 a na druhé straně odpovídající překlad v jazyce L2. Cílem cvičení je si vzpomenout na správný překlad daného slovíčka. Po kliknutí na obrazovku se kartička otočí a uživateli se zobrazí správný překlad. Uživatel následně sám zvolí, zda si na slovo vzpomněl správně.

1.1.4.2.3 Opakování formou psaní Třetí formou opakování je překlad frází a slov z jazyka L1 do jazyka L2. Zde se nejedná o doplňování slov, ale o psaní celých překladů na klávesnici.

1.1.4.2.4 Opakování formou mluvení Poslední formou opakování jsou cvičení zaměřená na mluvení. Ve cvičení je nejdříve přehrána audionahrávka daného slova a uživatel má následně za úkol slovo zopakovat do mikrofону.

Babel dále nabízí celou řadu doplňkových cvičení, díky kterým může uživatel trénovat poslech, mluvení a gramatiku i mimo hlavní kurz. Kromě těchto cvičení aplikace také nabízí několik jednoduchých her, díky kterým si uživatel může procvičovat slovní zásobu.

1.1.4.3 Způsoby motivace uživatele

Babel obsahuje ze všech analyzovaných aplikací nejméně způsobů motivace uživatele. Jedinou formou gamifikace je zde denní řada, která je ovšem velice nevýrazná.

V aplikaci má každý uživatel svůj profil, který je ovšem soukromý a ostatní uživatelé si ho nemohou zobrazovat. Babel také neumožňuje si v aplikaci přidávat přátele nebo s ostatními uživateli jakkoliv interagovat.

Jak jsem již zmínil, aplikace obsahuje několik her. Ty jsou ovšem příliš jednoduché na to, aby je mohl uživatel používat k systematickému osvojování daného jazyka.

1.1.4.4 Silné stránky

Babel, podobně jako Busuu, nabízí uživateli možnost si nastavit, které dny a ve který čas se bude učit. Dále umožňuje uživateli si v každém jazyce vybrat z různých specifických kurzů, kterými jsou například kurz zaměřený na medicínu, byznys nebo časté idiomy.

Aplikace také umožňuje uživateli si stáhnout lekce do svého zařízení a procházet je tak i bez přístupu k internetu.

Na hlavní stránce kurzu je uživateli nabízen podcast o daném jazyce, který lze poslouchat přímo v aplikaci. Jednotlivé epizody jsou zaměřeny na různé aspekty jazyka a mohou tak sloužit jako efektivní doplnění látky daného kurzu.

V aplikaci dále existuje stránka, na které si uživatel může zarezervovat místo na online výuce s reálným učitelem. Učitel v průběhu výuky vždy vysvětluje předem dané téma. Tato funkce může být efektivním způsobem, jak propojit učitele se studenty daného jazyka. Žádná z předchozích aplikací tuto možnost nenabízí.

Kromě samotných lekcí lze v aplikaci také najít stránku s doplňujícími aktivitami a materiály. Na ní se nachází dříve zmíněné hry a doplňková cvičení. Dále si na této stránce může uživatel přečíst Babel blog, nebo podrobná vysvětlení gramatických jevů. To považuji za velice silnou stránku této aplikace. Babel v množství i kvalitě poskytnutých materiálů skutečně vyniká. Vysvětlení gramatických jevů může být pro uživatele velice užitečné.

1.1.4.5 Příležitosti pro zlepšení

Babel je především placená aplikace. Zdarma si uživatel může vyzkoušet pouze první lekci daného kurzu. To může uživatele odradit, protože jsou první lekce příliš jednoduché a nemusí být dostačující, aby se uživatel rozhodl, jestli si chce předplatit zaplatit.

Babel ze všech analyzovaných aplikací klade nejmenší důraz na gamifikaci a motivaci uživatelů. Dále aplikace neumožňuje uživatelům mezi sebou jakkoliv interagovat. To může být pro některé uživatele klíčové při rozhodování, zda budou tuto aplikaci používat.

Jak jsem již zmínil, tato aplikace vyniká v poskytování kvalitních materiálů a dodatečných cvičeníh ke kurzu. Ty jsou ovšem umístěny mimo lekce kurzu a uživatel si je musí sám dohledávat. S tím souvisí opakování jednotlivých cvičení. Aplikace poskytuje čtyři možnosti, jak si opakovat látku v daném kurzu. Toto opakování je ovšem opět v rukách uživatele a není zařazeno do samotných lekcí. Celkově má tak uživatel výrazně větší zodpovědnost a musí se sám rozhodnout, jak bude aplikaci používat, aby se daný jazyk co nejefektivněji naučil. To mohou někteří uživatelé vnímat jako výhodu, ale pro mnoho z nich to může být značně nepříjemné. Materiálů a cvičení je v aplikaci celá řada a může být velice těžké se rozhodnout, na co se má uživatel při učení a opakování zaměřit.

Další nevýhodou je již dříve zmíněná navigace. V aplikaci je mnoho funkcí a stránek, ale kvůli příliš zjednodušené navigaci může být pro nové uživatele obtížné najít, co hledají.

1.1.4.6 Další poznatky

Kromě výše zmíněných výhod a nevýhod jsem došel k následujícím poznatkům.

Babel má odlišnou formu rozřazovacího testu, než mají ostatní analyzované aplikace. V tomto testu uživatel není přímo zkoušen z dané látky, ale jen odpovídá, do jaké míry si myslí, že je schopný vyjádřit se o určitém tématu. Příkladem toho je otázka, do jaké míry si uživatel myslí, že zvládne v daném jazyce popsat barvy, které vidí kolem sebe.

Uživatel si může v aplikaci ukládat slovíčka do vlastních kolekcí. Tyto kolekce ovšem nemají žádné další funkce. Uživatel například nemá možnost spustit učení pouze se slovíčky z dané kolekce.

1.1.5 Závěr analýzy

V této části textu zformuluji nejdůležitější závěry z analýzy existujících řešení. Tento výstup bude sloužit jako podklad k sestavení funkčních požadavků, kterými se zabývá kolega Jan Hlaváč v jeho bakalářské práci. Tyto poznatky dále využiji při sestavení případů užití a při samotném návrhu uživatelského rozhraní.

1.1.5.1 Uživatelská zkušenost

V rámci analýzy použitelnosti jednotlivých aplikací jsem došel k následujícím závěrům.

Většina aplikací má přehledné a jednoduše použitelné uživatelské rozhraní. I přes to jsem našel řadu nedostatků, kterými jsou například nevhodná volba palety barev nebo problematická navigace.

Všechny analyzované aplikace hojně využívají fotografie, kreslené obrázky a ikony. Při používání těchto prvků je však třeba dbát na zachování konzistentního dojmu, kterého některé analyzované aplikace nebyly schopné docílit.

Aplikace dále zahrnují celou řadu efektů, jak zvukových, tak grafických. Společně s nimi obsahují i značné množství animací nebo videí. Některé aplikace tak místy připomínají spíše mobilní hru než aplikaci na učení se jazyků. Udržení pozornosti uživatelů při učení je klíčové, a proto bude třeba zvážit použití těchto prvků při návrhu uživatelského rozhraní.

Některé analyzované aplikace obsahovaly v uživatelském rozhraní prvky, které mohou zapříčinit negativní uživatelskou zkušenost. Z tohoto důvodu bude vhodné při návrhu využít některý ze standardů pro návrh uživatelských rozhraní, kterým je například Material Design [8].

Uživatelská zkušenost je v této aplikaci klíčová, a proto bude vhodné v rámci testování zvážit upřednostnění uživatelského testování, díky kterému bude možné se vyhnout nedostatkům v návrhu.

1.1.5.2 Učení

Analyzované aplikace využívají při učení různé druhy cvičení. V rámci návrhu bude třeba rozhodnout, které druhy cvičení budou v této aplikaci zahrnuty, aby se jimi pokryla kompletní výuka daného jazyka.

V rámci učení některé aplikace poskytují nápovědy k daným cvičením. To může být pro uživatele přínosné, zejména pokud mu daná látka nebyla dříve vysvětlena.

Při učení aplikace často využívají kreslené obrázky a fotografie, které se týkají daného tématu, ze kterého je uživatel zkoušen. Díky tomu mohou cvičení působit poutavějším dojmem. Proto bude vhodné při návrhu zvážit zahrnutí různých grafických prvků do samotných cvičení.

Dále většina aplikací ve cvičeních využívá audionahrávky správných výslovností daných slov. Ty bývají často spouštěny automaticky například po dokončení daného cvičení nebo při kliknutí na dané slovíčko. Díky tomu může uživatel trénovat poslech v daném jazyce, aniž by cíleně dělal poslechová cvičení.

Většina analyzovaných aplikací umožňuje uživateli na konci učení znovu vyplnit cvičení, ve kterých chyboval. Při návrhu bude třeba zvážit, jestli bude toto opakování pro uživatele povinné, nebo čistě dobrovolné.

Aplikace strukturují jednotlivé lekce kurzu buď formou seznamu nebo formou připomínající cestu s jednotlivými lekcemi. Při procházení aplikací byly seznamy praktičtější a jednodušší na používání, nicméně lekce ve formě cesty působily vizuálně poutavějším dojmem. Proto bude třeba při návrhu aplikace zvážit obě zmíněné varianty.

1.1.5.3 Způsoby motivace uživatele

Všechny analyzované aplikace používají určité formy gamifikace. Některé aplikace, jako například Duolingo, používají gamifikaci velice často, zatímco jiné aplikace, jako například Babel, gamifikaci používaly velice zřídka. V rámci návrhu bude třeba zvážit, do jaké míry a v jakých formách bude gamifikace do této aplikace zahrnuta.

Dalším významným faktorem jsou sociální interakce. Některé analyzované aplikace využívají systém přidávání přátel, sledování jejich profilů, pokroku a dosažených úspěchů. To může opět motivovat uživatele a udělat celou uživatelskou zkušenost pozitivnější.

1.1.5.4 Odlišení aplikace od existujících řešení

V rámci analýzy jsem identifikoval následující funkce, kterými se může tato aplikace odlišit od již existujících řešení a díky kterým může uživatelům poskytnout lepší požitek.

1.1.5.4.1 Přizpůsobení kurzů Existující řešení neumožňují uživatelům si přizpůsobovat kurzy vlastním potřebám a vybírat si například témata, která je zajímají. Aplikace by tak mohla umožňovat uživatelům si přesně zvolit, které lekce a která témata se chtějí učit. Díky tomu by mohlo být učení efektivnější, protože by se uživatel skutečně mohl zaměřit na látku, kterou potřebuje umět.

1.1.5.4.2 Podrobné výukové materiály Většina analyzovaných aplikací neposkytuje dostatečné vysvětlení jednotlivých aspektů jazyka, kterými jsou například gramatické jevy nebo správná výslovnost. Existující řešení často spoléhají na to, že si uživatel dané jevy přirozeně osvojí plněním cvičení, nebo že si jejich vysvětlení sám vyhledá. To může, obzvláště u pokročilejších jevů, vést k obtížnému učení a negativní uživatelské zkušenosti. Aplikace by tak mohla poskytovat materiály k jednotlivým lekcím. Materiály by mohly existovat jak v textové formě, tak například ve formě krátkých videí, která by uživateli danou látku vysvětlila.

1.1.5.4.3 Přeskakování známé látky Existující řešení uživateli také neumožňují při učení jednoduše přeskocit látku, kterou již zná. Když například uživatel narazí při učení na nové slovíčko, musí se ho učit, přestože ho již umí například ze školy. V aplikaci by tak mohla existovat možnost označit látku za naučenou, díky čemuž by uživatel neztrácel čas procházením látky, kterou již zná.

1.1.5.4.4 Vlastní kolekce slovíček Většina existujících řešení dále neumožňuje uživatelům si vytvářet vlastní kolekce slovíček. Žádné z analyzovaných řešení neumožňuje se tyto kolekce samostatně učit. Tato funkce by opět mohla umožnit uživateli si přizpůsobit učení jeho potřebám.

1.1.5.4.5 Podrobnější informace o slovíčkách Přestože všechny analyzované aplikace obsahují mnoho slovíček, žádná z nich neobsahuje o slovíčkách podrobnější informace. Proto by aplikace mohla poskytovat informace, jako je například správná výslovnost, příklady užití slovíčka ve větách, nebo znázornění, do jaké míry si uživatel dané slovíčko pamatuje.

1.1.5.4.6 Vyhledávání lekcí a slovíček Většina analyzovaných řešení také neumožňuje jednoduše vyhledávat jednotlivá slovíčka nebo lekce pomocí vyhledávacího pole. Tato funkce by opět mohla přinést jednodušší navigaci a pozitivnější uživatelskou zkušenost.

1.1.5.4.7 Odlišné formy gamifikace Analyzovaná řešení obsahují různé formy gamifikace. Žádné z těchto řešení však příliš nevyužívá například virtuální měnu, díky které by si uživatel mohl kupovat virtuální položky v aplikaci. Tím by se mohla tato aplikace odlišit od existujících aplikací, které využívají jiné formy gamifikace.

1.1.5.4.8 Oblíbené lekce a slovíčka Dále by mohla aplikace umožňovat uživateli označit lekce a slovíčka za oblíbená a přidávat je tak do seznamu oblíbených materiálů. Díky tomu by se uživatel mohl velice jednoduše vrátit k uloženým lekcím nebo slovíčkům a nemusel by je znovu vyhledávat. Tato funkce by opět mohla vést k lepší uživatelské zkušenosti.

1.1.5.4.9 Přehlednější rozdělení lekcí Většina analyzovaných aplikací kombinuje v rámci učení více aspektů jazyka do jedné lekce. V rámci jedné lekce si tak uživatel procvičuje například poslech, gramatiku a mluvení zároveň. Kvůli tomu může být pro uživatele obtížné se při učení zaměřit čistě na jeden aspekt jazyka, který mu dělá největší problémy. Rozdělení těchto aspektů jazyka do samostatných lekcí by potenciálně mohlo vést k přehlednější organizaci kurzu a uživatel by se tak mohl jednoduše zaměřit například pouze na trénování výslovnosti.

1.1.5.4.10 Větší důraz na opakování Většina analyzovaných aplikací také neklade dostatečný důraz na opakování již naučené látky. Uživatel si tak musí látku často opakovat z vlastní iniciativy, což může vést k zapomínání již nabytých znalostí. Aplikace by tak mohla zahrnovat průběžné opakování látky při každém učení. Díky tomu by aplikace zajistila, že uživatel již naučenou látku s vysokou pravděpodobností nezapomene.

Jak jsem již zmínil, všechny tyto poznatky budu brát v potaz při návrhu aplikace, sestavování případů užití a návrhu uživatelského rozhraní.

1.2 Analýza požadavků

V této sekci se zaměřím na analýzu požadavků. Funkční a nefunkční požadavky celé aplikace analyzuje kolega Jan Hlaváč v jeho bakalářské práci, která se zabývá backendovou částí této aplikace. Proto zde nebudu tyto požadavky opakovat a zaměřím se čistě na několik vybraných nefunkčních požadavků, které se týkají vývoje frontendové části aplikace.

V kapitole návrhu se pak zaměřím na podrobné popsání případů užití, které přesněji definují požadavky pro frontendovou část této aplikace.

1.2.1 Funkční požadavky

Jak jsem již zmínil, funkčními a nefunkčními požadavky celé aplikace se zabývá Jan Hlaváč v jeho bakalářské práci. Jelikož zmíněná práce ještě není v době psaní tohoto textu publikována, zahrnul jsem analýzu požadavků provedenou Janem Hlaváčem do přílohy `požadavky-aplikace.pdf`. Z této analýzy budu čerpat při návrhu aplikace, zejména při popisování případů užití.

1.2.2 Nefunkční požadavky

V této části textu se zaměřím na nejdůležitější nefunkční požadavky, které se týkají frontendové části aplikace. Tyto požadavky budu brát v potaz při návrhu, vývoji i testování aplikace.

1.2.2.1 N1 Rozšiřitelnost systému o další kurzy a jazyky

Aplikace by měla být jednoduše rozšiřitelná. Přestože je aplikace zaměřena na výuku angličtiny, měla by být navržena a implementována tak, aby v budoucnu umožnila případné rozšíření o další jazykové kurzy. Aplikace by také měla být vyvíjena tak, aby bylo možné v budoucnu poskytovat kurzy v i jiných jazycích než v češtině.

1.2.2.2 N2 Kvalitní uživatelská zkušenost

Jedním z cílů aplikace je vytvořit kvalitní uživatelskou zkušenost a přimět tak uživatele k jejímu dlouhodobému užívání. Z tohoto důvodu je potřeba se zaměřit na uživatelskou zkušenost při návrhu, implementaci i testování.

1.2.2.3 N3 Možnost optimalizace pro SEO

Je pravděpodobné, že bude v budoucnu potřeba různé stránky aplikace poskytnout i neregistrovaným uživatelům a přizpůsobit je pro SEO. Příkladem toho mohou být stránky s jednotlivými lekcemi, slovíčky, nebo s profily uživatelů. Proto je žádoucí, aby aplikace využívala technologie, které toto přizpůsobení v budoucnu umožní.

1.2.2.4 N4 Možnost rozšíření na další platformy

Cílem této práce je vytvořit webovou aplikaci na učení se angličtiny. Přestože se jedná čistě o webovou aplikaci, je velice pravděpodobné, že bude projekt v budoucnu rozšiřován na další platformy a systémy, kterými jsou například Windows, Linux, macOS, Android nebo iOS. Proto je potřeba zvolit pro vývoj vhodné technologie, které v budoucnu umožní aplikaci rozšířit na zmíněné platformy.

V této kapitole se zaměřím na popsání jednotlivých případů užití, které budou následně sloužit jako podklad pro implementaci frontendové části aplikace. Dále popíšu, jakým způsobem jsem postupoval při návrhu uživatelského rozhraní. Nakonec provedu výběr technologií, které budu v rámci implementace používat. V rámci výběru technologií se zaměřím na jednotlivé jazyky, frameworky, knihovny, ale také programy, které umožní dosáhnout cílů této aplikace.

2.1 Případy užití

V této sekci se zaměřím na sestavení případů užití. Případy užití se odvíjí od dříve popsané analýzy existujících řešení a analýzy funkčních požadavků, která byla provedena jako součást bakalářské práce Jana Hlaváče, který se zabývá backendovou částí této aplikace.

V rámci sestavování případů užití je třeba identifikovat jednotlivé aktéry v systému [9]. Tomu se budu věnovat v první části této podkapitoly. V druhé části se pak zaměřím na popis samotných případů užití.

Sestavené případy užití budou následně sloužit jako podklad pro návrh uživatelského rozhraní a také pro implementaci samotné aplikace.

2.1.1 Popis aktérů

V jednotlivých případech užití se vyskytují tzv. aktéři [9]. Případy užití popisují, které postupy a akce budou tito aktéři moci v systému vykonávat. Aktérem může být například uživatel systému, ale i třeba jiný systém. V tomto projektu jsem identifikoval následující aktéry.

2.1.1.1 Nepřihlášený uživatel

Nepřihlášený uživatel je osoba, která není v systému přihlášená. Má možnost se do systému zaregistrovat, přihlásit, ale i například požádat o obnovení hesla.

2.1.1.2 Uživatel

Uživatel je hlavní aktér v této aplikaci. Jedná se o osobu, která je v aplikaci zaregistrovaná a přihlášená. Aplikaci využívá typicky k tomu, aby se naučila daný jazyk. To zahrnuje například zapsání si kurzu, procházení lekcí a plnění jednotlivých cvičení.

2.1.1.3 Premium uživatel

Premium uživatel je uživatel, který má v systému aktivní předplatné, nebo jeho zkušební verzi. Premium uživatel může vykonávat v systému všechny akce běžného uživatele, ale i další akce, ke kterým nemá běžný uživatel přístup.

2.1.1.4 Administrátor

Administrátor je osoba, která nepoužívá frontend této aplikace, ale hraje roli v rámci celého projektu. Administrátor přistupuje k datům aplikace a upravuje je typicky pomocí komunikace s API backendové části projektu.

Případy užití, ve kterých je administrátor hlavním aktérem, zde nebudu popisovat, protože nejsou pro tuto bakalářskou práci relevantní. V některých případech užití ovšem administrátor figuruje, a proto ho zde zahrnuji.

Administrátor může například v systému vytvářet oznámení, která se následně zobrazují daným uživatelům.

2.1.2 Seznam případů užití

Případy užití lze sepisovat s různou úrovní detailu. V tomto projektu jsem se rozhodl do každého případu užití zahrnout název, popis, identifikaci hlavního aktéra a hlavní scénář. Pokud existuje více způsobů, kterými může aktér v rámci daného případu užití postupovat, zahrnul jsem do případu užití i alternativní scénář.

Případy užití jsem se také rozhodl rozdělit podle jejich priorit. Ty jsem se rozhodl určovat pomocí metody MoSCoW [10], která rozlišuje čtyři úrovně priority. Ty určují, které požadavky se v projektu musí splnit (*must have*), které by se měly splnit (*should have*) a které se mohou splnit (*could have*). Čtvrtá úroveň zahrnuje požadavky, které se v rámci projektu nebudou plnit (*won't have*). Díky této metodě lze pak při plánování projektu zajistit, aby byly všechny důležité požadavky splněny a aby se naopak neplýtvá čas na požadavcích, které nejsou podstatné.

Jednotlivé případy užití popíšu v následující části tohoto textu.

2.1.2.1 UC1 – Registrace uživatele

Popis:	Umožňuje nepřihlášenému uživateli se zaregistrovat do aplikace a stát se tak přihlášeným uživatelem.
Priorita:	Must have
Hlavní aktér:	Nepřihlášený uživatel

2.1.2.1.1 Hlavní scénář

1. Případ užití začíná, když se nepřihlášený uživatel rozhodne zaregistrovat se do aplikace.
2. Systém zobrazí formulář umožňující zadat uživatelské jméno, emailovou adresu, heslo a potvrdit souhlas s obchodními podmínkami a zpracováváním osobních údajů.
3. Nepřihlášený uživatel vyplní všechny položky formuláře včetně vyjádření souhlasu s obchodními podmínkami a zpracováváním osobních údajů.
4. Systém zaregistruje a dále přihlásí nepřihlášeného uživatele. Ten se tak stane přihlášeným uživatelem. Ihned po registraci systém umožní uživateli si zapsat a nastavit jazykový kurz. Tento proces je dále popsán v případě užití UC5 – Zapsání kurzu.

2.1.2.2 UC2 – Přihlášení uživatele

Popis:	Umožňuje nepřihlášenému uživateli se přihlásit do aplikace a stát se tak přihlášeným uživatelem.
Priorita:	Must have
Hlavní aktér:	Nepřihlášený uživatel

2.1.2.2.1 Hlavní scénář

1. Případ užití začíná, když se nepřihlášený uživatel rozhodne přihlásit se do aplikace.
2. Systém zobrazí formulář umožňující zadat emailovou adresu a heslo.
3. Nepřihlášený uživatel vyplní všechny položky formuláře.
4. Systém přihlásí nepřihlášeného uživatele, čímž se stane přihlášeným uživatelem. Systém ho následně přesměruje na hlavní stránku aplikace.

2.1.2.3 UC3 – Obnova zapomenutého hesla

Popis:	Umožňuje nepřihlášenému uživateli obnovit zapomenuté heslo.
Priorita:	Must have
Hlavní aktér:	Nepřihlášený uživatel

2.1.2.3.1 Hlavní scénář

1. Případ užití začíná, když se nepřihlášený uživatel rozhodne si obnovit heslo.
2. Systém zobrazí uživateli formulář umožňující zadat emailovou adresu.
3. Nepřihlášený uživatel vyplní formulář a potvrdí, že chce heslo obnovit.
4. Systém odešle uživateli na poskytnutou emailovou adresu email s odkazem pro změnu hesla a zobrazí mu informaci, že mu byl email odeslán.
5. Nepřihlášený uživatel otevře odkaz v emailu a přesune se na danou webovou stránku.
6. Systém zobrazí uživateli formulář pro změnu hesla. Formulář vyžaduje zadání nového hesla a zopakování tohoto hesla, aby se ověřilo, že bylo heslo zadáno správně.
7. Nepřihlášený uživatel vyplní formulář a potvrdí změnu hesla.
8. Systém změní nepřihlášenému uživateli heslo a zobrazí mu informaci, že bylo heslo změněno.

2.1.2.4 UC4 – Odhlášení uživatele

Popis:	Umožňuje uživateli se odhlásit z aplikace a stát se tak nepřihlášeným uživatelem.
Priorita:	Must have
Hlavní aktér:	Uživatel

2.1.2.4.1 Hlavní scénář

1. Případ užití začíná, když se uživatel rozhodne se odhlásit z aplikace.
2. Systém zobrazí uživateli hlavní nabídku aplikace.
3. Uživatel zvolí možnost odhlásit se z aplikace.
4. Systém odhlásí uživatele z aplikace. Uživatel se tak stane nepřihlášeným uživatelem.

2.1.2.5 UC5 – Zapsání kurzu

Popis:	Umožňuje uživateli si vybrat a nastavit jazykový kurz. Zapsání kurzu typicky nastane hned po dokončení registrace uživatele. Uživatel má ovšem možnost si v aplikaci kdykoliv zapsat libovolný z dalších jazykových kurzů, které aplikace nabízí.
Priorita:	Must have
Hlavní aktér:	Uživatel

2.1.2.5.1 Hlavní scénář

1. Případ užití začíná, když se uživatel rozhodne si zapsat nový jazykový kurz.
2. Systém zobrazí formulář pro výběr jazykového kurzu. Formulář se skládá z výběru jazyka, který uživatel ovládá a následně jazyka, který se chce uživatel naučit.
3. Uživatel vybere jazyk, který ovládá a jazyk, který se chce naučit.
4. Systém zobrazí uživateli seznam hlavních témat, které se v kurzu vyskytují.
5. Uživatel si vybere témata, která ho zajímají.
6. Systém zobrazí formulář pro zvolení tzv. denního cíle. Denní cíl určuje, kolik cvičení se bude uživateli zobrazovat při denním učení.
7. Uživatel zvolí svůj denní cíl.
8. Systém zobrazí formulář pro zvolení, kde v kurzu chce uživatel začít. Formulář umožňuje uživateli začít od první lekce, nastavit si vlastní úroveň daného jazyka, anebo splnit úvodní test.
9. Uživatel zvolí variantu, že chce začít od první lekce.
10. Systém zapíše uživateli kurz a přeměruje uživatele na hlavní stránku kurzu.

2.1.2.5.2 Alternativní scénář – volba vlastní úrovně

1. Scénář začíná v 8. kroku hlavního scénáře, jestliže uživatel zvolí, že si chce nastavit vlastní úroveň jazyka.
2. Systém zobrazí formulář pro volbu úrovně jazyka. Formulář obsahuje jednotlivé úrovně s jejich podrobnějším popisem.
3. Uživatel zvolí svoji úroveň jazyka.
4. Systém zapíše uživateli kurz a podle zvolené úrovně určí, kde v kurzu bude uživatel začínat. Následně přeměruje uživatele na hlavní stránku kurzu.

2.1.2.5.3 Alternativní scénář – úvodní test

1. Scénář začíná v 8. kroku hlavního scénáře, jestliže uživatel zvolí, že chce splnit úvodní test, který určí jeho úroveň.
2. Zahrnutí: UC41 – Úvodní test. Systém zobrazí úvodní test a po vyplnění testu uživatelem odhadne jeho úroveň v daném jazyce.
3. Systém zapíše uživateli kurz a podle odhadnuté úrovně určí, kde v kurzu bude uživatel začínat. Následně přeměruje uživatele na hlavní stránku kurzu.

2.1.2.5.4 Alternativní scénář – úvodní dotazník

1. Scénář začíná v 5. bodě hlavního scénáře, pokud je daný kurz prvním kurzem, do kterého se uživatel zapisuje.
2. Systém zobrazí uživateli formulář, který slouží jako dotazník. Cílem tohoto formuláře je zjistit, jakým způsobem se uživatel o aplikaci dozvěděl.
3. Uživatel zvolí jednu z nabízených možností.
4. Systém odpověď uloží. Dále se pokračuje 6. krokem hlavního scénáře.

2.1.2.6 UC6 – Prohlížení mapy učení

Popis: Umožňuje uživateli prohlížet seznam lekcí v daném kurzu a získat tak přehled o obsahu kurzu. Lekce jsou seskupeny podle úrovní daného jazyka. Lekce jsou dále seřazeny a vyobrazeny způsobem připomínajícím mapu. Tento seznam lekcí budu dále v textu označovat jako mapu učení.

Priorita: Must have

Hlavní aktér: Uživatel

2.1.2.6.1 Hlavní scénář

1. Případ užití začíná, když si uživatel chce zobrazit mapu učení.
2. Systém zobrazí uživateli hlavní nabídku aplikace.
3. Uživatel zvolí úroveň jazyka, pro kterou chce mapu učení zobrazit.
4. Systém zobrazí mapu učení včetně znázornění, kde se uživatel v kurzu, respektive na mapě nachází. Společně s mapou systém také zobrazí možnost spustit denní učení.

2.1.2.7 UC7 – Změna pozice na mapě učení

Popis: Umožňuje uživateli změnit svoji pozici v kurzu, respektive na mapě učení. Díky tomu může uživatel určit, které lekce se mu budou zobrazovat při denním učení.

Priorita: Must have

Hlavní aktér: Uživatel

2.1.2.7.1 Hlavní scénář

1. Případ užití začíná, když uživatel chce změnit svoji pozici v rámci mapy učení.
2. Systém zobrazí uživateli mapu učení.
3. Uživatel vybere lekci, na kterou se chce přesunout a vybere možnost přesunutí se na danou lekci.
4. Systém přesune uživatele na danou lekci.

2.1.2.8 UC8 – Prohlížení lekcí podle jejich typu

- Popis:** Umožňuje uživateli prohlížet seznam lekcí kurzu podle jejich typu. Díky tomu si může uživatel například zobrazit přehled všech gramatických lekcí. V aplikaci se bude vyskytovat šest typů lekcí odpovídající gramatice, slovní zásobě, čtení, poslechu, výslovnosti a mluvení.
- Priorita:** Should have
- Hlavní aktér:** Uživatel

2.1.2.8.1 Hlavní scénář

1. Případ užití začíná, když si uživatel chce zobrazit seznam lekcí kurzu daného typu.
2. Systém zobrazí uživateli hlavní nabídku aplikace.
3. Uživatel zvolí typ lekcí, které chce zobrazit.
4. Systém zobrazí seznam lekcí daného typu seskupený podle úrovně daného jazyka.

2.1.2.9 UC9 – Zobrazení lekce

- Popis:** Umožňuje uživateli zobrazit informace o konkrétní lekci.
- Priorita:** Must have
- Hlavní aktér:** Uživatel

2.1.2.9.1 Hlavní scénář - mapa učení

1. Případ užití začíná, když si chce uživatel zobrazit informace o dané lekci.
2. Zahrnutí: UC6 – Prohlížení mapy učení. Systém zobrazí uživateli mapu učení.
3. Uživatel vybere lekci, kterou chce zobrazit.
4. Systém zobrazí uživateli informace o lekci. Tyto informace mohou zahrnovat název lekce a vysvětlení lekce ve formě textu nebo videa. Pokud se jedná o lekci slovní zásoby, zobrazí systém také seznam slovíček obsažený v dané lekci. Dále systém také zobrazí možnost spustit učení se dané lekce.

2.1.2.9.2 Alternativní scénář – zobrazení lekcí podle typu

1. Scénář začíná v 1. kroku hlavního scénáře.
2. Zahrnutí: UC8 – Prohlížení lekcí podle typu lekce. Systém zobrazí uživateli seznam lekcí podle daného typu.
3. Scénář pokračuje v 3. kroku hlavního scénáře.

2.1.2.9.3 Alternativní scénář – vyhledání lekce

1. Scénář začíná v 1. kroku hlavního scénáře.
2. Zahrnutí: UC39 – Vyhledávání lekcí, slovíček a uživatelů. Systém zobrazí uživateli seznam lekcí jako výsledek vyhledávání.
3. Scénář pokračuje v 3. kroku hlavního scénáře.

2.1.2.9.4 Alternativní scénář – oblíbené lekce

1. Scénář začíná v 1. kroku hlavního scénáře.
2. Zahrnutí: UC15 – Zobrazení oblíbených lekcí a slovíček. Systém zobrazí uživateli seznam oblíbených lekcí.
3. Scénář pokračuje v 3. kroku hlavního scénáře.

2.1.2.10 UC10 – Přidání lekce do oblíbených

Popis:	Umožňuje uživateli přidat danou lekci do seznamu oblíbených lekcí. Uživatel si tak může lekci uložit, aby k ní měl v budoucnu jednodušší přístup.
Priorita:	Should have
Hlavní aktér:	Uživatel

2.1.2.10.1 Hlavní scénář

1. Příklad užití začíná, když si chce uživatel na stránce lekce nebo při učení uložit lekci do seznamu oblíbených lekcí.
2. Systém zobrazí uživateli informace o lekci s možností přidání lekce do seznamu oblíbených lekcí.
3. Uživatel zvolí možnost přidání lekce do seznamu oblíbených lekcí.
4. Systém přidá lekci do seznamu oblíbených lekcí.

Analogickým způsobem může uživatel lekci ze seznamu odstranit.

2.1.2.11 UC11 – Vyřazení lekce z denního učení

Popis:	Umožňuje uživateli vyřadit lekci ze systému denního učení. Díky tomu může například vyřadit lekci, kterou se nechce učit.
Priorita:	Should have
Hlavní aktér:	Uživatel

2.1.2.11.1 Hlavní scénář

1. Příklad užití začíná, když chce uživatel na stránce lekce nebo při učení vyřadit danou lekci z denního učení.
2. Systém zobrazí uživateli informace o lekci s možností vyřazení lekce z denního učení.
3. Uživatel zvolí možnost vyřazení lekce z denního učení.
4. Systém vyřadí lekci z denního učení.

Analogickým způsobem může uživatel lekci opět zařadit do denního učení.

2.1.2.12 UC12 – Poskytnutí zpětné vazby k lekci

Popis:	Umožňuje uživateli poskytnout zpětnou vazbu k lekci a přispět tak ke zlepšení kvality kurzů v aplikaci.
Priorita:	Could have
Hlavní aktér:	Uživatel

2.1.2.12.1 Hlavní scénář

1. Příklad užití začíná, když chce uživatel poskytnout zpětnou vazbu ke konkrétní lekci.
2. Systém zobrazí uživateli informace o lekci s formulářem pro vyjádření, zda se uživateli lekce líbí nebo ne. V případě, že se uživateli lekce nelíbí, umožňuje formulář vyplnit důvod nespokojenosti.
3. Uživatel vyplní formulář s případným komentářem, proč je s lekcí nespokojený.
4. Systém uloží zpětnou vazbu uživatele.

2.1.2.13 UC13 – Zobrazení slovíčka

Popis: Umožňuje uživateli si zobrazit informace o daném slovíčku.
Priorita: Must have
Hlavní aktér: Uživatel

2.1.2.13.1 Hlavní scénář – zobrazení lekce

1. Příklad užití začíná, když si chce uživatel zobrazit informace o konkrétním slovíčku.
2. Zahrnutí: UC9 – Zobrazení lekce. Systém zobrazí uživateli lekci slovní zásoby se seznamem slovíček.
3. Uživatel vybere slovíčko, které chce zobrazit.
4. Systém zobrazí uživateli informace o daném slovíčku. To zahrnuje slovíčko v jazycích L1 a L2 daného kurzu, definici slovíčka, audio nahrávku správné výslovnosti a příklady užití slovíčka ve větách. Dále systém zobrazí, do jaké míry má uživatel slovíčko zapamatované.

2.1.2.13.2 Alternativní scénář – vyhledání slovíčka

1. Scénář začíná v 1. kroku hlavního scénáře.
2. Zahrnutí: UC39 – Vyhledávání lekcí, slovíček a uživatelů. Systém zobrazí uživateli seznam slovíček jako výsledek vyhledávání.
3. Scénář pokračuje v 3. kroku hlavního scénáře.

2.1.2.13.3 Alternativní scénář – oblíbená slovíčka

1. Scénář začíná v 1. kroku hlavního scénáře.
2. Zahrnutí: UC15 – Zobrazení oblíbených lekcí a slovíček. Systém zobrazí uživateli seznam oblíbených slovíček.
3. Scénář pokračuje v 3. kroku hlavního scénáře.

2.1.2.14 UC14 – Přidání slovíčka do oblíbených

Popis: Umožňuje uživateli přidat slovíčko do seznamu oblíbených slovíček. Uživatel si tak může slovíčko uložit, aby k němu měl v budoucnu jednodušší přístup.
Priorita: Should have
Hlavní aktér: Uživatel

2.1.2.14.1 Hlavní scénář

1. Případ užití začíná, když si chce uživatel na stránce slovíčka nebo při učení uložit slovíčko do seznamu oblíbených slovíček.
2. Systém zobrazí uživateli informace o slovíčku s možností přidání daného slovíčka do seznamu oblíbených slovíček.
3. Uživatel zvolí možnost přidání slovíčka do seznamu oblíbených slov.
4. Systém přidá slovíčko do seznamu oblíbených slov.

Analogickým způsobem může uživatel slovíčko ze seznamu odstranit.

2.1.2.15 UC15 – Zobrazení oblíbených lekcí a slovíček

Popis: Umožňuje uživateli zobrazit seznam slovíček a lekcí, které uživatel dříve označil jako oblíbené.

Priorita: Should have

Hlavní aktér: Uživatel

2.1.2.15.1 Hlavní scénář

1. Případ užití začíná, když si chce uživatel zobrazit seznam oblíbených slovíček a lekcí.
2. Systém zobrazí uživateli hlavní nabídku aplikace.
3. Uživatel zvolí, že chce zobrazit oblíbené lekce a slovíčka seznam oblíbených lekcí a slovíček.
4. Systém zobrazí seznam oblíbených slovíček a seznam oblíbených lekcí.

2.1.2.16 UC16 – Vytváření vlastních lekcí slovní zásoby

Popis: Umožňuje premium uživateli seskupovat slovíčka do vlastních lekcí. Díky tomu se pak premium uživatel může jednoduše učit vybraná slovíčka, která ho zajímají, nebo která se potřebuje naučit.

Priorita: Must have

Hlavní aktér: Premium uživatel

2.1.2.16.1 Hlavní scénář

1. Případ užití začíná, když chce premium uživatel vytvořit vlastní lekci slovíček.
2. Systém zobrazí seznam lekcí, které premium uživatel vytvořil s možností přidání nové lekce.
3. Premium uživatel zvolí možnost přidání nové lekce.
4. Systém zobrazí premium uživateli formulář pro vytvoření lekce. Formulář obsahuje pole pro zadání názvu lekce, popisu lekce a seznam slovíček, které byly do lekce přidány. Dále zobrazí možnost přidání nových slovíček do lekce. Nakonec také zobrazí možnost uložení lekce.
5. Premium uživatel vyplní položky formuláře a zvolí možnost přidání nových slovíček do lekce.
6. Systém zobrazí formulář pro přidání slov do lekce. Formulář se skládá z vyhledávacího pole a seznamu slov, které jsou výsledkem vyhledávání. U každého slova je pak zobrazena možnost slovo přidat do lekce. Pokud slovo již bylo přidáno, zobrazí se možnost slovo z lekce odebrat. Dále systém zobrazí možnost vrátit se zpět na předchozí formulář pro vytvoření lekce.

7. Premium uživatel pomocí formuláře vyhledá a přidá slova do lekce. Následně se vrátí zpět k formuláři pro vytvoření lekce a zvolí možnost uložení lekce.
8. Systém vytvoří lekci na základě informací, které premium uživatel ve formulářích vyplnil.

2.1.2.17 UC17 – Zobrazení vlastních lekcí slovní zásoby

Popis:	Systém umožňuje premium uživateli zobrazit seznam lekcí slovní zásoby, které vytvořil.
Priorita:	Must have
Hlavní aktér:	Premium uživatel

2.1.2.17.1 Hlavní scénář

1. Příklad užití začíná, když chce premium uživatel zobrazit seznam lekcí slovní zásoby, které vytvořil.
2. Systém zobrazí uživateli hlavní nabídku aplikace.
3. Uživatel zvolí možnost zobrazení vlastních lekcí.
4. Systém zobrazí uživateli seznam lekcí, které vytvořil.

2.1.2.18 UC18 – Úprava vlastní lekce slovíček

Popis:	Umožňuje premium uživateli upravit lekce slovíček, které dříve vytvořil.
Priorita:	Must have
Hlavní aktér:	Premium uživatel

2.1.2.18.1 Hlavní scénář

1. Příklad užití začíná, když chce premium uživatel upravit vlastní lekci slovíček.
2. Systém zobrazí seznam lekcí, které premium uživatel vytvořil. U každé lekce zobrazí možnost pro úpravu lekce.
3. Premium uživatel zvolí možnost pro úpravu lekce.
4. Systém zobrazí premium uživateli formulář pro úpravu lekce. Formulář obsahuje pole pro úpravu názvu lekce, popisu lekce a seznam slovíček, které byly do lekce přidány. Dále zobrazí možnost přidání nových slovíček do lekce, možnost odstranění lekce a možnost uložení lekce.
5. Premium uživatel dle potřeby upraví položky formuláře a zvolí možnost přidání nových slovíček do lekce.
6. Systém zobrazí formulář pro přidání slov do lekce. Formulář se skládá z vyhledávacího pole a seznamu slov, které jsou výsledkem vyhledávání. U každého slova je pak zobrazena možnost slovo přidat do lekce. Pokud slovo již bylo přidáno, zobrazí se možnost slovo z lekce odebrat. Dále systém zobrazí možnost vrátit se zpět na předchozí formulář pro úpravu lekce.
7. Premium uživatel pomocí formuláře dle potřeby vyhledá a přidá slova do lekce. Následně se vrátí zpět do formuláře pro úpravu lekce a zvolí možnost uložení lekce.
8. Systém upraví lekci na základě informací, které premium uživatel ve formulářích vyplnil.

2.1.2.19 UC19 – Odstranění vlastní lekce slovíček

Popis:	Umožňuje premium uživateli odstranit lekci slovíček, kterou dříve vytvořil.
Priorita:	Must have
Hlavní aktér:	Premium uživatel

2.1.2.19.1 Hlavní scénář

1. Příklad užití začíná, když chce premium uživatel odstranit vlastní lekci slovíček.
2. Systém zobrazí seznam lekcí, které premium uživatel vytvořil. U každé lekce zobrazí možnost pro úpravu lekce.
3. Premium uživatel zvolí možnost pro úpravu lekce.
4. Systém zobrazí premium uživateli formulář pro úpravu lekce s možností odstranění dané lekce.
5. Premium uživatel zvolí možnost odstranění lekce.
6. Systém upozorní uživatele, že je tato akce nevratná a umožní uživateli potvrdit svoji volbu, anebo se vrátit zpět.
7. Premium uživatel potvrdí svoji volbu.
8. Systém odstraní lekci.

2.1.2.20 UC20 – Přepnutí aktivního kurzu

Popis:	Umožňuje uživateli přepnout kurz, který se učí. Kurz, který se uživatel učí bude dále v textu nazývat aktivním kurzem.
Priorita:	Must have
Hlavní aktér:	Uživatel

2.1.2.20.1 Hlavní scénář

1. Příklad užití začíná, když chce uživatel přepnout kurz, který se učí.
2. Systém zobrazí seznam kurzů, do kterých je uživatel zapsán.
3. Uživatel vybere kurz, který se chce učit.
4. Systém nastaví daný kurz jako aktivní a přesměruje uživatele na hlavní stránku tohoto kurzu.

2.1.2.21 UC21 – Zobrazení nápovědy

Popis:	Umožňuje uživateli si zobrazit nápovědu, kde může nalézt odpovědi na základní otázky ohledně aktivního kurzu a používání aplikace.
Priorita:	Could have
Hlavní aktér:	Uživatel

2.1.2.21.1 Hlavní scénář

1. Případ užití začíná, když si chce uživatel zobrazit nápovědu a získat tak odpověď na jeho dotaz.
2. Systém zobrazí hlavní nabídku aplikace s možností zobrazení nápovědy.
3. Uživatel zvolí možnost zobrazení nápovědy.
4. Systém zobrazí jednotlivé kategorie nápovědy.
5. Uživatel zvolí kategorii, která se jeho problému týká.
6. Systém zobrazí seznam článků ve zvolené kategorii. Každý článek typicky pokrývá jeden dotaz nebo problém, který chce uživatel vyřešit.
7. Uživatel zvolí článek odpovídající jeho dotazu.
8. Systém zobrazí článek obsahující odpovídající nápovědu.

2.1.2.22 UC22 – Změna nastavení

Popis: Umožňuje uživateli změnit nastavení svého účtu a aktivního kurzu.
Priorita: Must have
Hlavní aktér: Uživatel

2.1.2.22.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel změnit své uživatelské jméno, zobrazované jméno nebo emailovou adresu v rámci celé aplikace, nebo když chce změnit svůj denní cíl v rámci aktivního kurzu.
2. Systém zobrazí formulář sloužící k úpravě nastavení. Formulář obsahuje pole pro změnu uživatelského jména, zobrazovaného jména nebo emailové adresy. Dále také obsahuje možnost změny denního cíle v rámci aktivního kurzu. S formulářem je také zobrazena možnost uložit změny, které uživatel v nastavení provedl.
3. Uživatel změní dle potřeby údaje ve formuláři a zvolí možnost pro uložení změn.
4. Systém uloží zadané změny.

2.1.2.23 UC23 – Změna hesla

Popis: Umožňuje uživateli změnit své heslo.
Priorita: Must have
Hlavní aktér: Uživatel

2.1.2.23.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel změnit své heslo.
2. Systém zobrazí formulář sloužící k úpravě nastavení včetně možnosti požádání o změnu hesla.
3. Uživatel zvolí možnost požádání o změnu hesla.
4. Systém uživateli odešle na emailovou adresu email s odkazem pro změnu hesla a informuje uživatele, že mu byl email odeslán.

5. Uživatel otevře odkaz v emailu a přesune se na danou webovou stránku.
6. Systém na stránce zobrazí uživateli formulář pro změnu hesla. Formulář vyžaduje zadání nového hesla a zopakování tohoto hesla, aby se ověřilo, že bylo heslo zadáno správně.
7. Uživatel vyplní formulář a potvrdí změnu hesla.
8. Systém změní uživateli heslo a zobrazí mu informaci, že bylo heslo změněno.

2.1.2.24 UC24 – Přizpůsobení témat kurzu

Popis:	Umožňuje uživateli přizpůsobit aktivní kurz vybráním témat, která se mají při učení zobrazovat. Uživatel může jednotlivá témata vypínat a zapínat. Vypnutím tématu se z denního učení vyřadí všechny lekce, které jsou v daném tématu obsaženy.
Priorita:	Should have
Hlavní aktér:	Uživatel

2.1.2.24.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel zapnout nebo vypnout některá z témat aktivního kurzu.
2. Systém zobrazí formulář sloužící k úpravě nastavení včetně možnosti přizpůsobení témat aktivního kurzu. Společně s formulářem je také zobrazena možnost uložit provedené změny.
3. Uživatel vybere možnost přizpůsobení témat kurzu.
4. Systém zobrazí uživateli seznam všech dostupných témat v aktivním kurzu. Tato témata jsou pro přehlednost rozdělena do dvou kategorií. První kategorie obsahuje hlavní témata kurzu. Druhá kategorie obsahuje pokročilá nebo odborná témata, která nejsou ve výchozím nastavení kurzu zapnutá. Dále systém zobrazí možnost vrátit se zpět k předchozímu formuláři.
5. Uživatel dle potřeby zapne nebo vypne jednotlivá témata. Následně zvolí možnost pro návrat k předchozímu formuláři, kde pak provedené změny uloží.
6. Systém přizpůsobí kurz podle předchozího vstupu uživatele a uloží provedené změny.

2.1.2.25 UC25 – Zobrazení profilu uživatele

Popis:	Umožňuje uživateli zobrazit svůj profil.
Priorita:	Should have
Hlavní aktér:	Uživatel

2.1.2.25.1 Hlavní scénář

1. Případ užití začíná, když si chce uživatel zobrazit svůj profil.
2. Systém zobrazí uživateli hlavní nabídku aplikace s možností zobrazení uživatelského profilu.
3. Uživatel vybere možnost zobrazení uživatelského profilu.
4. Systém zobrazí uživateli jeho profil. Ten obsahuje uživatelské jméno, zobrazované jméno, úroveň uživatele, profilový obrázek uživatele a počet sledujících a sledovaných uživatelů. Dále systém zobrazí statistiku, kolik cvičení uživatel splnil za poslední týden a jeho denní řadu. Systém také zobrazí seznam ocenění získaných uživatelem.

2.1.2.26 UC26 – Zobrazení profilu jiného uživatele

Popis: Umožňuje uživateli zobrazit profil jiného uživatele.
Priorita: Should have
Hlavní aktér: Uživatel

2.1.2.26.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel zobrazit uživatelský profil jiného uživatele.
2. Zahrnutí: UC39 – Vyhledávání lekcí, slovíček a uživatelů. Systém zobrazí seznam uživatelů, které uživatel vyhledal.
3. Uživatel zvolí jednoho z vyhledaných uživatelů.
4. Systém zobrazí profil daného uživatele. Ten obsahuje uživatelské jméno, zobrazované jméno, úroveň uživatele, profilový obrázek uživatele a počet sledujících a sledovaných uživatelů. Dále systém poskytne možnost začít nebo přestat sledovat daného uživatele a zobrazit seznam sledovaných a sledujících uživatelů. Systém také zobrazí statistiku, kolik cvičení uživatel splnil za poslední týden a jeho denní řadu. Systém dále zobrazí seznam ocenění získaných uživatelem.

2.1.2.27 UC27 – Sledování uživatelů

Popis: Umožňuje uživateli začít nebo přestat sledovat jiného uživatele. Systém následně uživateli průběžně zobrazuje novinky o uživatelích, které sleduje.
Priorita: Should have
Hlavní aktér: Uživatel

2.1.2.27.1 Hlavní scénář – využití profilu uživatele

1. Případ užití začíná, když chce uživatel začít nebo přestat sledovat jiného uživatele.
2. Zahrnutí: UC26 – Zobrazení profilu jiného uživatele.
3. Uživatel zvolí možnost, pomocí které začne nebo přestane sledovat daného uživatele.
4. Systém uloží volbu uživatele.

2.1.2.27.2 Alternativní scénář – využití seznamu sledovaných a sledujících uživatelů

1. Případ užití začíná, když chce uživatel začít nebo přestat sledovat jiného uživatele.
2. Zahrnutí: UC28 – Zobrazení sledovaných a sledujících uživatelů. Systém zobrazí seznam sledujících nebo sledovaných uživatelů na profilu libovolného uživatele.
3. Uživatel zvolí možnost, kterou začne nebo přestane sledovat daného uživatele ze seznamu.
4. Systém uloží volbu uživatele.

2.1.2.28 UC28 – Zobrazení sledovaných a sledujících uživatelů

Popis: Umožňuje uživateli zobrazit seznam sledujících a seznam sledovaných uživatelů na profilu libovolného uživatele.
Priorita: Could have
Hlavní aktér: Uživatel

2.1.2.28.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel zobrazit seznam sledujících nebo sledovaných uživatelů na profilu libovolného uživatele.
2. Zahrnutí: UC26 – Zobrazení profilu jiného uživatele. Systém zobrazí profil daného uživatele.
3. Uživatel zvolí možnost pro zobrazení sledujících nebo sledovaných uživatelů.
4. Systém zobrazí uživateli seznam sledovaných nebo sledujících uživatelů. U každého uživatele systém také zobrazí možnost začít nebo přestat sledovat daného uživatele.

2.1.2.29 UC29 – Zobrazení novinek

- Popis:** Umožňuje uživateli zobrazit novinky v aplikaci. Novinky mohou obsahovat oznámení o dosažení úspěchu některým ze sledovaných uživatelů. Dále mohou obsahovat odkazy na články a videa, které administrátor vložil do aplikace.
- Priorita:** Could have
- Hlavní aktér:** Uživatel

2.1.2.29.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel zobrazit seznam novinek.
2. Systém zobrazí uživateli hlavní nabídku aplikace s možností zobrazení novinek.
3. Uživatel zvolí možnost zobrazení novinek.
4. Systém zobrazí seznam novinek. U novinek, které jsou oznámením o dalších uživatelích systém umožňuje zareagovat na dané oznámení jedním z nabízených emotikonů. Tím může uživatel například vyjádřit gratulaci jinému uživateli k dosažení určitého úspěchu.

2.1.2.30 UC30 – Reagování na novinky

- Popis:** Umožňuje uživatelům v seznamu novinek reagovat na oznámení o úspěších ostatních uživatelů pomocí emotikonů.
- Priorita:** Could have
- Hlavní aktér:** Uživatel

2.1.2.30.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel zareagovat na určité oznámení o jiném uživateli v seznamu novinek.
2. Zahrnutí: UC29 – Zobrazení novinek. Systém zobrazí seznam novinek.
3. Uživatel zvolí možnost zareagování na oznámení o úspěchu určitého uživatele. Dále vybere jednu z nabízených reakcí pomocí emotikonu.
4. Systém reakci uloží a zobrazí ji u daného oznámení.

2.1.2.31 UC31 – Zakupování položek

Popis:	Umožňuje uživateli v obchodu za virtuální měnu koupit danou virtuální položku. Aplikace bude takto umožňovat nakupování profilových obrázků. Obchod v aplikaci by měl být rozšiřitelný a připravený na přidání dalších druhů položek.
Priorita:	Should have
Hlavní aktér:	Uživatel

2.1.2.31.1 Hlavní scénář

1. Příklad užití začíná, když chce uživatel za virtuální měnu zakoupit danou položku v obchodě.
2. Systém zobrazí uživateli hlavní nabídku aplikace s možností zobrazení obchodu.
3. Uživatel zvolí možnost zobrazení obchodu.
4. Systém zobrazí uživateli položky obchodu a virtuální měnu, kterou má uživatel k dispozici. Pokud uživatel nemá danou položku zakoupenou, zobrazí systém u položky cenu. V opačném případě zobrazí, zdali uživatel danou položku používá.
5. Uživatel vybere položku, kterou chce zakoupit.
6. Systém zobrazí uživateli detaily položky s možností potvrzení zakoupení.
7. Uživatel potvrdí zakoupení položky.
8. Systém volbu uloží a zobrazí uživateli, že je položka zakoupená.

2.1.2.32 UC32 – Používání zakoupených položek

Popis:	Umožňuje uživateli v obchodu použít již zakoupenou položku.
Priorita:	Should have
Hlavní aktér:	Uživatel

2.1.2.32.1 Hlavní scénář

1. Příklad užití začíná, když chce uživatel použít již zakoupenou položku.
2. Systém zobrazí uživateli hlavní nabídku aplikace s možností zobrazení obchodu.
3. Uživatel zvolí možnost zobrazení obchodu.
4. Systém zobrazí uživateli položky obchodu a virtuální měnu, kterou má uživatel k dispozici. Pokud uživatel nemá danou položku zakoupenou, zobrazí systém u položky cenu. V opačném případě zobrazí, zdali uživatel používá danou položku.
5. Uživatel vybere zakoupenou položku, kterou chce použít.
6. Systém zobrazí uživateli detaily položky s možností potvrzení použití položky.
7. Uživatel potvrdí použití položky.
8. Systém volbu uloží a zobrazí uživateli, že je položka používaná.

2.1.2.33 UC33 – Zobrazení výzev a ocenění

- Popis:** Umožňuje uživateli zobrazit seznam ocenění a probíhajících výzev. Systém uděluje uživatelům ocenění, pokud daný uživatel dosáhne desáté, padesáté nebo sté úrovně. Aplikace by měla být do budoucna rozšiřitelná o další druhy ocenění. Výzvy v systému vytváří administrátor typicky na omezenou časovou dobu. Cílem výzvy je zpravidla dokončit v daném období určitou skupinu lekcí. Za dosažení ocenění nebo dokončení výzvy si může uživatel vyzvednout odměnu ve formě virtuální měny.
- Priorita:** Could have
- Hlavní aktér:** Uživatel

2.1.2.33.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel zobrazit seznam ocenění nebo probíhající výzvy.
2. Systém zobrazí uživateli hlavní nabídku aplikace s možností zobrazení seznamu ocenění a výzev.
3. Uživatel zvolí možnost zobrazení seznamu ocenění a výzev.
4. Systém zobrazí seznam ocenění a výzev. V případě, že uživatel dosáhl ocenění nebo splnil výzvu, může si u dané položky vyzvednout odměnu ve formě virtuální měny.

2.1.2.34 UC34 – Zobrazování oznámení

- Popis:** Umožňuje uživateli si na hlavní stránce kurzu zobrazit oznámení. Jednotlivé druhy oznámení jsou popsány níže.
- Priorita:** Must have
- Hlavní aktér:** Uživatel

2.1.2.34.1 Hlavní scénář

1. Případ užití začíná, když si uživatel zobrazí hlavní stránku aktivního kurzu.
2. Systém zobrazí uživateli všechna nová oznámení, která uživatel ještě neviděl.

2.1.2.34.2 Druhy oznámení

V aplikaci budou využívány následující druhy oznámení.

Statistika o učení Po každém učení se uživateli zobrazí oznámení s přehledem, v kolika cvičeních uživatel odpověděl správně a v kolika chyboval.

Dotazník Systém po určitém čase používání aplikace zobrazí uživateli jednoduchý dotazník, který má za cíl zjistit, jaká je pravděpodobnost, že by uživatel doporučil aplikaci ostatním lidem.

Reklama Po dokončení každého učení se uživateli zobrazí oznámení s reklamou. Toto oznámení se nebude zobrazovat premium uživatelům.

Získání odměny Pokud uživatel v aplikaci získá odměnu ve formě virtuální měny, zobrazí se mu oznámení.

Dosažení úspěchu Pokud uživatel v aplikaci dosáhne libovolného úspěchu v daném kurzu, zobrazí se mu oznámení.

Zkušební verze premium účtu Systém jednou za daný čas zobrazí uživateli oznámení propagující zkušební verzi premium účtu s možností spuštění zkušební verze.

Skončení zkušební verze premium účtu Po uplynutí doby zkušební verze premium účtu se uživateli zobrazí oznámení, že mu zkušební doba skončila.

2.1.2.35 UC35 – Zobrazení informací o premium účtu

Popis: Umožňuje uživateli si zobrazit informace o výhodách předplatného aplikace.
Priorita: Must have
Hlavní aktér: Uživatel

2.1.2.35.1 Hlavní scénář

1. Případ užití začíná, když si chce uživatel zobrazit informace o výhodách předplatného.
2. Systém zobrazí uživateli hlavní nabídku aplikace s možností zobrazení stránky s informacemi o předplatném.
3. Uživatel vybere možnost zobrazení informací o předplatném.
4. Systém zobrazí uživateli stránku se seznamem výhod předplatného s možností přihlášení se k předplatnému. Pokud uživatel již má aktivní předplatné, zobrazí se místo této možnosti možnost spravovat předplatné.

2.1.2.36 UC36 – Přihlášení k předplatnému

Popis: Umožňuje uživateli se přihlásit k předplatnému a stát se tak premium uživatelem.
Priorita: Must have
Hlavní aktér: Uživatel

2.1.2.36.1 Hlavní scénář

1. Případ užití začíná, když si chce uživatel aktivovat předplatné v aplikaci.
2. Zahrnutí: UC35 – Zobrazení informací o premium účtu. Systém zobrazí uživateli informace o předplatném s možností přihlášení se k předplatnému.
3. Uživatel zvolí možnost přihlášení se k předplatnému.
4. Systém zobrazí informace o ceně předplatného a formulář pro vyplnění platebních údajů. Dále zobrazí možnost požádání o fakturu. V takovém případě musí uživatel vyplnit dodatečné údaje potřebné k vyplnění faktury. Na konci formuláře systém zobrazí možnost potvrdit přihlášení k předplatnému.
5. Uživatel vyplní formulář a potvrdí přihlášení k předplatnému.
6. Systém zajistí zpracování platby a přihlášení uživatele k předplatnému. Následně změní roli uživatele na premium uživatele. Nakonec zobrazí uživateli informaci, že mu bylo předplatné aktivováno. Pokud uživatel požádal o fakturu, systém mu ji zašle na emailovou adresu.

2.1.2.36.2 Alternativní scénář – požádání o fakturu

1. Scénář začíná ve 4. kroku hlavního scénáře, jestliže uživatel zvolí, že chce požádat o fakturu.
2. Systém zobrazí uživateli dodatečnou část formuláře pro vyplnění celého jména uživatele a jeho adresy.
3. Uživatel vyplní do formuláře potřebné údaje.
4. Scénář pokračuje v 5. bodě hlavního scénáře.

2.1.2.37 UC37 – Zobrazení stavu předplatného

Popis:	Umožňuje premium uživateli si zobrazit informace o probíhajícím předplatném.
Priorita:	Must have
Hlavní aktér:	Premium uživatel

2.1.2.37.1 Hlavní scénář

1. Případ užití začíná, když si chce premium uživatel zobrazit stav probíhajícího předplatného.
2. Zahrnutí: UC35 – Zobrazení informací o premium účtu. Systém zobrazí informace o předplatném s možností spravování předplatného.
3. Premium uživatel zvolí možnost spravování předplatného.
4. Systém zobrazí informace o probíhajícím předplatném včetně stavu předplatného a data nadcházející platby. Dále systém zobrazí možnost zrušit předplatné.

2.1.2.38 UC38 – Odhlášení předplatného

Popis:	Umožňuje premium uživateli zrušit předplatné.
Priorita:	Must have
Hlavní aktér:	Premium uživatel

2.1.2.38.1 Hlavní scénář

1. Případ užití začíná, když chce premium uživatel zrušit probíhající předplatné.
2. Zahrnutí: UC37 – Zobrazení stavu předplatného. Systém zobrazí informace o probíhajícím předplatném s možností zrušení předplatného.
3. Premium uživatel vybere možnost zrušení předplatného.
4. Systém zobrazí uživateli formulář, který slouží jako dotazník. Cílem tohoto formuláře je zjistit důvod, proč chce premium uživatel předplatné zrušit.
5. Premium uživatel vybere jednu z nabízených možností a potvrdí svou volbu.
6. Systém zobrazí informace o funkcích, ke kterým premium uživatel ztratí přístup po zrušení předplatného. Dále zobrazí možnosti ponechat předplatné a potvrdit zrušení.
7. Premium uživatel zvolí možnost potvrdit zrušení.
8. Systém zruší uživateli předplatné a zobrazí mu informaci, že bylo předplatné zrušeno. Premium uživatel bude mít nadále přístup k funkcím premium účtu až do data, kdy by měla proběhnout další platba. Po dovršení tohoto data bude premium uživateli změněna role na uživatele, čímž ztratí přístup k některým funkcím aplikace.

2.1.2.39 UC39 – Vyhledávání lekcí, slovíček a uživatelů

Popis:	Umožňuje uživateli vyhledat libovolnou lekci nebo slovíčko v aktivním kurzu. Dále umožňuje vyhledat ostatní uživatele v rámci celé aplikace.
Priorita:	Should have
Hlavní aktér:	Uživatel

2.1.2.39.1 Hlavní scénář

1. Případ užití začíná, když chce uživatel vyhledat lekci, slovíčko nebo libovolného uživatele.
2. Systém zobrazí uživateli hlavní nabídku aplikace s možností vyhledávání.
3. Uživatel vybere možnost vyhledávání.
4. Systém zobrazí uživateli vyhledávací pole a možnost spustit vyhledání.
5. Uživatel vyplní do vyhledávacího pole název lekce, slovíčko, uživatelské jméno nebo zobrazované jméno daného uživatele a spustí vyhledávání.
6. Systém zobrazí výsledky vyhledávání se seznamy odpovídajících lekcí, slovíček a uživatelů. Jednotlivé položky seznamů následně umožňují se v aplikaci přesunout na danou lekci, slovíčko, nebo profil uživatele.

2.1.2.40 UC40 – Učení a opakování

Popis:	Umožňuje uživateli se učit a opakovat si látku aktivního kurzu.
Priorita:	Must have
Hlavní aktér:	Uživatel

2.1.2.40.1 Hlavní scénář - denní učení

1. Případ užití začíná, když se chce uživatel učit novou látku kurzu a zopakovat si látku, kterou se již dříve učil.
2. Zahrnutí: UC6 – Prohlížení mapy učení. Systém zobrazí uživateli mapu učení s možností spuštění denního učení.
3. Uživatel vybere možnost spuštění denního učení.
4. Systém vybere nejvhodnější cvičení pro učení se nové látky a zopakování látky, kterou se uživatel dříve učil. Tato cvičení systém postupně uživateli zobrazí. Jednotlivé druhy cvičení jsou sepsány níže. Pokud systém uživateli zobrazí cvičení zaměřené na gramatický jev, se kterým se uživatel v aplikaci dříve nesetkal, zobrazí mu před daným cvičením vysvětlení odpovídajícího gramatického jevu. Pokud systém zobrazí cvičení zaměřené na slovíčko, které se uživatel ještě neučil, zobrazí mu před daným cvičením definici tohoto slovíčka.
5. Uživatel postupně splní všechna cvičení.
6. Systém vyhodnotí odpovědi uživatele a odkáže ho na hlavní stránku kurzu. Za splněná cvičení také připíše uživateli odměnu ve formě virtuální měny.
7. Zahrnutí: UC34 – Zobrazování oznámení. Systém zobrazí uživateli oznámení se statistikou, kolik cvičení vyplnil správně a v kolika chyboval.

2.1.2.40.2 Alternativní scénář – učení se konkrétní lekce

1. Příklad užití začíná, když se chce uživatel naučit konkrétní lekci, nebo si ji zopakovat.
2. Zahrnutí: UC9 – Zobrazení lekce. Systém zobrazí uživateli lekci s možností spuštění učení dané lekce.
3. Uživatel vybere možnost spuštění učení.
4. Systém vybere nejvhodnější cvičení pro učení se dané lekce, nebo pro její zopakování. Tato cvičení systém postupně uživateli zobrazí. Jednotlivé druhy cvičení jsou sepsány níže. Pokud systém uživateli zobrazí cvičení zaměřené na gramatický jev, se kterým se uživatel dříve v aplikaci nesetkal, zobrazí mu před cvičením vysvětlení daného jevu. Pokud systém zobrazí cvičení zaměřené na slovíčko, které se uživatel ještě neučil, zobrazí mu před daným cvičením definici tohoto slovíčka.
5. Scénář pokračuje v 5. kroku hlavního scénáře.

2.1.2.40.3 Druhy cvičení V rámci učení a opakování systém vybírá z následujících druhů cvičení.

Sestavení slova z písmen Uživateli je zobrazeno slovo v jazyce L1. Cílem cvičení je sestavit dané slovo v jazyce L2 s pomocí nabízených písmen. Cvičení tak učí uživatele napsat správně dané slovíčko. Cvičení může být doplněno obrázkem, který odpovídá danému slovu.

Doplnění slov do věty Uživateli je zobrazena věta v jazyce L1 a věta v jazyce L2 s chybějícími slovy. Cílem cvičení je doplnit do věty chybějící slova z nabídky. Toto cvičení se například používá pro procvičení gramatiky. Cvičení může být doplněno obrázkem znázorňujícím danou větu.

Překlad textu Uživateli je zobrazena věta nebo slovo v jazyce L1. Cílem cvičení je text přepsat do jazyka L2. Toto cvičení se typicky používá na procvičení gramatiky.

Doplnění tabulky Uživateli je zobrazena tabulka s chybějícími buňkami a zadání cvičení. Uživatel má za úkol do tabulky doplnit chybějící data dle zadání. Toto cvičení se využívá například pro trénink skloňování nebo časování některých slov.

Spojení slova a audio nahrávky Uživateli jsou zobrazeny fráze či slova v jazyce L2 a jim odpovídající audio nahrávky. Uživatel má za úkol jednotlivé položky spojit a vytvořit tak odpovídající dvojice.

Spojení slova a obrázku Uživateli jsou zobrazeny fráze či slova v jazyce L2 a jim odpovídající obrázky. Uživatel má za úkol jednotlivé položky spojit a vytvořit tak odpovídající dvojice.

Spojení slov Uživateli jsou zobrazeny fráze či slova v jazyce L2 a jim odpovídající překlady v jazyce L1. Uživatel má za úkol jednotlivé položky spojit a vytvořit tak odpovídající dvojice.

Krátký poslech Uživatel si poslechne krátkou audio nahrávku slova nebo věty v L2 a do textového pole napíše text, který odpovídá nahrávce. Toto cvičení učí uživatele porozumět větám a slovíčkům v jejich mluvené podobě.

Dlouhý poslech Uživatel si poslechne audio nahrávku v L2 a vyplní do textového pole odpověď na danou otázku. Tento typ cvičení se používá k procvičení delšího poslechu, kterým může být například nahraná konverzace.

Čtení Uživateli je zobrazen text v jazyce L2 a textové pole s otázkou. Uživatel má za úkol si text přečíst a doplnit do textového pole odpověď na danou otázku.

Mluvení Uživateli je zobrazeno téma, podotázky a časovač. Uživatel má za úkol sám mluvit o daném tématu po danou dobu. Cílem cvičení je natrénovat improvizované mluvení o tématu a odpovědět na všechny podotázky. Systém nijak nehodnotí ani nenahrává mluvu uživatele. Uživatel může cvičení kdykoliv označit za hotové a pokračovat tak v dalším učení nebo opakování.

Výslovnost podle textu Uživateli se zobrazí text v L2, který má za úkol přečíst nahlas do mikrofonu. Systém následně ohodnotí výslovnost uživatele.

Výslovnost podle audia Uživatel si poslechne krátkou audio nahrávku slovíčka nebo věty v L2. Poté má za úkol slovo nebo větu zopakovat nahlas do mikrofonu. Systém následně ohodnotí výslovnost uživatele.

2.1.2.41 UC41 – Úvodní test

Popis:	Umožňuje uživateli v daném kurzu splnit úvodní test, díky kterému systém odhadne jazykovou úroveň uživatele.
Priorita:	Must have
Hlavní aktér:	Uživatel

2.1.2.41.1 Hlavní scénář

1. Příklad užití začíná, když se uživatel rozhodne splnit úvodní test.
2. Systém zobrazí uživateli sérii předem určených cvičení. Druhy cvičení jsou identické s druhy cvičení v UC40 – Učení a opakování.
3. Uživatel postupně splní všechna cvičení.
4. Systém vyhodnotí odpovědi uživatele a odhadne jeho úroveň v daném jazyce.

2.2 Návrh uživatelského rozhraní

Na základě dříve popsaných případů užití jsem pomocí nástroje Figma vytvořil návrh uživatelského rozhraní¹. Při návrhu jsem zvolil přístup *mobile-first* [11]. Tento přístup se soustředí primárně na vytvoření kvalitního uživatelského rozhraní pro mobilní zařízení. To například zahrnuje vytváření jednodušších formulářů, používání méně textu a celkově jednodušší návrh uživatelského rozhraní. Volba barev, fontů a dalších grafických prvků je v rámci *mobile-first* designu také primárně přizpůsobena mobilním zařízením. Aplikace bude samozřejmě plně funkční i na ostatních zařízeních, ovšem návrh se bude především zaměřovat na mobilní zařízení.

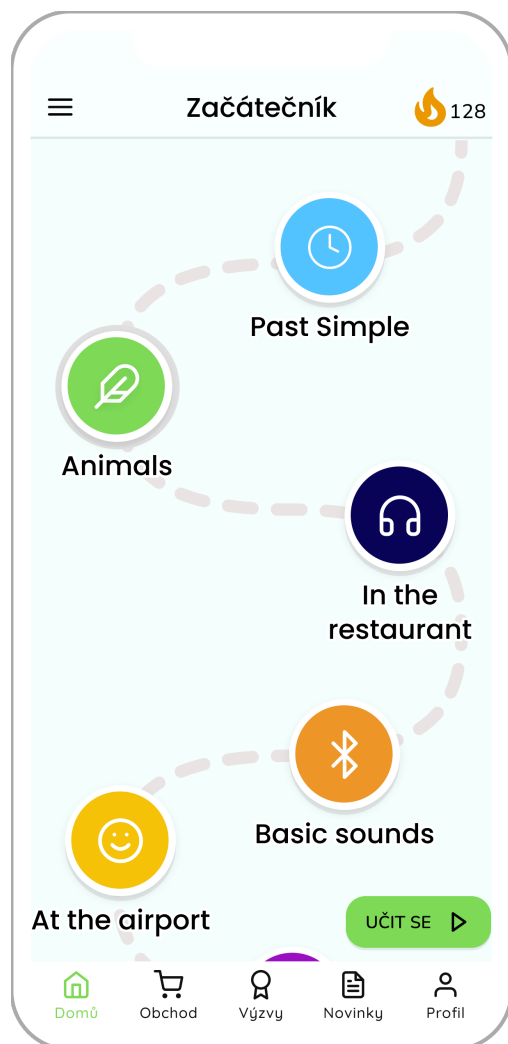
Jelikož bude celá frontendová část aplikace responzivní a jednotlivé komponenty a rozložení stránek se nebudou na různých obrazovkách příliš lišit, rozhodl jsem se vytvořit návrh pouze pro mobilní zařízení.

Nástroj Figma poskytuje možnost vytváření vlastních komponentů a stylů [12], které lze následně opakovaně používat. Toho jsem v rámci návrhu využil, což značně zjednodušilo práci

¹Dostupné na: <https://www.figma.com/file/wu3UCREJ36lT6BSwDVwkJb>

s jednotlivými prvky uživatelského rozhraní. Mezi navrženými stránkami jsem také následně vytvořil jednoduché interakce. Díky nim bylo jednodušší návrh prezentovat při schůzkách a konzultacích.

Pro ilustraci jsem vybral několik stránek z navrženého uživatelského rozhraní. Tyto stránky lze najít na obrázcích 2.1, 2.2, 2.3 a 2.4.



■ Obrázek 2.1 Hlavní stránka kurzu



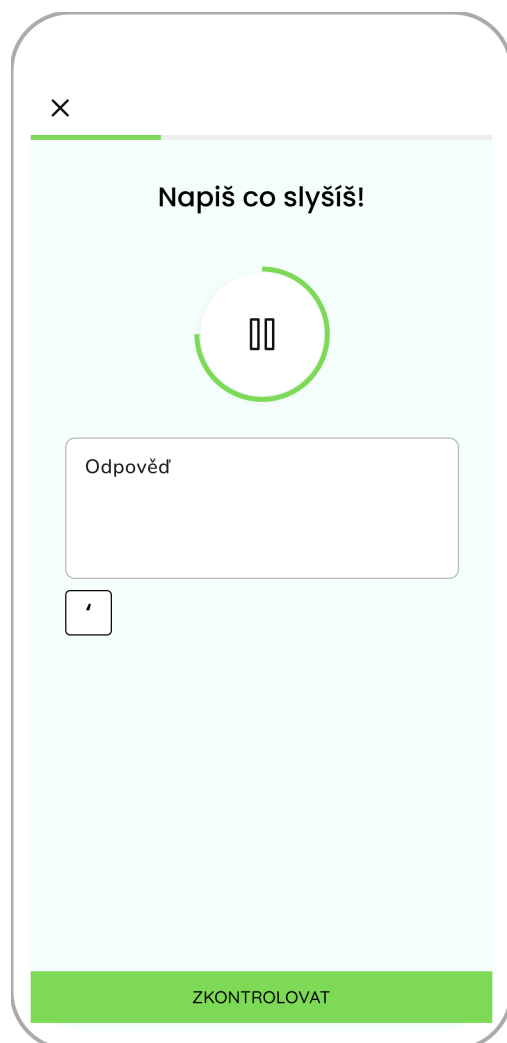
■ Obrázek 2.2 Stránka zápisu do kurzu

2.3 Výběr technologií

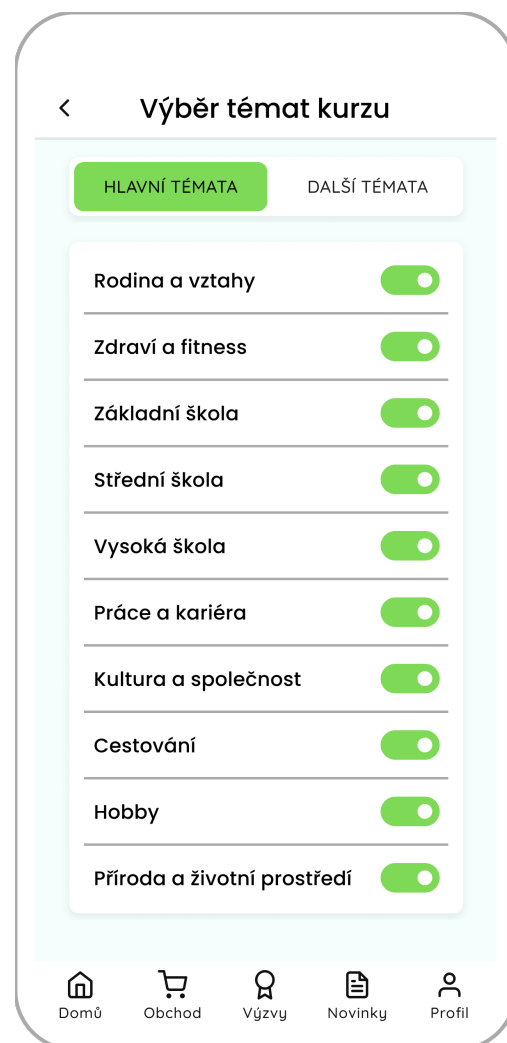
V této části kapitoly se zaměřím primárně na výběr frameworku a jazyka, pomocí kterých budu implementovat frontendovou část aplikace. Tento výběr ovlivní směr celého projektu a proto mu budu věnovat větší pozornost.

Následně provedu výběr grafické knihovny, která bude obsahovat předpřipravené prvky uživatelského rozhraní. Tato knihovna bude sloužit jako základ pro sestavení uživatelského rozhraní.

Dále pak provedu výběr programů, které umožní vytvoření mocku REST API, vytvoření návrhu uživatelského rozhraní a správu požadavků projektu.



■ **Obrázek 2.3** Poslechové cvičení



■ **Obrázek 2.4** Stránka s volbou témat kurzu

Nakonec také vyberu podstatné knihovny, které budu při implementaci využívat. Tyto knihovny například zajistí sledování chyb v nasazené aplikaci, nebo kontrolování kvality kódu.

2.3.1 Framework a jazyky

Do analýzy jsem se rozhodl zahrnout čtyři nejpoblárnější frameworky [13], které se pro tvorbu webových aplikací používají. Mezi ně patří React, Angular, Next.js a Vue.js.

Mezi nejpoužívanější webové technologie patří také ASP.NET Core. Tuto technologii využívá backend tohoto projektu, z čehož přirozeně vyplývá možnost využít ASP.NET Core i ve front-endové části. Konkrétně se nabízí využití frameworku Blazor, který je součástí ASP.NET Core a umožňuje vývoj webových aplikací v jazyce C#. Díky tomu by se zachovala konzistence používaných jazyků napříč celým projektem. Proto jsem se rozhodl Blazor do analýzy zařadit, přestože není zdaleka tak používaný, jako jeho výše zmínění konkurenti.

Dále je vhodné poznamenat, že knihovna Next.js je nadstavbou frameworku React. Proto budu v rámci analýzy brát v potaz, že použití Next.js by také znamenalo pro vývoj použít knihovnu React.

Frameworky jsem se rozhodl porovnat podle jejich licencí, možnosti vývoje pro více platform, míry používání, počtu dostupných knihoven, oblíbenosti, obtížnosti osvojení a počtu dostupných vývojářů na trhu.

U jednotlivých kritérií jsem se rozhodl frameworky ohodnotit body od 1 do 5 podle toho, do jaké míry dané kritérium splňují. Nejvíce bodů vždy získá framework, který nejlépe splňuje dané kritérium. Při konečném výběru frameworku zvažím součty těchto bodů a také další okolnosti, které plynou z požadavků a cílů tohoto projektu.

2.3.1.1 Licence

Při volbě frameworků je důležité brát v potaz licence, pod kterými byly vydány. Pokud by byla licence frameworku příliš restriktivní a nebyla by v souladu s licencí vyvíjené aplikace, nebylo by možné daný framework použít.

V tabulce 2.1 je vyobrazen přehled licencí jednotlivých frameworků.

	React	Angular	Next.js	Vue.js	Blazor
Licence	MIT	MIT	MIT	MIT	MIT

■ **Tabulka 2.1** Licence frontend frameworků

Z analýzy vyplynulo, že všechny uvažované frameworky jsou vyvíjeny pod MIT licencí, která je v souladu s licencí tohoto projektu. Proto je možné využít každý z těchto frameworků.

2.3.1.2 Možnost budoucího vývoje pro jiné platformy

Z nefunkčních požadavků tohoto projektu plyne, že musí být aplikace vyvíjena tak, aby ji bylo možné v budoucnu rozšířit na další platformy, kterými jsou například Windows, Android nebo iOS.

Proto je důležité, aby zvolené technologie tento vývoj umožňovaly. V tabulce 2.2 je vyobrazeno, zda jednotlivé frameworky umožňují vývoj na tyto platformy.

	React	Angular	Next.js	Vue.js	Blazor
Podpora vývoje pro jiné platformy	ano	ano	ano	ano	ano

■ **Tabulka 2.2** Podpora vývoje pro jiné platformy u frontend frameworků

Při používání každého analyzovaného frameworku lze vytvořit tzv. progresivní webovou aplikaci [14], kterou lze nainstalovat, spustit a používat na všech požadovaných platformách.

2.3.1.3 Míra používání

Dalším důležitým faktorem je míra používání jednotlivých frameworků. Používanější frameworky typicky bývají aktivněji vyvíjeny a udržovány, díky čemuž mohou například rychleji reagovat na nové trendy, standardy a na případné chyby v samotném frameworku. Dále mohou mít lepší podporu a aktivnější komunitu vývojářů, což může hrát při vývoji významnou roli. U frameworků s aktivnější komunitou tak může být výrazně jednodušší nalézt řešení na případné problémy s daným frameworkem.

Pro analýzu míry používání jsem se rozhodl čerpat z několika zdrojů. Jako první metriku použiji počet hvězdiček daného frameworku na platformě GitHub². Tuto metriku lze interpretovat jako velikost komunity vývojářů, kteří se o tento framework zajímají nebo ho používají.

²Dostupné na: <https://github.com/>

Jako druhou metriku budu uvažovat míru používání jednotlivých frameworků podle výsledků dotazníku webové stránky Stack Overflow [13].

Jednotlivé metriky zaokrouhlené na řády stovek jsou zobrazeny v tabulce 2.3.

	React	Angular	Next.js	Vue.js	Blazor
Počet hvězdiček na platformě GitHub	215 700	92 600	115 100	205 800	9 300
Počet respondentů využívající framework	29 100	12 500	12 000	11 800	3 500
Bodové ohodnocení	5	3	3	5	1

■ **Tabulka 2.3** Míra používání frontend frameworků

Blazor byl v roce 2019 přesunut z vlastního repozitáře [15], který měl 9 300 hvězdiček, do repozitáře ASP.NET Core, který má v době psaní textu 33 100 hvězdiček. V novém repozitáři je společně s frameworkem Blazor vyvíjeno několik dalších knihoven, a proto není možné jednoznačně určit, kolik hvězdiček přísluší které knihovně. K tomuto faktu přihlédnou při výsledné volbě frameworku.

Z analýzy vyplývá, že nejpopulárnějšími frameworky jsou React a Vue.js. Dále následují Next.js a Angular. Nejméně populárním je pak Blazor.

2.3.1.4 Počet dostupných knihoven

Při výběru frameworku je třeba brát v potaz i počet dostupných knihoven pro daný framework. Knihovny umožňují vývojářům využívat již naprogramované funkce. Větší množství knihoven tak může vést k výrazně vyšší efektivitě vývoje samotné aplikace.

Pro vyhledání počtů knihoven jsem se rozhodl využít registr npm [16], který obsahuje knihovny v jazyce JavaScript. Pro analýzu dostupných balíčků pro framework Blazor jsem využil správce .NET balíčků NuGet [17].

V tabulce 2.4 lze nalézt počty dostupných knihoven. Knihovny pro framework React jsou typicky použitelné i pro framework Next.js. Proto jsem je zahrnul do počtu knihoven Next.js.

	React	Angular	Next.js	Vue.js	Blazor
Počet dostupných knihoven	257 600	71 100	273 500	84 000	5 100
Bodové ohodnocení	4	2	5	2	1

■ **Tabulka 2.4** Počet dostupných knihoven frontend frameworků

2.3.1.5 Oblíbenost frameworků

Do analýzy jsem se rozhodl zahrnout i oblíbenost frameworků mezi vývojáři. Oblíbenost může potenciálně ovlivnit, jak se bude měnit míra používání a popularita těchto frameworků v následujících letech.

Data jsem čerpal z dotazníku [18] stránky Stack Overflow, který mimo jiné zjišťoval, do jaké míry chtějí vývojáři pracovat s danými technologiemi v budoucnu a s jakými technologiemi již pracovali. Výsledkem dotazníku bylo zjištění, kolik procent respondentů obdivuje dané technologie a kolik procent respondentů chce s danými technologiemi v budoucnu pracovat. Tato data lze nalézt v tabulce 2.5.

	React	Angular	Next.js	Vue.js	Blazor
Respondenti obdivující framework	63,6 %	50,8 %	66,0 %	57,9 %	61,0 %
Respondenti toužící po práci s frameworkem	35,3 %	13,9 %	20,3 %	17,4 %	6,7 %
Bodové ohodnocení	5	2	5	3	2

■ **Tabulka 2.5** Oblíbenost frontend frameworků

2.3.1.6 Obtížnost osvojení

Při analýze je třeba brát v potaz obtížnost učení se jednotlivých frameworků. Složitější frameworky často vyžadují větší zkušenosti vývojářů s danými technologiemi a zabírá více času, než si je vývojáři osvojí. Složitost frameworku tak může mít dopad na délku zaučování nových vývojářů, počet chyb v aplikaci a celkovou udržovatelnost systému.

Při hodnocení složitosti jsem bral v potaz kvalitu dokumentací, zejména jejich přehlednost, srozumitelnost a obsáhlost. Dále jsem bral v potaz kvalitu oficiálních materiálů a kurzů.

Do analýzy jsem také zahrnul ostatní faktory, které ovlivňují složitost frameworku. Například Vue.js může být, kvůli své syntaxi, velice jednoduchý pro vývojáře, kteří již mají zkušenosti s HTML. Blazor by zase mohl být potenciálně jednoduchý pro C# vývojáře, kteří by například pracovali na backendové části tohoto projektu.

V tabulce 2.6 lze nalézt výsledné zhodnocení těchto frameworků.

	React	Angular	Next.js	Vue.js	Blazor
Vzdělávací materiály	kvalitní	dostačující	kvalitní	kvalitní	kvalitní
Obtížnost osvojení	těžší	nejtěžší	nejtěžší	lehká	lehká
Bodové ohodnocení	4	3	3	5	5

■ **Tabulka 2.6** Složitost osvojení frameworků

2.3.1.7 Počet vývojářů

Poslední důležitou metrikou je počet dostupných vývojářů, kteří pracují s danými frameworky. Tato metrika je podstatná z dlouhodobého hlediska, zejména kvůli udržitelnosti a budoucímu vývoji projektu. Vyšší počet dostupných vývojářů na trhu typicky umožní jednodušší rozšiřování projektu.

Pro analýzu jsem se rozhodl využít vyhledávání lidí na sociální síti LinkedIn³. Do výsledků jsem zahrnul uživatele, kteří pracují nebo hledají práci v oboru informačních technologií a tvrdí o sobě, že mají zkušenosti s daným frameworkem.

	React	Angular	Next.js	Vue.js	Blazor
Počet vývojářů	2 340 000	719 000	213 000	384 000	26 000
Bodové ohodnocení	5	4	3	3	1

■ **Tabulka 2.7** Počet vývojářů ve frameworkcích

³Dostupné na: <https://www.linkedin.com/>

2.3.1.8 Závěr analýzy

Výsledky analýzy s celkovým bodovým ohodnocením frameworků lze nalézt v tabulce 2.8.

	React	Angular	Next.js	Vue.js	Blazor
Celkové bodové ohodnocení	23	14	19	18	10

■ **Tabulka 2.8** Celkové bodové ohodnocení frontend frameworků

Kromě bodového ohodnocení frameworků je třeba brát v potaz požadavky a cíle tohoto projektu. Jedním z požadavků tohoto projektu je možnost budoucího přizpůsobení webové aplikace pro SEO. Next.js, na rozdíl od ostatních frameworků, přímo poskytuje řadu nástrojů, které toto přizpůsobení umožňují. Příkladem toho jsou techniky *static site generation* a *server-side rendering*, které umožňují rychlejší načítání stránek, což může být pro SEO velice důležité [19].

Dle bodového ohodnocení se jeví React jako nejvhodnější framework pro tento projekt. Po důkladném přezkoumání jsem došel k názoru, že by framework Next.js, jakožto nadstavba frameworku React, přinesl do projektu řadu dalších užitečných funkcí.

Jelikož Next.js staví na frameworku React, a ve svém základu se mu velice podobá, sdílí s ním celou řadu výhod. Díky tomu je pro React vývojáře snadné se naučit s frameworkem Next.js pracovat. Zároveň lze jednotlivé funkcionality Next.js do projektu zakomponovat postupně a využívat je tak pouze na místech, kde jsou skutečně potřeba.

Kvůli výše zmíněným výhodám jsem se tedy rozhodl pro frontendovou část této aplikace zvolit framework Next.js.

2.3.1.9 Jazyky

Na základě zvoleného frameworku se přirozeně nabízí jazyky JavaScript a TypeScript, pomocí kterých lze projekt implementovat.

TypeScript rozšiřuje JavaScript o typy a rozhraní [20], díky kterým se v projektu jednodušeji předchází určitým chybám a vývojáři tak mohou mít větší důvěru v naprogramovaný kód. Výhodou jazyka TypeScript je také jeho možnost pracovat s kódem naprogramovaným v jazyce JavaScript. V projektu je tak například možné jednoduše používat knihovny nebo části kódu, které jsou naprogramovány v jazyce JavaScript.

Z výše uvedených důvodů jsem se rozhodl zvolit TypeScript jako hlavní jazyk, ve kterém budu tuto aplikaci programovat.

2.3.2 Knihovny a další programy

Kromě vybraného frameworku a jazyka budu využívat celou řadu dalších knihoven a technologií. Ty pomohou se samotnou implementací, organizací požadavků, odchyťáváním chyb a zajištěním kvality kódu.

2.3.2.1 Material UI

Material UI⁴ je velice populární knihovna, která poskytuje předpřipravené prvky uživatelského rozhraní. Tato knihovna je hojně využívána a má na platformě GitHub přes 91 000 hvězdiček [21]. Knihovna se řídí standardem Material Design [8], díky čemuž je možné jednodušeji a efektivněji vytvořit frontend aplikace s konzistentním uživatelským rozhraním a kvalitní uživatelskou zkušeností.

⁴Dostupné na: <https://mui.com/>

2.3.2.2 Mockoon

Aplikace Mockoon⁵ umožňuje vytvořit mock REST API. Mockoon jsem zvolil zejména kvůli jeho jednoduchosti používání a jeho pokročilejším funkcím, které umožňují například vytvářet dynamické odpovědi REST API na základě obsahu jednotlivých dotazů. Aplikace ukládá definici mocku v JSON souboru. Díky tomu je možné mock verzovat a jednoduše s ním pracovat při nasazení.

2.3.2.3 i18next

V rámci nefunkčního požadavku na rozšiřitelnost aplikace je potřeba zajistit, aby bylo možné v budoucnu kurzy poskytovat ve více jazycích. Proto jsem se rozhodl využít knihovnu i18next⁶, díky které bude možné zajistit internacionalizaci této aplikace. Frontend aplikace budu vytvářet v jazyce čeština, nicméně díky této knihovně bude možné do aplikace přidávat i ostatní jazyky.

2.3.2.4 Figma

Jak jsem již dříve zmínil, pro návrh uživatelského rozhraní jsem se rozhodl využít nástroj Figma⁷. Tento software umožňuje vytváření návrhů aplikací a obsahuje i řadu užitečných funkcí, kterými jsou například vytváření komponentů nebo správa vlastních stylů napříč celým návrhem.

2.3.2.5 Jira

Jelikož souběžně s touto prací vzniká i backend této aplikace, za který je zodpovědný kolega Jan Hlaváč, je klíčové zajistit přesnou komunikaci a specifikaci požadavků systému. Proto jsem se rozhodl využít aplikaci Jira⁸, ve které je možné spravovat jednotlivé úkoly a požadavky.

2.3.2.6 Storybook

Knihovna Storybook⁹ umožňuje vyvíjet komponenty uživatelského rozhraní v izolaci od zbytku aplikace. Následně lze ke komponentům vytvářet interaktivní dokumentaci, která umožňuje vývojáři si vyzkoušet, jak jednotlivé komponenty fungují. Díky tomu se může vývojář jednodušeji orientovat v celém systému komponentů a efektivněji s nimi pracovat. Proto jsem se rozhodl v aplikaci tuto knihovnu využít.

2.3.2.7 Sentry

Nástroj Sentry¹⁰ jsem se rozhodl použít v souvislosti s nasazením aplikace. Sentry umožňuje velice podrobně sledovat chyby, které v rámci produkčního prostředí aplikace nastaly. To bude klíčové zejména pro budoucí správu a údržbu aplikace po jejím nasazení.

2.3.2.8 SWR

V zájmu vytvoření pozitivní uživatelské zkušenosti jsem se rozhodl v aplikaci využít knihovnu SWR¹¹, která se stará o vytvoření a správu cache pomocí metody *stale-while-revalidate* [22]. Tato cache umožní rychlejší načítání uživatelského rozhraní a rychlejší reagování systému na vstupy od uživatele.

⁵Dostupné na: <https://mockoon.com/>

⁶Dostupné na: <https://www.i18next.com/>

⁷Dostupné na: <https://www.figma.com/>

⁸Dostupné na: <https://www.atlassian.com/software/jira>

⁹Dostupné na: <https://storybook.js.org/>

¹⁰Dostupné na: <https://sentry.io/>

¹¹Dostupné na: <https://swr.vercel.app/>

2.3.2.9 Nástroje pro zajištění kvality kódu

V aplikaci jsem se také rozhodl využít nástroje na zajištění kvality kódu. Mezi ně patří knihovna ESLint¹², která pomocí statické analýzy kódu zajišťuje konzistenci kódu a správné praktiky. Dále jsem se rozhodl využít knihovnu Husky¹³, která umožňuje jednoduché sestavování aplikace a spouštění analýz kódu při akcích `commit` nebo `push` v programu Git¹⁴. Díky tomu je kód často automaticky spouštěn a kontrolován. Nakonec jsem se také rozhodl využít program Prettier¹⁵, který automaticky zajišťuje konzistentní formát a styl kódu.

¹²Dostupné na: <https://eslint.org/>

¹³Dostupné na: <https://typicode.github.io/husky/>

¹⁴Dostupné na: <https://git-scm.com/>

¹⁵Dostupné na: <https://prettier.io/>

Implementace a nasazení

V této kapitole se nejprve zaměřím na realizaci samotné aplikace a popíšu, jakým způsobem jsem při implementaci frontendu postupoval. Dále také popíšu, jakým způsobem jsem komunikoval s kolegou Janem Hlaváčem, který se věnuje backendové části aplikace, abychom zajistili funkčnost celé aplikace po jejím nasazení. Následně popíšu samotný proces nasazení frontendové části aplikace, kde se budu věnovat jednak nasazení do vývojového prostředí a jednak nasazení do produkčního prostředí.

3.1 Postup při realizaci projektu

V této části textu popíšu, jakým způsobem jsem postupoval při realizaci frontendové části aplikace. Nejprve popíšu, jakým způsobem probíhala příprava celého projektu. Dále popíšu, jakým způsobem jsem postupoval při samotné implementaci frontendu. Nakonec se budu věnovat tomu, jak jsme spolupracovali s kolegou Janem Hlaváčem, aby byla zajištěna správná komunikace mezi frontendovou a backendovou částí aplikace.

3.1.1 Příprava realizace projektu

Jelikož byla aplikace vytvářena od úplného začátku, bylo potřeba před implementací provést několik úvodních kroků, které připraví projekt k implementaci a umožní efektivní vývoj. Této přípravě se budu věnovat v následujících částech textu.

3.1.1.1 Kroky před implementací

Před započítím fáze implementace jsem již měl dokončenou analýzu konkurenčních řešení a sepsané všechny případy užití. Dále jsem měl v programu Figma připravený hrubý návrh celého uživatelského rozhraní. Tento návrh obsahoval rozložení uživatelského rozhraní a základní vzhled jednotlivých stránek a komponentů. V této fázi ještě nebyl návrh dokončený. Detailní vzhled komponentů a stránek jsem se rozhodl finalizovat až v průběhu iterativního vývoje.

Díky tomuto přístupu jsem se vždy mohl v rámci dané iterace zaměřit na podrobný návrh a implementaci daných komponentů a stránek. Zároveň jsem ale měl už na začátku implementace obecný přehled o vzhledu a struktuře celého uživatelského rozhraní. To mi velice pomohlo při organizaci a strukturování kódu aplikace. Dále jsem měl díky tomuto iterativnímu přístupu větší flexibilitu při návrhu a implementaci. Mohl jsem tak velice jednoduše provádět průběžné změny v návrhu a neustále ho tak zlepšovat.

3.1.1.2 Příprava implementace

Před zahájením implementace jednotlivých funkcionalit bylo potřeba vytvořit a nastavit samotný repozitář aplikace tak, aby byl připravený pro vývoj.

Jako první krok přípravy jsem vytvořil Git repozitář na platformě GitHub a nastavil jsem projekt pomocí nástroje npm¹. Následně jsem nainstaloval a nastavil hlavní knihovny React, Next.js, Material UI a Storybook, které byly klíčové pro začátek vývoje.

Dále jsem vytvořil strukturu složek projektu a stanovil jsem určité konvence, které jsem poté při vývoji dodržoval. Při vývoji jsem se rozhodl využít knihovnu Storybook, kvůli které bylo třeba ke stránkám a komponentům vytvořit řadu dodatečných souborů. Ty mohly značně znepráhlednit celý projekt, a proto jsem se rozhodl pro každou stránku a komponent vytvořit vlastní složku se všemi relevantními soubory. Dále jsem se rozhodl pro komponenty a stránky vytvořit předpřipravené složky se všemi potřebnými soubory. Tyto složky sloužily jako šablony, pomocí kterých bylo možné velice jednoduše a rychle vytvářet nové stránky a komponenty. Zároveň byla díky těmto šablonám zachována základní konzistence struktury kódu napříč celým projektem. Kód byl tak výrazně přehlednější a jednoduše se v něm orientovalo. Podobné šablony jsem poté vytvořil i například u služeb komunikujících s API, u kterých bylo také třeba zachovat určitou formu konzistence v kódu.

Následně jsem zprovoznil jednotlivé knihovny pro zajištění kvality kódu. Mezi ně patří již dříve zmíněné knihovny ESLint a Prettier. Dále jsem pomocí knihovny Husky nastavil *git hook* [23], který v projektu při použití příkazu `git commit` automaticky spustí ESLint a při každém příkazu `git push` sestaví celý projekt. Díky tomu je zajištěna určitá úroveň kvality kódu ve vzdálených větvích repozitáře. Vývojář tak například nemůže odeslat kód, který je zjevně nefunkční. Cenou za tuto kontrolu je ovšem výrazně pomalejší provádění zmíněných příkazů programu Git. Jelikož na frontendové části této aplikace pracuji sám, nebyl jsem touto nevýhodou příliš ovlivněn. Kdyby ovšem v budoucnu na projektu pracovalo více vývojářů, stálo by za úvahu přesunout tyto kontroly na vzdálené servery. Kontroly by tak například byly prováděny pouze kdyby byl v repozitáři vytvořen *merge request* a vývojáři by tak nemuseli sestavovat projekt na svých zařízeních.

Poslední částí přípravy projektu bylo několik úvodních schůzek s kolegou Janem Hlaváčem, který se zabývá implementací backendové části této aplikace. V rámci těchto schůzek jsme diskutovali, jakým způsobem by měla aplikace fungovat, a která data bude aplikace využívat. Díky těmto schůzkám jsme získali přehled o celé aplikaci a odhadli, které části implementace budou nejpracnější a ve kterých částech projektu mohou potenciálně nastat problémy. Na základě těchto schůzek jsme pak byli schopni přesněji projekt naplánovat a dohodnout se na postupu implementace.

3.1.2 Postup při implementaci frontendu

Po dokončení přípravy projektu jsem mohl začít se samotnou implementací jednotlivých funkcionalit. Při vývoji frontendu jsem se rozhodl postupovat iterativní metodou. V rámci každé iterace jsem vybral několik souvisejících případů užití, na kterých jsem následně pracoval. Díky tomu, že byly jednotlivé případy užití rozděleny podle priorit, bylo velice jednoduché postupovat od nejdůležitějších funkcionalit k méně důležitým. Základní funkčnost aplikace tak byla k dispozici již v rané fázi vývoje.

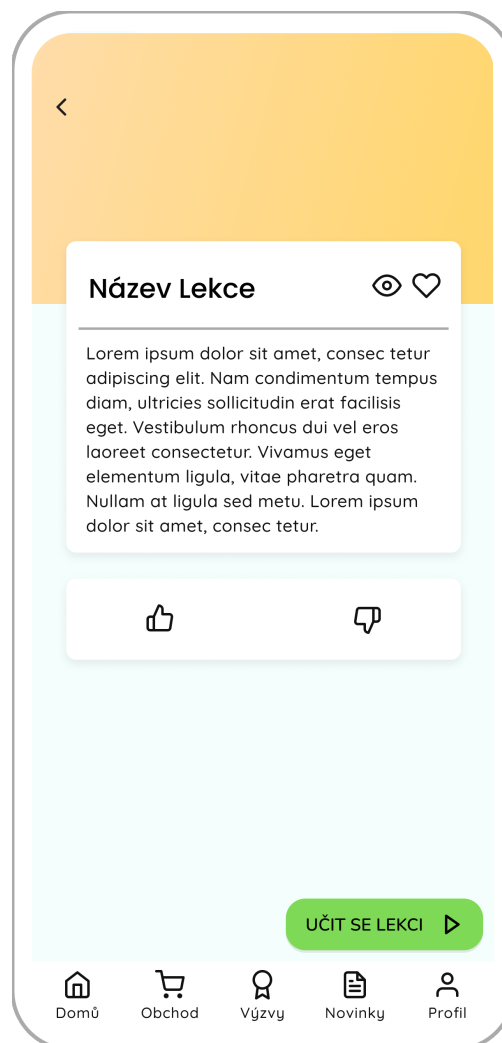
V následujících částech této kapitoly popíšu jednotlivé kroky, pomocí kterých jsem na základě případů užití implementoval jednotlivé stránky a funkcionality aplikace. Pro ilustraci ukážu jednotlivé kroky vývoje na konkrétním případě užití UC9 – Zobrazení lekce, v rámci kterého jsem se zabýval vývojem stránky, která vyobrazuje přehled dané lekce v kurzu angličtiny.

¹Dostupné na: <https://www.npmjs.com/>

3.1.2.1 Dokončení návrhu

Před začátkem vývoje jsem měl již připravený návrh celého uživatelského rozhraní s rozložením jednotlivých stránek a komponentů. V této fázi vývoje jsem se zaměřil na dokončení návrhu dané stránky v programu Figma. Protože jsem se při vývoji snažil zaměřit na pozitivní uživatelskou zkušenost, rozhodl jsem se vždy danou stránku aplikace detailně navrhnut tak, aby vzhledem co nejpřesněji odpovídala výsledné stránce v aplikaci. Díky tomu jsem mohl případné vizuální nedostatky v uživatelském rozhraní odhalit ještě před samotnou implementací. Ukázkou dokončeného návrhu přehledu lekce v aplikaci lze nalézt na obrázku 3.1.

Díky detailnímu návrhu a dříve popsaným případům užití jsem tak při implementaci mohl vycházet z přesné specifikace, jakým způsobem má aplikace vypadat a fungovat. To samotnou implementaci značně zjednodušilo.



■ Obrázek 3.1 Návrh přehledu lekce mluvení

3.1.2.2 Implementace komponentů

Po dokončení návrhu dané stránky jsem nejdříve začal s implementací jednotlivých komponentů uživatelského rozhraní, které se na stránce nachází.

Komponenty jsem vyvíjel pomocí knihovny Storybook. Tato knihovna umožňuje komponenty vyvíjet samostatně a nezávisle na již existujícím uživatelském rozhraní. Knihovna poskytuje funkce, které umožňují předávat komponentám různé sady dat, na základě kterých se komponenty vykreslují. Storybook následně, na základě těchto dat, sestaví jednoduchou interaktivní dokumentaci, v níž je možné komponenty zobrazovat v různých stavech a prostředích, například na různých velikostech obrazovek. S těmito komponenty lze pak v rámci dokumentace interagovat a manuálně je testovat.

Tento postup, ve kterém jsem nejdříve izolovaně vyvíjel samostatné komponenty, pak je manuálně testoval, a až poté z nich vytvářel uživatelské rozhraní, se ukázal být velice užitečným. Otestované komponenty pak bylo velice jednoduché začlenit do existujícího uživatelského rozhraní. Díky knihovně Storybook souběžně s komponenty vznikala i jejich jednoduchá interaktivní dokumentace, díky které bylo snadné komponenty vícekrát využívat a vracet se k nim při vývoji dalších stránek.

3.1.2.3 Vytvoření mock API

Po implementaci a manuálním testování potřebných komponentů jsem dále pokračoval vytvořením mocku koncových bodů REST API, které bylo třeba na dané stránce využívat.

Mock REST API jsem vytvářel pomocí programu Mockoon, který umožňuje snadné vytváření mocku REST API pomocí grafického uživatelského rozhraní a jeho následné spuštění v podobě serveru. Definice mocku je ukládána do jednoho JSON souboru, který lze následně verzovat například pomocí programu Git. Na základě tohoto souboru je pak možné mock nasadit s využitím konzolové verze aplikace Mockoon CLI² nebo s pomocí balíčku Mockoon Serverless³. Samotnému nasazení se budu věnovat v dalších částech této kapitoly.

V rámci vytváření mocku jsem využil některé pokročilejší funkce aplikace Mockoon. Jednou z těchto funkcí je tzv. *templating*, který umožňuje vytváření dynamických odpovědí serveru [24]. Tuto funkci jsem využil například v kombinaci s knihovnou Faker.js⁴, díky které lze generovat data pro zmíněné odpovědi serveru. Příklad definice takové odpovědi lze nalézt v ukázce kódu 3.1.

```
{
  "id": "{{faker 'string.uuid'}}",
  "name": "{{faker 'lorem.words' 4}}",
  "favorite": false,
  "visible": true,
  "description": "{{faker 'lorem.paragraph' 10}}",
  "type": "SPEAKING"
}
```

■ **Výpis kódu 3.1** Definice odpovědi koncového bodu pro získání dané lekce

3.1.2.4 Vytvoření služby pro komunikaci s API

Po vytvoření mocku koncových bodů API jsem vždy v kódu frontendu vytvořil službu, která měla za úkol zprostředkovávat komunikaci s daným koncovým bodem API. Příklad takové služby lze nalézt v ukázce kódu 3.2.

Při implementaci těchto služeb jsem pro dotazy typu GET vytvořil funkce, které jsou v kontextu vývoje ve frameworku React nazývány *custom hooks* [25]. Příkladem takové funkce je `useLesson`

²Dostupné na: <https://mockoon.com/cli/>

³Dostupné na: <https://mockoon.com/serverless/>

⁴Dostupné na: <https://fakerjs.dev/>

```

const LessonsAPI = {
  URI: (courseId: Id) => `user/courses/${courseId}/lessons`,

  useLesson(
    courseId: Id,
    lessonId: Id
  ): Modify<FetchHook<Lesson>, { lesson: Lesson }> {
    const { data, ...rest } = useAPI<Lesson>(
      `${this.URI(courseId)}/${lessonId}`
    );
    return { lesson: data, ...rest };
  },

  /* ... */

  async updateLesson(
    courseId: Id,
    lesson: LessonUpdateDTO
  ): Promise<Lesson> {
    return API.patch(`${this.URI(courseId)}/${lesson.id}`, lesson);
  },

  /* ... */
};

```

■ Výpis kódu 3.2 Ukázka služby LessonsAPI

v ukázce 3.2. Díky těmto funkcím je možné snadno přistupovat k datům API ze samotných React komponentů.

V těchto funkcích jsem dále využíval *custom hook useAPI*, který se stará o zachycování případných chyb a také využívá knihovnu SWR pro cachování odpovědí ze serveru.

Pro všechny typy dotazů jsem pak využíval vlastní službu pojmenovanou API. Tato služba se stará o zajištění samotné komunikace mezi frontendem a backendovým API pomocí knihovny Axios⁵.

3.1.2.5 Implementace samotné stránky

Po dokončení všech předchozích kroků již zbývalo sestavit samotnou stránku aplikace. K tomu jsem využil připravené komponenty a výše popsanou službu pro komunikaci s API. V ukázce kódu 3.3 je možné nalézt ukázkou implementace stránky zobrazující přehled lekce kurzu v aplikaci.

Ve zmíněné ukázce je vidět využitý dříve popsany *custom hook useLesson*, který jednak poskytuje data získaná z API a jednak poskytuje funkci *mutate* z knihovny SWR, která umožňuje data měnit v místní *cache*. Použití této funkce je vidět ve funkci *handleFavoriteChange*, která je spuštěna, když uživatel označí danou lekci za svoji oblíbenou.

Tato funkce odešle pomocí *LessonsAPI* požadavek o změnu dat. Zároveň je využita funkce *mutate*, která okamžitě změní data v místní *cache*. Tato technika je nazývána *optimistic updates* [26]. Díky ní uživatel v aplikaci okamžitě vidí danou změnu a nemusí čekat na odpověď ze serveru. Vyřízení odeslaného požadavku pak probíhá v pozadí. Pokud by odeslaný požadavek z jakéhokoliv důvodu selhal, je uživatel upozorněn.

⁵Dostupné na: <https://axios-http.com/>

Dále je v ukázce kódu využitý *custom hook* `useTranslation`, který poskytuje funkci `t`. Ta pomocí knihovny `i18next` zobrazuje text v uživatelském rozhraní na základě zvoleného jazyka. Díky tomu je zajištěna základní forma internacionalizace.

```
const LessonOverview: React.FC<ILessonOverview> = ({ lessonId, courseId }) => {
  const { t } = useTranslation("common");
  const { lesson, mutate } = LessonsAPI.useLesson(courseId, lessonId);

  /* ... */

  function handleFavoriteChange(value: boolean) {
    mutate(
      LessonsAPI.updateLesson(courseId, {
        id: lessonId,
        favorite: value
      }),
      optimisticMutationOption({
        ...lesson,
        favorite: value,
      })
    );
  }

  return (
    <ContentContainer>
      /* ... */
      <BottomFab
        header={t("studying.studyLesson")}
        icon={icons.startStudy}
        onClick={() => router.push(`/study?lessonId=${lessonId}`)}
      />
    </ContentContainer>
  );
};
```

■ **Výpis kódu 3.3** Ukázka implementace stránky s přehledem lekce

3.1.2.6 Testování implementované stránky

Kromě dříve provedeného testování jednotlivých komponentů jsem vždy po dokončení dané stránky provedl i její manuální otestování. Toto testování již probíhalo v samotné aplikaci s využitím mocku API.

V rámci tohoto testování jsem se soustředil hlavně na samotnou funkčnost aplikace. Dále jsem se také zaměřil na ověření, zda daná stránka aplikace splňuje 10 heuristik uživatelské použitelnosti [27], které popsal Jakob Nielsen, Ph.D. Jednotlivé heuristiky popisují základní principy, pomocí kterých by mělo být uživatelské rozhraní vytvořeno, aby poskytovalo kvalitní uživatelskou zkušenost. Ověření, zda aplikace tato doporučení splňuje, mi umožnilo najít a opravit případné nedostatky v uživatelském rozhraní.

Na tomto místě je třeba zdůraznit, že ověření, zda aplikace splňuje zmíněné heuristiky jsem prováděl pouze já za účelem nalezení základních nedostatků. Nejednalo se tedy o komplexní heuristickou analýzu, ve které by se například angažovali experti se znalostmi dané domény.

Složitější heuristická analýza by byla mimo rozsah této bakalářské práce, a proto jsem se jí rozhodl neprovádět.

3.1.3 Postup při spolupráci s backendem

Jelikož je cílem projektu vytvořit a nasadit funkční aplikaci, bylo potřeba v průběhu vývoje efektivně komunikovat s kolegou Janem Hlaváčem, který souběžně s touto prací vytvářel backendovou část této aplikace. V této části kapitoly se zaměřím na popis, jakým způsobem jsme v rámci implementace postupovali, abychom zajistili konzistenci mezi mockem API, který je součástí této práce a reálným API, které je součástí práce zmíněného kolegy.

V této části kapitoly budu mne a kolegu Jana Hlaváče pro jednoduchost označovat jako tým, přestože jsme každý na svých bakalářských pracích pracovali individuálně a komunikovali jsme spolu převážně kvůli zajištění konzistence mezi specifikacemi mock API a reálného API a kvůli následnému nasazení aplikace.

3.1.3.1 Zvolená metodika

Při vývoji softwaru v týmu lze postupovat různými způsoby. V rámci tohoto projektu jsme se rozhodli využít agilní metodiku Kanban [28]. Tuto metodiku jsme upřednostnili před jinými metodikami kvůli několika důvodům.

Hlavním důvodem byla jednoduchost metodiky Kanban. V týmu o dvou vývojářích by například striktní dodržování metodiky Scrum [29] mohlo vést ke zbytečnému komplikování pracovního postupu.

Dalším důvodem byla nekonzistentní a nepředvídatelná dostupnost obou vývojářů. Jelikož se tým nenacházel v korporátním prostředí s konzistentní pracovní dobou, nýbrž v prostředí akademickém, ve kterém bylo třeba v různé časy pracovat na různých projektech, nebo se věnovat jiným akademickým povinnostem, bylo velice obtížné předvídat dostupnost členů týmu. K tomu přispěl i fakt, že se oba členové týmu rozhodli v průběhu vývoje vycestovat do zahraničí a dočasně se tak věnovat studiím na jiných univerzitách. Použití jednodušší a flexibilnější metodiky se tak jevilo jako nejvhodnější varianta pro tento projekt.

Kanban nám tak umožnil pracovat na sobě téměř nezávisle. Tato metodika se tak ukázala být velice vhodná pro tento konkrétní projekt.

3.1.3.2 Postup při implementaci

Implementace zpravidla začínala na frontendové části aplikace. Dříve popsaným způsobem jsem navrhl uživatelské rozhraní, vytvořil mock koncových bodů API a implementoval danou stránku v aplikaci. Mock API byl v této fázi pouze mým prvotním návrhem a zdaleka neodpovídal finální verzi API. Díky tomuto mocku jsem byl ovšem schopen přesně určit a definovat, která data jsou ve frontendové části aplikace potřeba.

Následně jsme s kolegou Hlaváčem vytvořili úkoly v programu Jira, které obsahovaly požadavky na vytvoření koncových bodů REST API. Kolega pak na základě těchto úkolů vytvořil přesnou specifikaci daného koncového bodu API a implementoval ho společně s backendovou částí aplikace.

Na základě finální specifikace od kolegy Hlaváče jsem následně provedl případné změny v mocku API a frontendové části aplikace.

Díky tomuto postupu jsme byli v týmu schopni pracovat takřka nezávisle na sobě a zajistit konzistenci mezi mock API a reálným API, která byla klíčová při následném nasazení aplikace.

3.2 Nasazení aplikace

V této části kapitoly se zaměřím na popis nasazení frontendové části aplikace. V rámci nasazení jsem pracoval se dvěma prostředími. Aplikaci jsem nasadil jednak ve vývojovém prostředí pro účely interního testování. V tomto prostředí frontend aplikace využívá mock API, který jsem společně s aplikací vytvářel v průběhu implementace. Dále jsem, v souladu s hlavním cílem tohoto projektu, nasadil frontend aplikace do produkčního prostředí. V produkčním prostředí bylo již třeba zajistit komunikaci s reálným REST API, které je součástí backendové části tohoto projektu.

3.2.1 Vývojové prostředí frontendu

Vývojové prostředí aplikace jsem zprovoznil již v rané fázi vývoje. Toto prostředí jsem využíval zejména pro interní testování, diskuze na schůzkách a dále pro uživatelské testování. Toto prostředí také umožnilo otestovat základní funkčnost frontendu aplikace na vybraných serverech před nasazením aplikace do produkčního prostředí. Díky tomu jsem předešel řadě chyb, které se projeví až po nasazení do testovacího prostředí.

Pro nasazení aplikace jsem nejprve zvolil hostingovou službu Netlify. Ta se v rámci testování aplikace ukázala být nevhodná pro nejnovější verze frameworku Next.js. Některé funkce tohoto frameworku nebyly plně podporovány a řada stránek v aplikaci tak nefungovala dle očekávání.

Proto jsem se rozhodl přesunout aplikaci k hostingové službě společnosti Vercel. Společnost Vercel je zodpovědná za vývoj frameworku Next.js, což přináší při provozu aplikace řadu výhod. Nejnovější funkce frameworku jsou v rámci hostingové služby podporovány a prostředí je přizpůsobeno pro optimální funkčnost Next.js aplikací. Díky tomu bylo samotné nasazení velice jednoduché a všechny problémy s předchozím hostingem byly tímto přesunem vyřešeny.

V průběhu vývoje jsem pro verzování kódu využíval již dříve zmiňovanou technologii Git a službu GitHub. Tyto technologie proces nasazení značně zjednodušily. V Git repozitáři jsem využíval dvě hlavní větve pojmenované `dev` a `master`. První zmíněná větev odpovídá verzi aplikace, která je nasazena ve vývojovém prostředí. Hostingovou službu Vercel jsem propojil s GitHub repozitářem tak, že se při aktualizaci větve `dev` aplikace znovu automaticky sestaví a nasadí do testovacího prostředí.

3.2.1.1 Mock API

Společně s aplikací v testovacím prostředí bylo také třeba nasadit samotný mock API, který tato aplikace využívá.

Jelikož jsem pro verzování mocku také využíval Git a GitHub, byl proces podobný jako při nasazování aplikace. Mock jsem nasadil s využitím služby Netlify a dříve zmíněného balíčku Mockoon Serverless. Dříve zmíněné problémy se službou Netlify se u mocku nevyskytovaly, a proto jsem se rozhodl ho na serverech Netlify ponechat.

3.2.2 Produkční prostředí frontendu

Nasazení do produkčního prostředí proběhlo až v posledních fázích vývoje po dokončení hlavních funkcionalit aplikace. Pro nasazení frontendu do produkčního prostředí jsem již rovnou použil hostingovou službu společnosti Vercel, která se osvědčila při nasazování aplikace do testovacího prostředí.

Jak jsem již zmínil, v repozitáři jsem využíval dvě hlavní větve `dev` a `master`. Druhá větev odpovídá verzi aplikace, která je nasazena v produkčním prostředí. Pro vytvoření vydání aplikace jsem vytvořil tzv. *release branch*, což je standardní postup při pracovním postupu Gitflow [30]. V této větvi jsem následně provedl úpravy kódu tak, aby byla aplikace kompatibilní s aktuální verzí REST API backendové části aplikace. Po dokončení úprav jsem dle postupu Gitflow sloučil

větev *release branch* do větví **dev** a **master**. Z větve **master** pak byl frontend aplikace automaticky sestaven a nasazen do produkčního prostředí.

Pro monitorování stavu frontendové části aplikace v produkčním prostředí jsem použil již dříve zmíněný program Sentry. Tento program umožňuje zejména sledovat chyby, které při používání aplikace nastávají, a značně tak ulehčuje provoz a údržbu aplikace.

Kapitola 4

Testování

Testování je nedílnou součástí vývoje softwaru a je klíčové pro realizaci funkční a kvalitní aplikace. V této kapitole se nejprve budu věnovat výběru vhodné formy testování na základě požadavků tohoto projektu. Způsobů testování je celá řada, a proto je třeba s ohledem na rozsah a priority tohoto projektu zvolit ty nejvhodnější formy testování. V dalších částech této kapitoly pak popíšu, jakým způsobem jsem toto testování realizoval. Nakonec zhodnotím výstupy celé bakalářské práce a navrhuji možné úpravy do budoucna.

4.1 Volba vhodných forem testování

Při volbě vhodných forem testování front-endové části aplikace jsem vybíral z hojně používaných způsobů testování front-endu, kterými jsou například unit testy, integrační testy, smoke testy, regresní testy, ale i různé formy uživatelského testování. Při výběru jsem se také rozhodoval mezi manuálním testováním a vytvářením automatizovaných testů.

Jak jsem již zmínil, pro správnou volbu testů bylo třeba brát v úvahu jednak rozsah a jednak priority projektu. Z analýzy existujících řešení a nefunkčních požadavků této aplikace vyplynulo, že kvalitní uživatelská zkušenost je klíčová pro tento projekt. Z tohoto důvodu jsem se rozhodl realizovat uživatelské testování ve formě testování použitelnosti aplikace. Toto testování umožní odhalit problémy v uživatelském rozhraní a také příležitosti pro zlepšení aplikace [31].

Dále jsem se, s ohledem na rozsah tohoto projektu, rozhodl v průběhu vývoje provádět průběžné manuální testování aplikace. Tomu jsem dal přednost před automatizovanými testy zejména kvůli jejich časové náročnosti. Díky využití knihovny Storybook, je ovšem aplikace do budoucna připravena pro vytvoření automatizovaných testů, což umožňují různá rozšíření této knihovny, kterým je například knihovna Storybook Test Runner [32].

4.2 Manuální testování

V průběhu celého vývoje jsem průběžně prováděl manuální testování aplikace. Testování bylo zaměřeno zejména na správnou funkčnost jednotlivých komponentů a stránek v aplikaci.

4.2.0.0.1 Testování komponentů Prvním testováním, které jsem prováděl, bylo ověřování správné funkčnosti jednotlivých komponentů v aplikaci. Komponenty jsem testoval manuálně v prostředí knihovny Storybook, ve kterém jsem tyto komponenty také vyvíjel. Díky tomuto testování jsem tak mohl uživatelské rozhraní sestavovat z funkčních komponentů a potenciálně tak předejít složitějším chybám v uživatelském rozhraní.

4.2.0.0.2 Testování stránek Druhým testováním, které jsem při vývoji prováděl, bylo testování dokončených stránek aplikace. Tyto stránky se skládaly z již otestovaných komponentů a využívaly vytvořený mock REST API. Jak jsem již dříve zmínil, v rámci tohoto testování jsem také zkontroloval, zda daná stránka aplikace dodržuje 10 heuristik uživatelské použitelnosti [27]. Díky tomuto testování jsem ověřil, že daná funkcionalita nebo stránka v aplikaci funguje dle očekávání.

4.2.0.0.3 Testování s reálným API Posledním testováním, které jsem prováděl, bylo testování jednotlivých funkcí a stránek v nasazené aplikaci. Díky tomuto testování jsem ověřil, že frontendová část aplikace skutečně funguje dle očekávání s reálným REST API.

4.3 Uživatelské testování

V rámci uživatelského testování jsem se rozhodl uskutečnit testování použitelnosti dle doporučení společnosti Nielsen Norman Group [31]. Testování se skládá z fáze plánování, ve které se připraví jednotlivé testovací scénáře a vybere se skupina testerů. Tyto testery budu v této kapitole také označovat jako účastníky testování. Po dokončení plánování se s účastníky testování provedou jednotlivé testy. Nakonec se v poslední fázi zhodnotí výsledky celého testování.

4.3.1 Plán testování

Samotné testování lze provádět jednak prezenčně a jednak online. V této práci jsem se rozhodl provést prezenční testování, zejména protože mi umožnilo přesněji sledovat reakce účastníků. Osobnější přístup mi také umožnil po dokončení testování od účastníků získat další zpětnou vazbu, kterou bych pravděpodobně nebyl schopen získat, kdybych testování prováděl distančně.

Nielsen Norman Group dále rozlišuje moderované a nemoderované testování [31]. V rámci moderovaného testování testeři interagují s tzv. moderátorem, tedy s výzkumníkem, který má testování na starosti a dává testerům úkoly, které mají splnit. Nemoderované testování naopak probíhá bez komunikace s moderátorem a testeři provádějí testy sami ve svém volném čase. Nemoderované testování typicky testeři nahrávají a následně organizátorům testů tuto nahrávku odešlou společně s výsledky testů. V této práci jsem se rozhodl realizovat moderované testování, protože od testerů nevyžaduje takovou technickou zdatnost a je jednodušší na provedení.

Před provedením testů je také třeba rozhodnout, kolik testerů bude pro realizaci testování potřeba. Podle doporučení Nielsen Norman Group [33] jsem se rozhodl pro testování využít 5 účastníků testování. Hlavním důvodem tohoto doporučení je, že s přibývajícím počtem testerů se jejich zpětná vazba čím dál tím více překrývá. S každým dalším testerem proto klesá počet nových poznatků, které testování přináší.

Pro testování je také třeba pečlivě vybrat jednotlivé testery. Účastníci testování by se měli co nejvíce blížit reálným uživatelům [31]. Proto jsem jako účastníky zvolil jednotlivce, kteří mají zájem zlepšit svoje dovednosti v anglickém jazyce. Dále jsem se snažil do testování zahrnout účastníky s různou úrovní angličtiny a různou zkušeností s ostatními aplikacemi na výuku jazyků. Přehled účastníků testování lze najít v tabulce 4.1.

V rámci přípravy testování je dále třeba rozhodnout, které funkcionality budou testovány. Do testování jsem se rozhodl zahrnout nejdůležitější funkcionality aplikace, kterými jsou například registrace, zapsání kurzu, učení se, vyhledávání lekcí, vypínání témat v kurzu a vytváření vlastních lekcí.

4.3.2 Testovací scénáře

V rámci testování jsem vytvořil následující scénáře. Každý scénář se zaměřuje na určitou funkcionalitu aplikace a skládá se ze zadání a očekávaného průchodu aplikací. Očekávaný průchod

Jméno	Věk	Popis	Úroveň v angličtině
Marie	23 let	Aktivně nepoužívá aplikace na výuku jazyků.	B2 až C1
Jiří	23 let	Vášnivý uživatel několika aplikací na výuku jazyků.	C1
Jakub	23 let	Pravidelný uživatel aplikace Duolingo.	B2
Anna	20 let	Pravidelná uživatelka aplikace Duolingo.	C1
Tomáš	56 let	Nikdy nepoužíval aplikace na výuku jazyků.	A2 až B1

■ **Tabulka 4.1** Vybraní testeři pro uživatelské testování

aplikací sloužil pouze jako specifikace daného scénáře a účastníkům jsem ho při testování neukazoval.

4.3.2.1 TS1 – Registrace a zapsání kurzu

4.3.2.1.1 Zadání Dostal jste se na stránku aplikace na učení se angličtiny. Vaším cílem je si v aplikaci založit účet a zapsat si kurz angličtiny. Při zakládání účtu můžete použít libovolné smyšlené údaje. Nemusíte si je pamatovat.

4.3.2.1.2 Očekávaný průchod aplikací

1. Uživatel začíná na stránce přihlášení.
2. Ze stránky přihlášení se přesune na stránku registrace, kde do formuláře vyplní uživatelské jméno, emailovou adresu a heslo. Dále zaškrtně políčko, kterým vyjádří souhlas s podmínkami. Odesláním formuláře se zaregistruje.
3. Dále si uživatel zvolí kurz, který si chce zapsat.
4. Následně si uživatel vybere témata, na která se chce v kurzu zaměřit.
5. Dále si uživatel zvolí svůj denní cíl.
6. Následně si uživatel zvolí, jakým způsobem chce v aplikaci začít. K dispozici je možnost začít od první lekce a možnost zvolit si svoji úroveň v daném jazyce.
7. Po zvolení začátku je uživatel přesměrován na hlavní stránku kurzu.

4.3.2.2 TS2 – Učení a procvičování

4.3.2.2.1 Zadání Otevřel jste aplikaci a chcete se začít učit anglicky. Zapněte učení a splňte v něm jednotlivá cvičení.

1. Uživatel začíná na hlavní stránce kurzu.
2. Na stránce uživatel stiskne tlačítko s nápisem *Učit se*.
3. Následně uživatel splní pět vybraných cvičení.
4. Po dokončení všech cvičení se uživateli zobrazí statistiky o učení. Uživatel tyto statistiky přeskóčí a dostane se opět na hlavní stránku kurzu.

4.3.2.3 TS3 – Změna úrovně učení

4.3.2.3.1 Zadání Při zapsání kurzu angličtiny jste zvolil, že máte úroveň angličtiny *A1 Začátečnick*. Při učení jste zjistil, že je vaše angličtina spíše na úrovni *B1 Středně pokročilý*. Aplikace Vám tak při spuštění učení dává příliš lehká cvičení z látky, kterou již znáte. Zaříd'te, aby Vám aplikace při učení ukazovala cvičení z lekcí na úrovni *B1 Středně pokročilý*.

1. Uživatel začíná na hlavní stránce kurzu.
2. Uživatel se pomocí hlavní nabídky přesune na mapu učení na úrovni B1.
3. Dále uživatel vybere libovolnou lekci a klikne na tlačítko k přesunutí se na dané políčko.

4.3.2.4 TS4 – Učení se konkrétní lekcí

4.3.2.4.1 Zadání Chcete si procvičit slovíčka zaměřená na ovoce (*fruits*) v angličtině. Udělejte několik cvičení zaměřených na tuto slovní zásobu.

1. Uživatel začíná na hlavní stránce kurzu.
2. Uživatel otevře vyhledávání v hlavní nabídce aplikace a vyhledá klíčové slovo *fruits*. Lekci může případně také nalézt pomocí seznamu lekcí v kurzu.
3. Na stránce dané lekce uživatel spustí učení kliknutím na tlačítko *Učit se lekci*.
4. Následně uživatel splní tři vybraná cvičení.
5. Po dokončení všech cvičení se uživateli zobrazí statistiky o učení. Uživatel tyto statistiky přeskóčí a dostane se opět na hlavní stránku kurzu.

4.3.2.5 TS5 – Vypnutí témat v kurzu

4.3.2.5.1 Zadání V rámci kurzu narazíte na různé lekce zaměřené na cestování. Vás cestování nezajímá a nechcete se ho v kurzu učit. Zaříd'te, aby Vám aplikace lekce zaměřené na cestování nezobrazovala.

1. Uživatel začíná na hlavní stránce kurzu.
2. Uživatel pomocí hlavní nabídky otevře stránku nastavení. Na této stránce uživatel klikne na tlačítko *Přizpůsobit témata kurzu*.
3. Uživatel vypne téma cestování.

4.3.2.6 TS6 – Vytvoření vlastní lekce

4.3.2.6.1 Zadání Budete psát ve škole test a musíte se naučit specifická slovíčka oceán (*ocean*), medúza (*jellyfish*) a bambus (*bamboo*). Vytvořte vlastní lekci s těmito slovíčky.

1. Uživatel začíná na hlavní stránce kurzu.
2. Uživatel otevře stránku *Moje slovíčka* v hlavní nabídce aplikace.
3. Dále uživatel klikne na ikonu pro vytvoření nové lekce.
4. Následně uživatel vyplní název a popis lekce. Dále vyhledá jednotlivá slovíčka a přidá je do lekce kliknutím na tlačítko ve tvaru znaménka plus.
5. Poté uživatel stiskne tlačítko *Vytvořit*.

4.3.3 Průběh testování

Testování jsem prováděl s pomocí nasazené aplikace v testovacím prostředí, která využívala mock API. Mock jsem pro testování upravil tak, aby obsah aplikace co nejvíce odpovídal obsahu reálné aplikace. Díky tomu, že byla aplikace nasazena v testovacím prostředí a byla tak dostupná na webových stránkách, nemusel jsem při přípravě testování nic instalovat a testování tak probíhalo velice hladce.

Jelikož jsem aplikaci vyvíjel přístupem *mobile-first*, rozhodl jsem se testování provádět na mobilních zařízeních. Pro zachování konzistence a zjednodušení přípravy jsem při testování každému účastníkovi poskytl mobilní zařízení, na kterém mohl testování provádět.

V průběhu testování jsem nahrával záznam obrazovky a audia¹. K testům jsem se tak mohl kdykoliv zpětně vrátit a v průběhu testování jsem se nemusel tolik soustředit na vytváření detailních poznámek. Díky tomu jsem se mohl více soustředit na pozorování samotných účastníků a jejich reakcí při používání aplikace.

Samotné testování probíhalo následujícím způsobem. Nejdříve jsem danému účastníkovi vysvětlil, jakým způsobem bude testování probíhat a zeptal jsem se ho na několik úvodních otázek, abych zjistil, jaké zkušenosti má s učením se jazyků a používáním podobných aplikací. Výsledky tohoto dotazníku jsou dostupné v příloze A. Následovalo samotné testování, ve kterém jsem účastníkovi postupně zadal šest úkolů, které se snažil v aplikaci splnit. Po dokončení každého úkolu měl účastník možnost vyjádřit svůj názor ohledně dané funkcionality. Většina účastníků se k funkcím vyjadřovala v průběhu plnění úkolu, a proto většinou na konci úkolu neměli další připomínky. Po dokončení všech úkolů jsem účastníkům předložil k vyplnění dotazník *System Usability Scale*, jehož cílem bylo zhodnotit celkovou použitelnost systému. Výsledky tohoto dotazníku lze nalézt v příloze B. Při vyplňování dotazníku jsem pobídl účastníky, aby komentovali, proč v dotazníku zvolili dané odpovědi.

Při testování jsem upozoroval, že měla většina účastníků k aplikaci ještě řadu cenných komentářů a nápadů, které mi sdělili až po ukončení nahrávaného testování. Proto jsem se rozhodl po konci každého testování chvíli vést neformální konverzaci s daným účastníkem, díky čemuž jsem od každého účastníka získal řadu doplňujících připomínek a nápadů.

4.3.4 Výsledky testování

Testování použitelnosti se ukázalo být velice přínosným. V průběhu testování byla odhalena řada drobných nedostatků a příležitostí pro zlepšení uživatelského rozhraní. Přehled těchto nedostatků a příležitostí pro zlepšení je k dispozici v příloze C. Mezi tyto nedostatky patří například málo výrazné znázornění, zda uživatel při děláni cvičení odpověděl správně nebo špatně. Aplikace v některých cvičeních například jen podtrhávala špatné odpovědi červenou barvou a uživateli tak trvalo příliš dlouho, než zjistil, jestli ve cvičení odpověděl správně nebo ne. Proto by například aplikace měla uživateli dávat zřetelnější a výraznější zpětnou vazbu, zda při cvičení odpověděl správně nebo špatně. Procházení cvičení tak bude pro uživatele výrazně rychlejší a plynulejší.

Kromě drobnějších nedostatků byl odhalen jeden hlavní nedostatek, kterým je matoucí navigace na mapě učení. Mapu učení lze nalézt na obrázku 2.1. Tato mapa slouží jako přehled lekcí v kurzu a je rozdělena do šesti sekcí podle úrovní angličtiny. Uživatel se v rámci mapy učení nachází na jedné konkrétní lekci. Pozice uživatele na mapě určuje, která cvičení mu bude aplikace při učení a opakování zobrazovat. Svoji pozici na mapě může uživatel libovolně měnit. Jelikož je mapa rozdělena do zmíněných sekcí podle úrovní angličtiny, bylo pro uživatele matoucí přecházení mezi těmito sekcemi. Uživatelé si pak například po přechodu do jiné sekce nebyli jistí, kde se na mapě nachází. Zjednodušení tohoto přehledu a navigace by tak mohlo vést k lepší uživatelské zkušenosti.

Po dokončení testování jsem opravil většinu nedostatků, které byly při testování objeveny. Některé nedostatky, jako například zmíněná navigace na mapě učení, budou vyžadovat kom-

¹Dostupné na: <https://drive.google.com/drive/folders/1gFmYaNS1h7j47vmjnx.YL7003nF.FBP0>

plexnější návrh a změny v uživatelském rozhraní a další iterace testování použitelnosti. Těmto změnám jsem se proto rozhodl v této práci dále nevěnovat a ponechat je tak na budoucím vývoji aplikace.

Po testování účastníci aplikaci hodnotili velice kladně. Většina účastníků tvrdila, že po opravení nedostatků, na které se přišlo v průběhu testování, by si uměli představit, že by aplikaci pravidelně používali.

Celkově na základě výsledků dotazníku *System Usability Scale* aplikace získala skóre 82,5. Toto skóre lze interpretovat tak, že má aplikace *dobrou použitelnost* [34]. Systém tak lze považovat za přijatelný pro běžné používání.

Přesto má aplikace stále velký prostor pro zlepšení. Testování použitelnosti by se tak měla, dle mého názoru, dále věnovat pozornost při budoucím vývoji. Provádění pravidelných iterací uživatelského testování by tak umožnilo zdokonalení jednotlivých funkcí a zajištění kvalitní uživatelské zkušenosti.

4.4 Zhodnocení aplikace a návrh úprav do budoucna

V této části kapitoly nejprve zhodnotím realizovanou aplikaci a následně navrhu možné úpravy aplikace do budoucna. Při zhodnocení aplikace budu brát v potaz celý proces vývoje včetně provedeného testování.

4.4.1 Zhodnocení aplikace

V rámci návrhu jsem definoval 41 případů užití, které vycházely z požadavků pro tuto aplikaci a analýzy existujících řešení. Jednotlivé případy užití měly různé priority. V rámci implementace se mi podařilo každý z těchto případů realizovat a manuálně otestovat.

V rámci analýzy byly také vytyčeny 4 nefunkční požadavky týkající se frontendové části aplikace. Požadavek na rozšiřitelnost systému byl splněn, díky vhodnému návrhu a rozvržení aplikace při implementaci. Požadavku na zajištění kvalitní uživatelské zkušenosti jsem věnoval pozornost při návrhu, implementaci a také testování. I přes to je takřka nemožné v první verzi aplikace dosáhnout dokonalé uživatelské zkušenosti. Proto bude třeba při budoucím vývoji nadále věnovat pozornost uživatelskému testování a přizpůsobovat aplikaci potřebám uživatelů. Požadavky týkající se možnosti optimalizace aplikace pro SEO a možnosti rozšíření aplikace na další platformy byly splněny vhodnou volbou technologií. Díky frameworku Next.js bude možné stránky aplikace jednodušeji přizpůsobit pro SEO. Z frontendové části aplikace bude také možné vytvořit progresivní webovou aplikaci, kterou budou moci uživatelé využívat na požadovaných platformách.

Společně s implementací frontendové části aplikace se mi podařilo vytvořit kompletní mock REST API, který odpovídá reálnému API backendové části aplikace. Mock jsem následně nasadil do testovacího prostředí pomocí služby Netlify. Společně s mockem jsem do testovacího prostředí nasadil i frontend aplikace, který tento mock využíval.

Frontend jsem také nasadil do produkčního prostředí, ve kterém využívá reálné REST API backendové části této aplikace. Jelikož nebyly v rámci backendu implementovány všechny funkční požadavky, musel jsem některé funkcionality z počáteční produkční verze frontendu vyřadit. Nicméně všechny tyto funkce jsou již v rámci frontendu dokončené a bude je tak snadné v budoucnu do produkční verze aplikace zařadit.

V rámci vývoje jsem využíval několik technologií, díky kterým je aplikace připravena pro budoucí vývoj. Díky knihovně Next.js a SWR bude možné aplikaci dále optimalizovat tak, aby měla lepší výkon a rychlejší načítání. Knihovna Storybook zase bude moci sloužit jako základ pro budoucí automatizované testování aplikace a pro její podrobnější dokumentaci.

V rámci testování použitelnosti aplikace bylo odhaleno několik příležitostí pro zlepšení uživatelského rozhraní. Většinu připomínek a nalezených problémů jsem opravil v rámci této práce.

Některé problémy ovšem budou vyžadovat podrobnější návrh a další iterace uživatelského testování. Těmto změnám jsem se proto rozhodl v této práci dále nevěnovat a ponechat je tak na budoucím vývoji aplikace. Dalším výstupem testování byly výsledky dotazníku *System Usability Scale*, na základě kterých aplikace získala skóre 82,5. Aplikace tak má na základě tohoto dotazníku *dobrou použitelnost*.

Celkově jsem tak v rámci tohoto projektu položil základy této aplikace, na kterých bude možné aplikaci dále rozšiřovat a zlepšovat ji tak, aby co nejlépe vyhověla potřebám uživatelů.

4.4.2 Návrh úprav do budoucna

Aplikaci bude možné v budoucnu zlepšovat nejen z hlediska přidávání nových funkcionalit, ale i z hlediska údržby, testování, zvýšení kvality kódu a efektivity vývoje.

Prvním podstatným návrhem je postupné upravování aplikace na základě potřeb uživatelů a zlepšování uživatelské zkušenosti. V rámci uživatelského testování bylo odhaleno několik nedostatků a příležitostí pro zlepšení frontendové části aplikace. Většina objevených nedostatků již byla opravena jako součást této práce. Jiné ovšem vyžadují podrobnější návrh a další iterace uživatelského testování. Na to by se měl zaměřit budoucí vývoj této aplikace.

Zadruhé, jak jsem již dříve nastínil, bude vhodné do aplikace zařadit určitou formu automatizovaných testů. Ty budou moci do budoucna poskytnout jednodušší rozšiřitelnost aplikace a větší důvěru ve funkčnost kódu. Testy tak poskytnou větší stabilitu celé aplikace, která je se zvyšujícím se počtem uživatelů klíčová.

Dalším návrhem je zprovoznění knihoven pro analýzu chování uživatelů. V rámci této aplikace je kvalitní uživatelská zkušenost klíčová pro udržení uživatelů. Sběr a analýza dat o chování uživatelů by mohla pomoci odhalit, která cvičení v aplikaci uživatelům například dělají problémy a do jaké míry jsou jednotlivé funkce aplikace využívány. Tato data by pak mohla sloužit jako podklad k určení priorit projektu a dalšího směru vývoje.

Posledním návrhem je soustředění vývoje na vytvoření progresivní webové aplikace, kterou bude možné stáhnout a nainstalovat na různé platformy. Díky tomu by byla aplikace pro uživatele dostupnější a jednodušší na používání.

Aplikace již obsahuje klíčové funkcionality, se kterými jsou uživatelé schopni se učit anglický jazyk. Budoucí vývoj by se proto, dle mého názoru, neměl zaměřovat na přidávání velkého množství nových funkcionalit, ale spíše na neustálou optimalizaci a zvyšování kvality již existujících funkcí. Konkrétně by se měl vývoj soustředit na kvalitu samotné výuky, a to zejména na její efektivitu, zábavnost a přizpůsobitelnost. Toho lze docílit zejména pravidelným uživatelským testováním, díky kterému lze odhalit řadu příležitostí pro zlepšení celé aplikace.

Celkově tak lze aplikaci rozšiřovat v mnoha různých ohledech. Dle mého názoru by se měl budoucí vývoj primárně zaměřit na zvýšení kvality a stability již existujících funkcí. Toho bude možné docílit například zprovozněním automatizovaných testů a prováděním pravidelného uživatelského testování.

Závěr

Cílem této práce bylo vytvořit a nasadit funkční frontend aplikace na učení se angličtiny. K dosažení tohoto cíle byly provedeny dílčí kroky vývoje softwaru včetně analýzy, návrhu, implementace, testování a následného nasazení.

V rámci vývoje byla nejprve provedena analýza existujících aplikací na výuku jazyků. Do této analýzy byly zahrnuty aplikace Duolingo, Mondly, Babbel a Busuu. Následně byl proveden návrh front-endové části aplikace, který zahrnoval sepsání případů užití, návržení uživatelského rozhraní pomocí nástroje Figma a zvolení vhodných technologií, které byly využívány v dalších fázích vývoje. Po návrhu následovala implementace, v rámci které byl vytvořen samotný frontend aplikace pomocí technologií React, Next.js a TypeScript. V této fázi vývoje byl také vytvořen mock REST API, který aplikace využívala ve vývojovém prostředí. Při implementaci bylo prováděno manuální testování jednotlivých komponentů a stránek aplikace. Po dokončení implementace bylo dále provedeno uživatelské testování zaměřené na použitelnost aplikace. Výstupem testování byl poznatek, že má aplikace dobrou použitelnost a je tak připravená pro běžné používání. Testování také odhalilo několik příležitostí pro zlepšení uživatelského rozhraní. Po dokončení testování byl frontend nasazen do produkčního prostředí.

Na konci procesu vývoje byla front-endová část aplikace zhodnocena a byly navrženy možné úpravy do budoucna. Aplikaci bude možné rozšiřovat v mnoha různých ohledech. Budoucí vývoj by se měl zaměřit především na zvýšení kvality již existujících funkcí a celkové uživatelské zkušenosti.

Celkově tak bylo dosaženo všech vytyčených cílů této bakalářské práce. Výsledkem práce je nasazený a funkční frontend aplikace na učení se angličtiny. Díky aplikaci se tak mohou uživatelé začít zdokonalovat ve většině aspektech anglického jazyka včetně gramatiky, slovní zásoby, výslovnosti, poslechu a čtení.

Výsledky dotazníku před uživatelským testováním

V této příloze lze nalézt přepsané odpovědi na krátký dotazník, který jsem předložil k vyplnění jednotlivým účastníkům testování před zahájením testování. Dotazník sloužil primárně pro získání základních informací o daném účastníkovi a o jeho zkušenostech s podobnými aplikacemi na učení se jazyků.

Účastník testování 1

Jméno:	Marie
Pohlaví:	Žena
Věk:	23
Jazyky, které se učí:	Učí se pravidelně anglicky, zejména formou čtení a sledování seriálů.
Aplikace, které používá:	Nepoužívá aktivně žádné aplikace na učení se jazyků. Dříve používala aplikaci Duolingo.
Nedostatky, které vidí v jiných aplikacích:	Aplikace nevyučují látku pokročilých úrovní angličtiny. Dále nelze v aplikacích jednoduše přecházet mezi jednotlivými úrovněmi angličtiny.
Odhadovaná úroveň angličtiny:	B2 až C1

Účastník testování 2

Jméno:	Jiří
Pohlaví:	Muž
Věk:	23
Jazyky, které se učí:	Učí se francouzštinu, němčinu, italštinu, angličtinu.
Aplikace, které používá:	Používá řadu aplikací jako například Duolingo nebo Kwiziq.
Nedostatky, které vidí v jiných aplikacích:	Aplikace Duolingo je málo přizpůsobitelná a je příliš repetitivní.
Odhadovaná úroveň angličtiny:	C1

Účastník testování 3

Jméno:	Jakub
Pohlaví:	Muž
Věk:	23
Jazyky, které se učí:	Učí se angličtinu a němčinu přibližně 20 minut denně.
Aplikace, které používá:	Používá pravidelně aplikaci Duolingo.
Nedostatky, které vidí v jiných aplikacích:	Žádné nedostatky na ostatních aplikacích nevidí.
Odhadovaná úroveň angličtiny:	B2

Účastník testování 4

Jméno:	Tomáš
Pohlaví:	Muž
Věk:	56
Jazyky, které se učí:	Momentálně se žádné jazyky neučí, chce se začít učit anglicky.
Aplikace, které používá:	Žádné nepoužívá ani žádné dříve nepoužíval.
Nedostatky, které vidí v jiných aplikacích:	Žádné nedostatky na ostatních aplikacích nevidí.
Odhadovaná úroveň angličtiny:	A2 až B1

Účastník testování 5

Jméno:	Anna
Pohlaví:	Žena
Věk:	20
Jazyky, které se učí:	Učí se angličtinu a němčinu jednou za několik dní.
Aplikace, které používá:	Používá aktivně aplikaci Duolingo.
Nedostatky, které vidí v jiných aplikacích:	Aplikace nejsou efektivní, zejména učení se slovní zásoby.
Odhadovaná úroveň angličtiny:	C1

Výsledky dotazníku SUS

Po dokončení testování jsem účastníkům testování dal k vyplnění dotazník *System Usability Scale*, který se skládal z následujících otázek.

1. Myslím si, že bych tuto aplikaci rád používal pravidelně.
2. Myslím si, že je aplikace příliš komplexní.
3. Myslím si, že se aplikace snadno používá.
4. Myslím si, že bych potřeboval pomoc někoho technicky zdatného, abych mohl používat tento systém.
5. Jednotlivé funkce v aplikaci jsou dobře zpracované.
6. Myslím si, že byl systém příliš nekonzistentní.
7. Umím si představit, že by se většina lidí naučila aplikaci používat velice rychle.
8. Aplikace mi přišla velice těžko ovladatelná.
9. Při používání aplikace jsem se cítil velice jistě.
10. Potřeboval bych se naučit hodně věcí, než bych mohl začít aplikaci používat.

Na každou otázku účastníci testování odpovídali zaškrtnutím jednoho z pěti políček na škále od možnosti *Naprosto souhlasím* po možnost *Naprosto nesouhlasím*. Odpovědi převedené na číselnou reprezentaci lze nalézt v tabulce B.1. Číslo 1 odpovídá možnosti *Naprosto nesouhlasím* a číslo 5 možnosti *Naprosto souhlasím*.

	Účastník 1	Účastník 2	Účastník 3	Účastník 4	Účastník 5
Otázka 1	3	4	5	5	5
Otázka 2	3	1	2	2	3
Otázka 3	4	4	4	5	4
Otázka 4	1	2	1	1	1
Otázka 5	4	3	5	3	5
Otázka 6	1	1	1	1	1
Otázka 7	4	4	5	4	5
Otázka 8	2	2	2	1	3
Otázka 9	3	4	4	4	4
Otázka 10	2	1	2	1	1

■ **Tabulka B.1** Odpovědi účastníků dotazníku SUS

Výstupy uživatelského testování

V této příloze lze nalézt seznam problémů a příležitostí pro zlepšení aplikace, které byly objeveny v rámci uživatelského testování. Každému problému jsem přiřadil jeho závažnost, která může být *nízká, střední, vysoká* nebo *kritická*.

Problém 1 – Nabídka písmen při sestavování slov

- Popis:** U cvičení, v němž uživatel sestavuje slovo z nabízených písmen, se nabídka písmen při použití písmene vizuálně zužuje směrem ke středu obrazovky. Písmena z nabídky se tak po obrazovce pohybují, což značně zpomaluje vyplňování cvičení, protože se při každém výběru písmene nabídka vizuálně změní.
- Možné řešení:** Změnit uživatelské rozhraní tak, aby se nabídka písmen po vybrání libovolného písmene nijak neposouvala.
- Závažnost:** Střední
- Stav:** Opraveno

Problém 2 – Nevýrazná zpětná vazba při kontrole cvičení

- Popis:** Při děláni cvičení uživatel po kliknutí na tlačítko *Zkontrolovat* nedostává dostatečně výraznou zpětnou vazbu, zda udělal cvičení správně nebo špatně. Uživatel si tak nemusí všimnout, že se tlačítko *Zkontrolovat* změnilo na tlačítko *Pokračovat*. To může vést ke zmatení uživatele.
- Možné řešení:** Změnit uživatelské rozhraní tak, aby uživateli dávalo zřetelnější a jasnější zpětnou vazbu po tom, co stiskl tlačítko *Zkontrolovat*. Aplikace by měla jasněji uživateli sdělit, zda vyplnil dané cvičení správně nebo špatně. Toho lze docílit například výraznější změnou barvy některých prvků uživatelského rozhraní.
- Závažnost:** Střední
- Stav:** Neopraveno. Oprava bude vyžadovat podrobnější návrh a další uživatelské testování.

Problém 3 – Klávesnice zakrývá tlačítko pro kontrolu

Popis:	U některých cvičení mobilní klávesnice překrývá tlačítko <i>Zkontrolovat</i> . Uživatel tak musí po vyplnění cvičení klávesnicí zavřít, nebo scrollovat na obrazovce až ke zmíněnému tlačítku. To může být při vyplňování cvičení zdlouhavé.
Možné řešení:	Aplikace by měla buď zobrazovat tlačítko <i>Zkontrolovat</i> nad mobilní klávesnicí, nebo umožnit zkontrolování cvičení pomocí potvrzovacího tlačítka na dané klávesnici.
Závažnost:	Střední
Stav:	Opraveno

Problém 4 – Málo detailní statistiky učení

Popis:	Po dokončení učení aplikace zobrazuje málo detailní statistiky o daném učení. Nyní aplikace ukazuje pouze graf s popiskem. Většina účastníků testování zmínila, že by si zde představovala například zobrazení procenta správných odpovědí.
Možné řešení:	Aplikace by měla na stránce se statistikou zobrazit detailnější statistiky. Například by aplikace mohla zobrazit procento správných odpovědí.
Závažnost:	Nízká
Stav:	Opraveno

Problém 5 – Matoucí měnění úrovně na mapě učení

Popis:	Když si většina účastníků testování chtěla změnit úroveň angličtiny na mapě učení, pouze přešla do sekce mapy učení, která odpovídala dané úrovni angličtiny. Tato akce ovšem v aplikaci nijak nezmění pozici uživatele na mapě učení. Uživatel by měl kliknout na libovolnou lekci v dané úrovni mapy učení a zvolit možnost pro přesunutí se na danou lekci. Tím se teprve přesune do dané úrovně angličtiny.
Možné řešení:	Přepínání mezi úrovněmi mapy učení by se nemělo provádět v postranní nabídce aplikace, ale přímo na stránce mapy učení. Po přepnutí na jinou úroveň by měla aplikace automaticky přesunout uživatele na první lekci v dané úrovni. Dále by měla aplikace výrazněji znázorňovat, na které lekci se uživatel v rámci mapy nachází. Tyto změny by měly vést ke srozumitelnější navigaci a jednoduššímu přecházení mezi úrovněmi mapy učení.
Závažnost:	Vysoká
Stav:	Neopraveno. Oprava bude vyžadovat podrobnější návrh, úpravu backendové části aplikace a další uživatelské testování.

Problém 6 – Nevýrazná funkce vyhledávání

Popis:	V aplikaci existuje funkce pro vyhledávání lekcí, uživatelů a slovíček. Tuto funkci ovšem žádný účastník při testování nevyužil. Tato funkce by proto mohla být pro uživatele lépe dohledatelná.
Možné řešení:	Možnost vyhledávání by měla být v postranní nabídce aplikace přesunuta výše, aby byla pro uživatele jednodušeji dohledatelná.
Závažnost:	Nízká
Stav:	Opraveno

Problém 7 – Tlačítka u poslechových cvičení

Popis:	V poslechovém cvičení, u kterého uživatel spojuje slovo a jeho audionahrávku, je tlačítko s audionahrávkou rozděleno do dvou částí. První část spouští audio a druhá část vybere dané tlačítko. To je pro uživatele při prvních interakcích lehce matoucí. Účastníci testování očekávali, že se tlačítko vybere, když ho stisknou na jakémkoliv místě.
Možné řešení:	Tlačítko by mělo být zjednodušeno tak, aby se při kliknutí na něj spustila nahrávka a automaticky se vybralo, nehledě na to, na jakou část tlačítka uživatel klikne.
Závažnost:	Střední
Stav:	Opraveno

Problém 8 – Vyhledávací pole

Popis:	Vyhledávací pole je pro uživatele matoucí, jelikož musí uživatel pro spuštění vyhledávání stisknout ikonu s lupou. Většina účastníků testování očekávala, že se bude vyhledávání spouštět tlačítkem na klávesnici daného mobilního zařízení. Někteří uživatelé naopak očekávali, že bude aplikace vyhledávání provádět automaticky a rovnou bude při vyplňování vyhledávacího pole ukazovat výsledky vyhledávání.
Možné řešení:	Aplikace by měla při vyhledávání automaticky zobrazovat výsledky vyhledávání v průběhu psaní do vyhledávacího pole, nebo by měla umožňovat spuštění vyhledávání pomocí potvrzovacího tlačítka na mobilní klávesnici.
Závažnost:	Střední
Stav:	Opraveno

Problém 9 – Malé tlačítko pro pokračování

Popis:	Tlačítko <i>Pokračovat</i> , které se zobrazuje ve spodní části obrazovky například při vyplňování formulářů nebo při učení, je dle účastníků testování příliš malé. Uživatelé tak mají někdy problém tlačítko stisknout.
Možné řešení:	Všechna tlačítka, která se takto zobrazují ve spodní části obrazovky, by měla být zvětšena tak, aby je bylo možné jednoduše stisknout.
Závažnost:	Střední
Stav:	Opraveno

Problém 10 – Nejasné přidávání slovíček do vlastní lekce

Popis:	Při přidávání slovíček do vlastní lekce si někteří uživatelé nebyli jistí, zda bylo slovíčko kliknutím na tlačítko ve tvaru znaménka plus přidáno.
Možné řešení:	Aplikace by měla uživatelům po kliknutí na dané tlačítko zobrazit zprávu, že bylo slovíčko úspěšně přidáno.
Závažnost:	Nízká
Stav:	Opraveno

Bibliografie

1. CURRY, David. *Language Learning App Revenue and Usage Statistics (2024)* [online]. 2024-01-08. [cit. 2024-04-07]. Dostupné z: <https://www.businessofapps.com/data/language-learning-app-market/>.
2. BLANCO, Cindy. *2022 Duolingo Language Report* [online]. 2022-12-06. [cit. 2024-04-07]. Dostupné z: <https://blog.duolingo.com/2022-duolingo-language-report/>.
3. DUOLINGO. *Shape Language* [online]. [cit. 2024-04-07]. Dostupné z: <https://design.duolingo.com/illustration/shape-language>.
4. HARTMAN, Greg. *Building character: How a cast of characters can help you learn a language* [online]. 2020-11-10. [cit. 2024-04-07]. Dostupné z: <https://blog.duolingo.com/building-character/>.
5. APRAMEYA, Lavanya. *Improving Duolingo, one experiment at a time* [online]. 2020-01-10. [cit. 2024-02-26]. Dostupné z: <https://blog.duolingo.com/improving-duolingo-one-experiment-at-a-time/>.
6. DETERDING, Sebastian; DIXON, Dan; KHALED, Rilla; NACKE, Lennart. From game design elements to gamefulness: defining “gamification”. In: *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments*. Tampere, Finland: Association for Computing Machinery, 2011, s. 9–15. MindTrek '11. ISBN 9781450308168. Dostupné z DOI: 10.1145/2181037.2181040.
7. KIM, Huhn; KIM, Miyoung; KIM, Jihye; PARK, Hyejeong. The Effects of Load Time, Contents, Loading Screen, and Animation Type on User Satisfaction with the Load Speed of Mobile Apps. In: *Archives of Design Research*, 2014, sv. 27, s. 107–119. Č. 2. Dostupné z DOI: 10.15187/adr.2014.05.110.2.107.
8. GOOGLE. *Material Design 2* [online]. [cit. 2024-03-08]. Dostupné z: <https://m2.material.io/>.
9. GOMAA, Hassan. Software modeling and design: UML, use cases, patterns, and software architectures. In: Cambridge [U.K.]: Cambridge University Press, 2011, s. 74–78. ISBN 978-0-521-76414-8.
10. HUDAIB, Amjad; MASADEH, Raja; HAJ QASEM, Mais; ALZAQEBAH, Abdullah. Requirements Prioritization Techniques Comparison. *Modern Applied Science*. 2018, roč. 12, č. 2, s. 64–65. ISSN 1913-1844. Dostupné z DOI: 10.5539/mas.v12n2p62.
11. TALARICO, Donna. What is mobile-first design? Why does it matter? *Recruiting & Retaining Adult Learners*. 2019, roč. 22, č. 2, s. 3–3. ISSN 2155-644X. Dostupné z DOI: <https://doi.org/10.1002/nsr.30533>.

12. FIGMA. *Components, styles, and shared library best practices* [online]. [cit. 2024-03-10]. Dostupné z: <https://www.figma.com/best-practices/components-styles-and-shared-libraries/>.
13. STACK OVERFLOW. *Most popular technologies* [online]. 2023-06-13. [cit. 2024-03-11]. Dostupné z: <https://survey.stackoverflow.co/2023/#section-most-popular-technologies-web-frameworks-and-technologies>.
14. MOZILLA. *Progressive web apps* [online]. 2023-10-25. [cit. 2024-03-11]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps.
15. BLAZOR. *Blazor moved to the ASP.NET Core repo* [online]. 2019-04-25. [cit. 2024-03-11]. Dostupné z: <https://github.com/dotnet/blazor>.
16. NPM. *The JavaScript Package Registry* [online]. 2023-01-05. [cit. 2024-03-11]. Dostupné z: <https://docs.npmjs.com/cli/v10/using-npm/registry>.
17. MICROSOFT. *NuGet* [online]. ©2024. [cit. 2024-03-11]. Dostupné z: <https://www.nuget.org/>.
18. STACK OVERFLOW. *Admired and Desired* [online]. 2023-06-13. [cit. 2024-03-11]. Dostupné z: <https://survey.stackoverflow.co/2023/#section-admired-and-desired-web-frameworks-and-technologies>.
19. VERCEL. *Rendering and Ranking* [online]. ©2024. [cit. 2024-03-11]. Dostupné z: [https://nextjs.org/learn-pages-router/seo/rendering-and-ranking/rendering-strategies](https://nextjs.org/learn/pages-router/seo/rendering-and-ranking/rendering-strategies).
20. MICROSOFT. *TypeScript for JavaScript Programmers* [online]. 2024-04-12. [cit. 2024-04-15]. Dostupné z: <https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>.
21. MATERIAL UI. *MUI Core* [online]. 2024-03-16. [cit. 2024-03-10]. Dostupné z: <https://github.com/mui/material-ui>.
22. VERCEL. *SWR* [online]. 2023-06-28. [cit. 2024-03-10]. Dostupné z: <https://swr.vercel.app/>.
23. CHACON, Scott; STRAUB, Ben. *8.3 Customizing Git - Git Hooks* [online]. 2024-04-05. [cit. 2024-04-17]. Dostupné z: <https://git-scm.com/book/en/v2/Customizing-Git-Git-Hooks>.
24. MOCKOON. *Templating overview* [online]. 2024-04-02. [cit. 2024-04-17]. Dostupné z: <https://mockoon.com/docs/latest/templating/overview/>.
25. META OPEN SOURCE. *Reusing Logic with Custom Hooks* [online]. 2024. [cit. 2024-03-18]. Dostupné z: <https://react.dev/learn/reusing-logic-with-custom-hooks>.
26. VERCEL. *Mutation and Revalidation* [online]. 2023-08-11. [cit. 2024-03-18]. Dostupné z: <https://swr.vercel.app/docs/mutation#optimistic-updates>.
27. NIELSEN, Jakob. Enhancing the explanatory power of usability heuristics. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Boston, Massachusetts, USA: Association for Computing Machinery, 1994, s. 152–158. CHI '94. ISBN 0897916506. Dostupné z DOI: 10.1145/191666.191729.
28. VACANTI, Daniel; COLEMAN, John. *The Kanban Guide* [online]. 2020-12. [cit. 2024-03-18]. Dostupné z: <https://kanbanguides.org/english/>.
29. SCRUM.ORG. *What is Scrum?* [online]. ©2024. [cit. 2024-04-18]. Dostupné z: <https://www.scrum.org/resources/what-scrum-module>.
30. ATLISSIAN. *Gitflow workflow* [online]. ©2024. [cit. 2024-04-18]. Dostupné z: <https://www.atlassian.com/git/tutorials/comparing-workflows/gitflow-workflow>.

31. MORAN, Kate. *Usability Testing 101* [online]. 2019-12-01. [cit. 2024-03-27]. Dostupné z: <https://www.nngroup.com/articles/usability-testing-101/>.
32. CHROMATIC. *Test runner* [online]. ©2024. [cit. 2024-03-25]. Dostupné z: <https://storybook.js.org/docs/writing-tests/test-runner>.
33. NIELSEN, Jakob. *Why You Only Need to Test with 5 Users* [online]. 2000-03-18. [cit. 2024-03-27]. Dostupné z: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.
34. BANGOR, Aaron; KORTUM, Philip; MILLER, James. Determining what individual SUS scores mean: adding an adjective rating scale. *J. Usability Studies* [online]. 2009, roč. 4, č. 3, s. 114–123 [cit. 2024-04-19]. Dostupné z: <https://dl.acm.org/doi/10.5555/2835587.2835589>.

Slovník pojmů

- L1** Jazyk, který uživatel používá k osvojení si jiného jazyka.
- L2** Jazyk, který si uživatel osvojuje pomocí již naučeného jazyka.

Obsah příloh

	požadavky-aplikace.zip	
	└─ požadavky-aplikace.pdf	požadavky aplikace ve formátu PDF
	└─ https://figma.com/file/wu3UCREJ361T6BSwDVwkJb	návrh uživatelského rozhraní
	└─ https://github.com/Morcinus/Linguino-Frontend	zdrojový kód
	└─ https://github.com/Morcinus/Linguino-API-Mock	mock API
	└─ https://dev.linguino.org/	aplikace ve vývojovém prostředí
	└─ https://app.linguino.org/	aplikace v produkčním prostředí