



**ČESKÉ VYSOKÉ
UČENÍ TECHNICKÉ
V PRAZE**

F3

**Fakulta elektrotechnická
Katedra počítačů**

Bakalářská práce

Webová aplikace pro odvykání kouření

Daniel Mareda
Otevřená Informatika

Leden 2021

Vedoucí práce: doc. Ing. Daniel Novák, Ph.D.

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Mareda** Jméno: **Daniel** Osobní číslo: **474723**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Otevřená informatika**
Studijní obor: **Software**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Webová aplikace pro odvykání kouření

Název bakalářské práce anglicky:

Web application for smoking cessation using web progressive framework

Pokyny pro vypracování:

- 1) Seznamte se s problematikou odvykání kouření
- 2) Zvolte technologii pro implementaci webové aplikace s prvky „Progresivní webové aplikace“ a navrhnete ji tak, aby obsahovala všechny funkce již existující mobilní aplikace pro odvykání kouření
- 3) Implementujte navrženou aplikaci. Vytvořte nejprve prototyp webové aplikace, dle existující mobilní aplikace a poté aplikaci upravte za účelem lepšího uživatelského prožitku a využití možností webové aplikace oproti aplikaci mobilní
- 4) Proveďte studii na menším počtu uživatelů (5 až 10 uživatelů) a vyhodnotte užitečnost aplikace.

Seznam doporučené literatury:

1. Cahill, K., Stevens, S., Perera, R., & Lancaster, T. (2013). Pharmacological interventions for smoking cessation: an overview and network meta-analysis. Cochrane Database Syst Rev(5).
2. Lancaster, T., & Stead, L. F. (2005). Individual behavioural counselling for smoking cessation. Cochrane Database of Systematic Reviews(2).
3. Stead, L. F., Buitrago, D., Preciado, N., Sanchez, G., Hartmann-Boyce, J., & Lancaster, T. (2013). Physician advice for smoking cessation. Cochrane Database of Systematic Reviews(5).

Jméno a pracoviště vedoucí(ho) bakalářské práce:

doc. Ing. Daniel Novák, Ph.D., Analýza a interpretace biomedicinských dat FEL

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

Datum zadání bakalářské práce: **23.09.2020**

Termín odevzdání bakalářské práce: **05.01.2021**

Platnost zadání bakalářské práce: **30.09.2022**

doc. Ing. Daniel Novák, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

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

III. PŘEVZETÍ ZADÁNÍ

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

Datum převzetí zadání

Podpis studenta

Poděkování / Prohlášení

Chtěl bych poděkovat své matce a svým sourozencům za podporu během studia. Děkuji také svému vedoucímu a všem, kdo se podíleli na testování aplikace, za konstruktivní kritiku a cenné nápady na zlepšení. V neposlední řadě děkuji svým spolužákům, především Michalovi Šípkovi, a také přednášejícím a cvičícím, že jsem se od nich mohl učit.

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

V Praze dne 05. 01. 2021

.....

Abstrakt / Abstract

Tato bakalářská práce se zabývá zmapováním existující mobilní aplikace pro odvykání kouření a následným návrhem a implementací webové aplikace sloužící převážně jako doplnění pro uživatele preferující počítače oproti telefonům. Stěžejním rozdílem implementované aplikace oproti standardním aplikacím na stejné téma je série interaktivních sezení, které napodobují schůzku s odborníkem na danou závilost za účelem dodávání motivace, pomáhají kuřákovi vytvořit si svůj vlastní plán k odvykání a poskytují mu odpovědi na jeho případné dotazy. Součástí práce je také prozkoumání progresivních webových aplikací a zkušební implementace vybraných prvků této technologie. Ve svém závěru se práce zabývá otestováním aplikace cílovou skupinou uživatelů a reakcemi na případné připomínky za účelem zlepšení uživatelského prožitku.

Klíčová slova: odvykání kouření; interaktivní terapie; webová aplikace; progresivní webová aplikace.

This bachelor thesis focuses on analysis of existing mobile application for smoking cessation followed by design and implementation of web application serving as a complementing solution to its mobile counterpart for people favoring desktops over phones. The important difference between this application and other applications focusing on the same topic is series of interactive appointments which mimic meeting with real expert on tobacco addiction in order to encourage smoker to stop, help him create his own plan and provide him with answers to frequently asked questions. Part of this thesis also aims to explore progressive web apps and attempts to implement chosen elements of this technology. In the end, this thesis records testing of the application by target audience and offers an overview of improvements made based on feedback from user testing.

Keywords: smoking cessation; interactive therapy; web application; progressive web application.

Title translation: Web application for smoking cessation using web progressive framework

Obsah /

1 Úvod	1
1.1 Motivace.....	1
1.2 Cíl	1
2 Teoretická část	2
2.1 Kouření.....	2
2.1.1 Závislost na kouření	2
2.1.2 Odvykání kouření	2
2.2 Webové aplikace vs mobilní aplikace.....	3
2.2.1 Výhody webových aplikací	4
2.2.2 Výhody mobilních aplikací	4
2.2.3 Preference	4
2.2.4 Návrh webových aplikací ..	4
2.3 Progresivní webové aplikace	5
2.3.1 Vlastnosti	6
2.3.2 Service worker	7
2.3.3 Cache.....	7
2.3.4 Omezení	8
2.4 Framework/Knihovna	8
2.4.1 React	9
2.4.2 Angular.....	10
2.4.3 Rozhodnutí.....	10
3 Analýza	11
3.1 Sezení.....	11
3.2 Funkční požadavky	11
3.3 Obrazovky.....	13
4 Návrh a Implementace	17
4.1 Návrh.....	17
4.1.1 První verze	17
4.1.2 Implementovaná verze ...	18
4.2 Rozšiřitelnost a udržitelnost...	22
4.3 Technologie.....	23
4.3.1 Typescript vs javascript .	23
4.3.2 Ostatní použité technologie a knihovny	23
4.4 Prozkoumání a dokumentace API	24
4.5 Zobrazování sezení	25
4.5.1 Nevýhody prototypu.....	25
4.5.2 Zjednodušené zadání vykreslování sezení.....	25
4.5.3 Potřeba předzpracování dat	26
4.5.4 Další překážky	28
4.5.5 Architektura řešení	28
4.6 Tvorba detailu odznaků a dotazníků.....	28
4.7 Implementace prvků PWA	29
4.8 Co je třeba zlepšit na serveru .	30
4.9 Nasazení.....	31
5 Uživatelské testování	33
5.1 Specifika testování	33
5.2 Dotazník.....	33
5.3 Příkladný scénář	34
5.4 Vyhodnocení	35
5.4.1 Registrace	35
5.4.2 Načítání sezení	35
5.4.3 Drobné změny.....	37
5.4.4 Zhodnocení prvků PWA .	37
6 Závěr	38
6.1 Zhodnocení.....	38
6.2 Rozšíření aplikace.....	38
Literatura	40
A Zkratky	43
B Obsah elektronické přílohy	44

/ **Obrázky**

2.1.	Možnosti podpory motivace při odvykání kouření	3
2.2.	Průměrný strávený čas konzumováním online médií	5
2.3.	Ukázka reportu z nástroje Lighthouse	7
2.4.	Zjednodušené zobrazení životního cyklu Service worker	8
3.1.	Ukázka administrace sezení na backendu	13
3.2.	Mobilní aplikace - úvod do sezení	14
3.3.	Mobilní aplikace - ukázka sezení	14
3.4.	Mobilní aplikace - sekce profil ..	15
3.5.	Mobilní aplikace - sekce plán ..	15
3.6.	Mobilní aplikace - sekce odznaky	15
3.7.	Mobilní aplikace - detail odznaku	15
3.8.	Mobilní aplikace - sekce účet ..	16
3.9.	Mobilní aplikace - dotazník	16
4.1.	Ukázka jednoduchého rozhraní pro průzkum preferencí ..	18
4.2.	Obrazovka registrace	19
4.3.	Obrazovka sezení	19
4.4.	Úvodní obrazovka webové aplikace	20
4.5.	Obrazovka můj plán	20
4.6.	Obrazovka moje podpora	21
4.7.	Obrazovka úspěchy	21
4.8.	Obrazovka můj účet	22
4.9.	Obrazovka detailu odznaku	22
4.11.	Vykreslení ukázkového sezení ..	26
4.10.	Ukázka podoby zjednodušeného sezení	27
4.12.	Deklarativní zadávání nových odznaků v kódu	29
4.13.	Typová definice odznaků za pomoci typescriptu	29
5.2.	Obrazovka přihlášení - doplnění nápovědy	35
5.1.	Obrazovka registrace - změny po testování	36
5.3.	Obrazovka sezení - doplnění načítání	36

Kapitola 1

Úvod

1.1 Motivace

Používání tabákových výrobků je přes své vážné zdravotní dopady na lidské zdraví rozšířené po celém světě. Dle dostupných statistik užívá ve světovém měřítku tabák více než 1.3 miliardy lidí, z čehož každý rok 8 milionů podlehně zdravotním následkům užívání tabáku, tedy z příčin, kterým můžeme zamezit [1]. Situace je natolik vážná, že zdravotní instituce mluví dokonce o tabákové epidemii [2]. Prevence a odvykání od užívání tabákových výrobků jsou důležitou zdravotní výzvou dnešního světa. Výsledky dostupných studií ukazují, že u kuřáků snažících se o odvykání bez vnější pomoci dosáhne pouze velmi nízké procento (v řádu jednotek) lidí svého cíle, a sice dlouhodobé absence bez znovu upadnutí do zaběhlého zvyku [3]. Možností pomoci, kterých se odvykajícímu kuřákovi může dostat, je hned několik, od léků na předpis až po psychosociální intervenci. Zajímavých výsledků dosahují také aplikace kombinující možnost interaktivních textových sezení s gamifikačními elementy [4].

1.2 Cíl

Cílem této práce je navrhnout a naimplementovat webovou aplikaci tak, aby obsahovala funkce již existující mobilní aplikace na odvykání kouření. Náplní práce je tedy vše od zaznamenání požadavků pomocí průzkumu existující aplikace, přes návrh nového uživatelského rozhraní využívajícího možností webových aplikací až po implementaci za pomoci moderního webového frameworku, otestování aplikace s uživateli a zadokumentování celého procesu. Webová aplikace bude přijímat data ze stejného backendového řešení, ze kterého přijímá data mobilní aplikace tak, aby uživatel mohl používat webovou aplikaci ke stejnému účelu jako aplikaci mobilní. Součástí práce je také průzkum technologie progresivních webových aplikací [5] spolu s posouzením a odůvodněním, které vlastnosti progresivních webových aplikací je vhodné naimplementovat jako součást aplikace. Po celou dobu práce na projektu je kladen důraz na rozšiřitelnost aplikace, zejména pomocí výběru správných technologií a dokumentace nejenom nově vytvořeného kódu, ale i rozhraní existujícího backendového řešení, se kterým musí aplikace komunikovat.

Kapitola 2

Teoretická část

2.1 Kouření

Důsledkem užívání tabákových výrobků je každé 6. úmrtí a 80% všech chronických plicních chorob [6]. Úmrtí a rozvoj plicních chorob postihuje nejenom hlavní aktéry procesu kouření, ale také ty nepřímé, tedy lidi, kteří dýchají cigaretový kouř, ač sami nekouří. Pro vážnost situace ohledně kouření hovoří i snaha o systematický boj proti užívání tabákových výrobků na celostátní úrovni. Státy se snaží omezení kouření dosáhnout pomocí několika metod, a to sice například povinným přidáváním obrazových varování na krabičky cigaret, zvýšenou daní na tabákové výrobky a zaváděním nekuřáckých sekcí v restauracích a jiných prostorech.

2.1.1 Závislost na kouření

Přítomnost nikotinu v cigaretách vytváří v člověku závislost, která je podobná závislosti z užívání heroinu. Závislost vzniklá na kouření není pojem označující pouze jeden jev. Nejdříve nastupuje vzorec naučeného chování, tedy závislost psychosociální, které se podobně jako jiných zvyků člověk vzdává jen velmi těžce, neboť si člověk navykne spojit rutinní činnosti (jako například čekání na tramvaj) s kouřením. Po určité době nastupuje fyzická závislost na nikotinu, která je způsobena simulací receptorů v mozku a následným vyplavením hormonů štěstí [7].

2.1.2 Odvykání kouření

Proces odvykání kouření se vyznačuje tím, že většina kuřáků, kteří přestali úspěšně kouřit, učinila předtím několik neúspěšných pokusů, při kterých vědomě rozhodli, že se od daného momentu chtějí nadobro zbavit cigaret. Jako průměrný počet pokusů o odvyknutí od kouření se nejčastěji udává 5 - 7 pokusů [8].

Léčba závislosti se zaměřuje jak na psychosociální tak na farmakologické složky. Mezi používané léky patří především vareniklin a bupropion, pro tuto práci je ale daleko důležitější léčba na psychosociální a behaviorální úrovni v podobě krátkých intervencí. Intervence jsou založené především na empatii a motivaci pacienta k odložení cigaret, například pomocí zdůraznění dopadů na zdraví, které se používají především k přesvědčení pacientů, kteří ještě nejsou plně rozhodnuti v otázce potřeby odvykání. Pakliže situace postoupí správným směrem co se motivace k odvykání týče, pokračují intervence rozebráním nejčastějších situací a přípravě plánu jak se jim vyhybat nebo jak řešit tyto situace bez použití cigaret. Další stěžejní prvky motivace kuřáka k odvykání jsou shrnuty v 2.1

Takové intervence by měly být vedené člověkem s dostatečnou odborností, tedy například adiktologem. Z toho vyplývá, že odvykání je časově náročný proces, jak z pohledu kuřáka tak z pohledu odborné pomoci. Znamená to, že jak kuřák tak odborník na odvykání musí nalézt společný čas a musí strávit mnoho času na zajištění individuality každé intervence, nemluvě o častém opakování si základních cílů, motivace a plánu, který si

relevance	podpořit pacienta v hledání důvodů proč přestat kouřit specifických pro něj (zdravotní stav, rodinná situace apod), relevantně vzhledem k věku či vzdělání
risks	identifikace potenciálních negativních dopadů kouření v souvislosti s pacientovou diagnózou, prognózou, zdůraznění relevantních zdravotních rizik
rewards	identifikace potenciálních přínosů zanechání kouření relevantně k danému pacientovi (zlepšení zdraví, chuti, čichu, úspora apod)
roadbloks	identifikace možných překážek v odvykání (abstinenční příznaky, obava ze selhání, přírůstek hmotnosti, deprese, nedostatku podpory, nedostatek informací o možnostech léčby apod) a poskytnout cílené poradenství
repetition	motivační intervenci opakovat při každém klinickém kontaktu s nemotivovaným pacientem sdělit pacientům, kteří mají za sebou neúspěšné pokusy, že je většinou potřeba více pokusů, než se podaří přestat kouřit

Obrázek 2.1. Možnosti podpory motivace při odvykání kouření, převzato z [6]

kuřák s odborníkem během intervencí vytvoří. V tento moment nastupují na scénu digitální technologie. Aplikace na odvykání kouření může lehce připomínat kuřákovi jeho důvody proč přestává s kouřením a plán, který si s odborníkem vytvořil. Jelikož intervence odborníka s kuřáky mají společnou kostru, situace s aplikací může postoupit až k tomu, kdy odborník vytvoří sérii textových sezení za pomoci jednoduchých prvků jako texty, obrázky a formuláře a aplikace poté na základě kuřákových odpovědí nabízí další vhodná sezení v podobě rad a tipů, tvoření vlastního plánu a zodpovězení častých dotazů. Aplikace na takovém principu již byly testovány na reálných uživateliích a jejich uživatelé zmiňují mezi výhodami následující aspekty [4]:

- Časovou flexibilitu jednotlivých intervencí
- Dlouhotrvající podporu pro každý den
- Autentickou simulaci dialogu
- Celkovou podporu a vytváření motivace
- Posilování vlastní motivace

2.2 Webové aplikace vs mobilní aplikace

Pro vývojáře, kteří chtějí prezentovat uživatelům své digitální řešení určitého problému pomocí aplikace, se nabízí otázka zda zvolit cestu webových nebo mobilních aplikací. Rozhodování by mělo brát v úvahu cílovou skupinu a potřebné funkce dané aplikace. Na začátek uvedme bodový přehled standardního rozdělení webových aplikací a mobilních aplikací. Ač se postupně rozšiřuje technologie progresivních webových aplikací (popsáno v 2.3) tak tato technologie má stále určitá omezení (různá míra podpory na různých platformách). Zároveň pokud již máme existující funkční mobilní aplikaci (příklad této práce) a chceme k ní vytvořit webovou aplikaci tak, abychom umožnili její používání i bez chytrých telefonů, je pro nás následující rozdělení (z pohledu vývojářů) o to relevantnější.

■ 2.2.1 Výhody webových aplikací

- **Snazší aktualizace** - Aktualizace mobilních aplikací musí projít přes příslušný obchod s aplikacemi, kdežto webová aplikace se načítá sama v aktuální verzi při každém vstupu na stránku.
- **Větší dosah** - Ač uživatelé tráví více času na mobilech, tak čas strávený na mobilech je rozložen mezi méně aplikací než čas strávený v prohlížeči mezi navštívené stránky [9].
- **Nižší poplatky při přijímání plateb** - Webové aplikace sice nemají jednotné rozhraní pro platbu, ale na druhou stranu se nemusí tvůrce dělit tolik o své zisky. Poplatek za používání platební brány je řádově nižší než poplatek za používání plateb od Googlu a Applu. [10–11].
- **Dohledatelnost** - Webové aplikace jsou snadněji dohledatelné (internetový vyhledávač oproti app storu).

■ 2.2.2 Výhody mobilních aplikací

- **Více stráveného času** - Co do celkového času uživatelé tráví více volného času konzumací online obsahu na telefonech [12].
- **Existující možnost plateb** - Mobilní aplikace poskytují již implementované možnosti platby pomocí app storu, přidat do aplikace platby je tedy jednodušší.
- **Uložiště a offline fungování** - Mobilní aplikaci je snazší přizpůsobit offline fungování neboť má lepší přístup k uložišti dat a lidé jsou zvyklí instalovat mobilní aplikace do mobilu narozdíl od webových aplikací.
- **Bohatší nativní API** - Mobilní aplikace mohou využívat bohatší (nativní) API oproti API webových prohlížečů, například různé senzory na telefonu (gyroskop), nebo push notifikace, které zatím nefungují ve všech prohlížečích.

■ 2.2.3 Preference

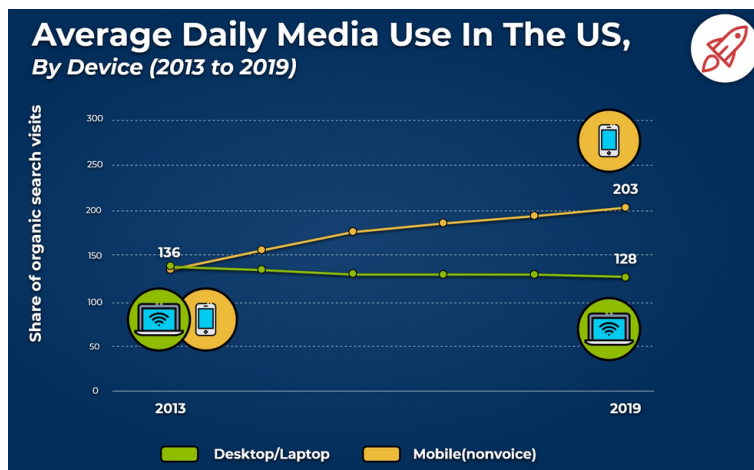
Ač lidé tráví čím dál ze svého času konzumováním internetového obsahu na telefonech, počítače mají v konzumování digitálního obsahu stále své místo. Podle dostupných výzkumů je podstatný rozdíl už mezi věkovou skupinou 39-54 let a lidí mladších 38 let. Skupina mladších 38 uvedla, že 54.2% preferuje k digitálním aktivitám telefon, zatímco 29.5% preferuje počítače. U starší z těchto skupin preferovalo 55.1% používání počítače oproti 34% preferujícím chytrý telefon. Čím starší věková skupina, tím byl rozdíl mezi telefonem a počítačem větší (ve prospěch počítače) [13]. Z toho průzkumu lze extrahovat dva hlavní závěry:

- Starší skupiny uživatelů preferují využití počítačů oproti mobilním telefonům
- I u mladších generací existuje významné procento, které preferuje počítače

■ 2.2.4 Návrh webových aplikací

Jelikož se tato práce věnuje implementaci webové aplikace, která má fungovat společně s mobilní aplikací, uvěďme ještě charakteristiky webové aplikace oproti mobilní v souvislosti uživatelskou přívětivostí a práce s aplikací. Webové aplikace se po stránce interakce vyznačují:

1. **Absence nutnosti instalace** - Uživatel má k vyzkoušení aplikace přes webový prohlížeč kratší cestu než k instalaci aplikace z obchodu aplikací.
2. **Více prostoru a větší písmo** - Počítače umožňují vložit více informací na jednu stránku a zároveň mohou zobrazit větší písmo.



Obrázek 2.2. Průměrný strávený čas konzumováním online médií (kam spadají knihy, hudba, sociální média a další online média) na telefonech a počítačích, převzato z [12]

3. **Multitasking** - Počítače obvykle umožňují uživatelům dělat více věcí najednou, například pomocí rozvržení dvou webových aplikací vedle sebe.
4. **Pohodlnější zadávání dat** - Za pomoci klávesnice a kurzoru myši uživatel zvládá editovat a vytvářet data snadněji než na mobilních přístrojích.
5. **Lepší ochrana soukromí** - Fakt, že mobilní aplikace mají lepší přístup k funkcím telefonu než webové aplikace k funkcím počítače, bývá považován za výhodu. Podíváme-li se na to z druhé strany mince, mobilní aplikace mohou také žádat o více nepotřebných oprávnění či zneužívat citlivé údaje. Zatímco ve webových prohlížečích jsou rozšířené doplňky na zlepšení ochrany soukromí, pro mobilní aplikace takové doplňky rozšířené nejsou. Dostupná studie ukazuje, že u zkoumaných aplikací, které nabízely jak webovou tak mobilní variantu, umožňovaly mobilní aplikace větší únik uživatelských dat, například identifikátorů zařízení. [14].

Navrhář uživatelských rozhraní si musí být vědom rozdílného potenciálu a ovládání mobilních telefonů oproti desktopovým aplikacím, které plynou částečně z rozdílu v interakci a využívání obou typů přístrojů a kterými jsou zejména: [15]:

- **Sloupce** - Jelikož je k dispozici více prostoru, návrh webové aplikace se nemusí a neměl by se omezovat na informace uspořádané v jednom sloupci, ale informace by měly být prezentovány i vedle sebe do případné mřížky.
- **Ovládání** - Zatímco při práci na mobilních telefonech uživatel využívá gesta a k zadávání informací je potřeba virtuální klávesnice, na počítačích má uživatel k dispozici kurzor myši, může vyvolávat události nejenom na kliknutí, ale třeba také na přejetí myši přes určitý prvek a také může používat klávesové zkratky. Tam, kde by vývojář rád využíval k ovládání gesta pomocí ruky musí být naopak kreativnější a vytvořit jiný způsob.
- **Množství vstupů** - Jelikož myš a klávesnice nabízejí snazší způsob vkládání a editování dat, může návrhář uživateli nabídnout k vyplnění více vstupů nebo žádat uživatele o přesnější zadávání dat.

2.3 Progresivní webové aplikace

PWA si kladou za cíl využívat nové funkce moderních prohlížečů a přiblížit tak prožitek z používání webové aplikace na úroveň nativní aplikace dostupné na různých platfor-

mách a kombinovat výhody webových aplikací s mobilními (push notifikace, přístup k uložišti). Celý koncept PWA je vcelku nový a standardy nejsou zatím příliš ustálené nebo stále probíhá práce na jejich implementaci [5].

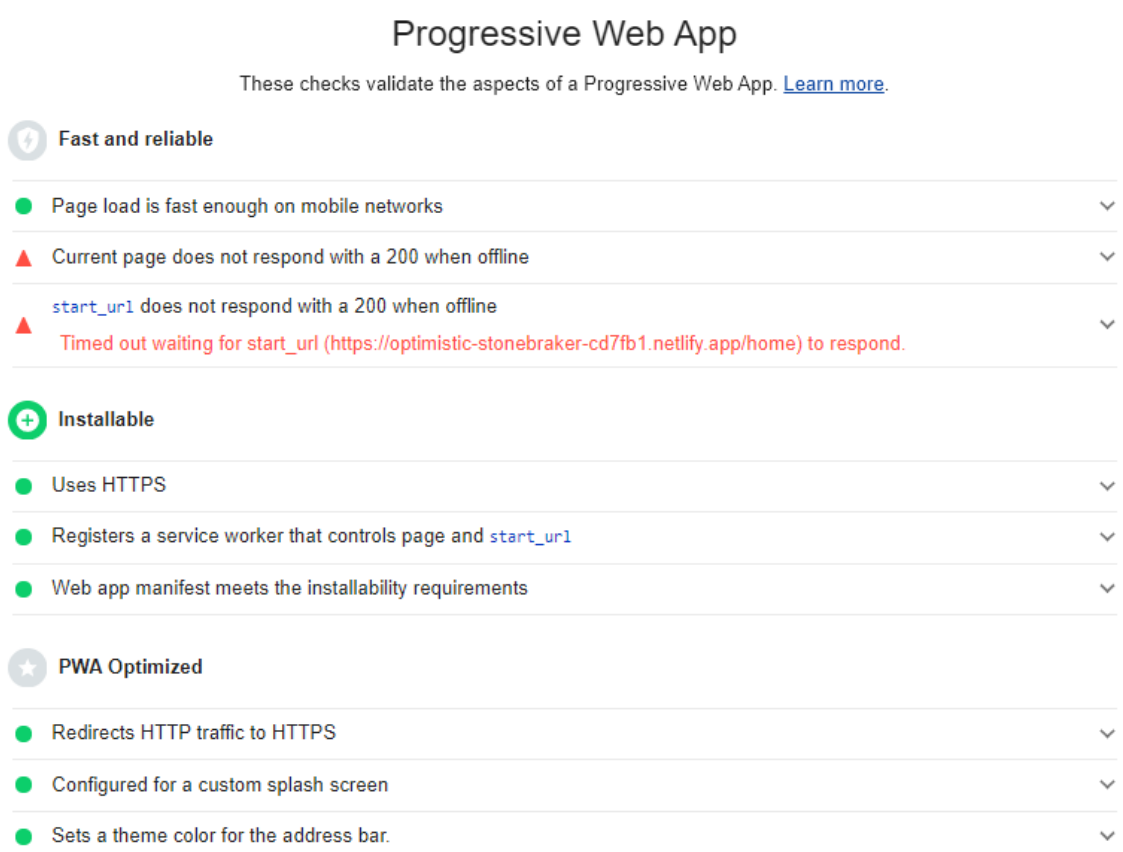
2.3.1 Vlastnosti

Koncept PWA dokážeme nejlépe charakterizovat pomocí soupisu vlastností a požadavků, které má taková aplikace splňovat. [16–17].

- **Progresivní** - Tento pojem zdůrazňuje přednost obsahu před pokročilými vlastnostmi. Pokud aplikaci používáme na zařízeních, které neumí využívat nejnovějších technologií, uživatel musí mít stále možnost aplikaci používat. Požitek z aplikace se poté progresivně zlepšuje podle možností a funkcionalit daného zařízení.
- **Responzivní** - Důraz na responzivitě byl kladen už před zavedením pojmu PWA. Uživatel by měl mít možnost prohlédnout si aplikaci na jakémkoliv zařízení tak, aby neutrpěly jeho dojmy a použitelnost aplikace. Tento pojem se vztahuje například (ale nejenom) k celkovému rozvržení aplikace a velikosti prvků.
- **Instalovatelná** - Dlouhou dobu platilo, že hlavním místem pro instalaci aplikací byly centralizované digitální obchody jednotlivých mobilních operačních systémů, tedy převážně Google Play a App Store. Nový přístup přidává sjednocenou možnost instalování aplikací na mobilech i desktopových zařízeních, kdy aplikaci můžeme jednoduše nainstalovat z prohlížeče a poté ji spouštět například pomocí ikony na ploše.
- **Nezávislá na připojení** - Jedním ze stěžejních principů je fungování bez připojení nebo pouze s pomalým internetovým připojením. Nemůžeme samozřejmě nahradit všechny funkce kompletně bez připojení, ale můžeme například ukázat uživateli poslední verzi stránky, kterou viděl, když byl ještě připojen.
- **Aktuální** - S fungováním offline přichází zároveň nová výzva, a to sice ukazovat uživateli aktuální obsah. Programátor musí vybrat správnou strategii při načítání stránek a případně upozornit uživatele, že je dostupná aktualizace obsahu nebo například aktualizovat periodicky obsah na pozadí.
- **Podobná nativní aplikaci** - Ideální PWA by měla být téměř nerozlišitelná od nativních aplikací.
- **Bezpečná** - Toto naráží především na výměnu veškerých informací mezi aplikací a serverem pomocí zašifrovaného a bezpečného spojení, což je realizováno pomocí SSL certifikátů.
- **Dohledatelné** - Uživatel může objevit aplikaci pomocí vyhledávačů jako například Google.
- **Odkazovatelné** - Aplikaci a její dílčí části lze sdílet pomocí odkazu jako je tomu u obyčejné webové stránky.
- **Znovuzapojení uživatele** - Aplikace dokáže rozpoznat, kdy byla zveřejněna nová aktualizace a dokáže uživatele upozornit na tuto skutečnost nebo aktualizaci stáhnout.

Některé z těchto pojmů je snadné definovat ale hůře ověřit v praxi. Testování potřebných vlastností ulehčuje programátorům google se svým doplňkem **Lighthouse**, který je součástí vývojářských nástrojů v prohlížeči Chrome. Tento nástroj je schopný automaticky vygenerovat report, jehož část je viditelná na obrázku 2.3:

Můžeme tak lehce zjistit co brání tomu, aby šla naše aplikace nainstalovat, zdali musíme doplnit nějaké další informace aby se správně zobrazovala ikonka po nainstalování, nebo v neposlední řadě jestli nalezl doplněk nějaké problémy při offline fungování.



Obrázek 2.3. Ukázka reportu z nástroje Lighthouse pro otestování vlastností progresivních webových aplikací

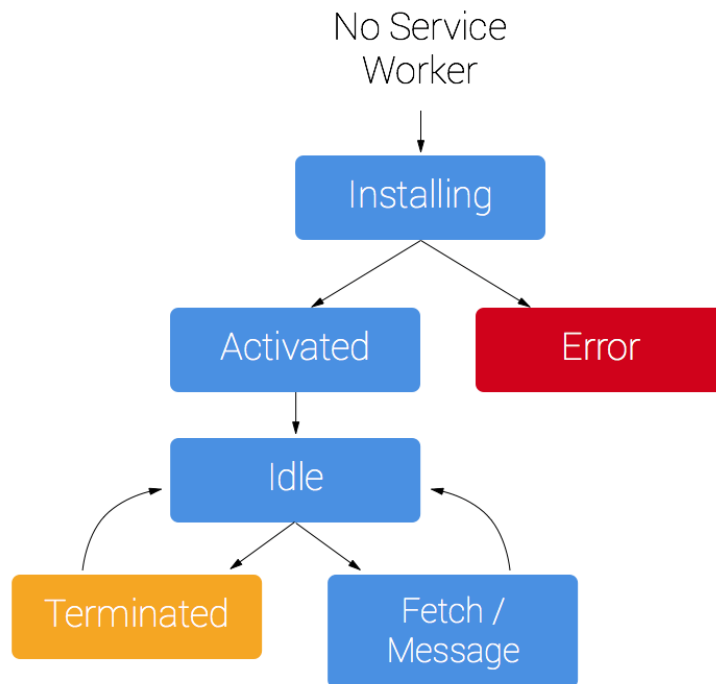
2.3.2 Service worker

Stěžejním konceptem pro bohaté progresivní aplikace je technologie service worker. Jedná se o skripty, které běží na pozadí, tedy současně s ostatními funkcemi webové aplikace. Hlavními funkcemi těchto skriptů je v tuto chvíli zpracovávání požadavků odeslaných na server a případně jejich odpovědí. Pomocí této funkcionality můžeme dosáhnout poskytnutí offline zážitku uživatelům aplikace.

Service worker má vlastní životní cyklus (zachyceno na obrázku 2.4) nezávislý na zbytku webové stránky. Nejdříve proběhne instalace skriptu, při které mohou být uloženy například statické soubory. Po úspěšné instalaci následuje aktivace, kdy je možné vymazat staré již nepotřebné uložené soubory. Jakmile je service worker aktivován, nic již nebrání tomu aby přebíral kontrolu nad odesílanými dotazy a přijatými odpověďmi, které může dále zpracovávat [18].

2.3.3 Cache

Pravděpodobně nejzajímavějším přírůstkem k vlastnostem PWA oproti běžným webovým aplikacím je možnost fungovat offline. K tomuto účelu slouží API Cache [19]. Pokud aplikace získává ze serveru data, na která se už jednou ptala, můžeme tyto data uložit a poskytnout je rovnou při příštím stejném dotazu z daného uložště místo posílání nového dotazu. Tento přístup však přináší další výzvu, a sice jak zajistit aby uživatel získal aktuální data. Nedokážeme totiž poznat z dotazu zdali se data náhodou nezměnila a zdali není potřeba aktualizovat uloženou hodnotu. V přístupu k práci s cache ve spojitosti s offline fungování existují tyto hlavní strategie [20]:



Obrázek 2.4. Zjednodušené zobrazení životního cyklu service worker, převzato z [18]

- **Cache falling back to the network** - U všech požadavků, které mají k sobě již uloženou odpověď z dřívějšího volání, vrátíme uloženou odpověď. U těch, které ještě aplikace uložené nemá, si vyžádá odpověď ze serveru a tu poté uloží.
- **Network falling back to the cache** - Pokud je síť k dispozici tak všechny požadavky posíláme po síti a zároveň je ukládáme do cache. Pokud je uživatel odpojen od sítě, vrátí se poslední zachovaná odpověď ze serveru, uživatel tak uvidí poslední aktuální obsah, ke kterému měl přístup před ztrátou připojení.
- **Cache then network** - Tento přístup zároveň zašle požadavek na server pro aktuální informace a zároveň ukáže data z cache. Když dorazí nová data ze serveru, tak aktualizuje obsah stránek i cache na novou hodnotu.

Nezvažovali jsme strategie Cache only a Network only. Strategie Network only jak už název vypovídá vůbec nepoužívá ke svému fungování cache a nefunguje tedy bez připojení. Strategie Cache online zase není obecně uplatnitelná protože pokud budeme všechny dotazy brát pouze s cache, tak nemáme kde vzít odpověď na první dotaz.

2.3.4 Omezení

Jelikož se jedná o novou technologii, míra podpory těchto funkcionalit se různí napříč prohlížeči. Možnost nainstalovat PWA zatím podporují pouze prohlížeče založené na chromiu, tedy například chrome nebo microsoft edge. Push notifikace zase nejsou podporované na prohlížeči Safari používaném na systémech od firmy Apple. V této práci vnímám implementaci některých součástí PWA spíše jako prototyp, který není nezbytnou součástí k fungování aplikace.

2.4 Framework/Knihovna

Frameworky pro tvorbu webových aplikací zažívají v posledních letech velký rozmach a těší se velké popularitě mezi vývojáři. Nabízí se tedy otázka jak vybrat ten správný pro

úspěšný projekt. Jako první kritérium pro zúžení výběru posloužila popularita daných frameworků v současné době a sice skrz každoroční dotazník komunity stackoverflow, u nějž je typicky jedna z otázek zaměřená na rozšířenost frameworku. Data z dotazníku pro rok 2020 ukazují že nejpoužívanější „webové frameworky“ společně s procentem vývojářů, kteří je používají, jsou [21]:

1. **jQuery** s 43%
2. **React** s 36%
3. **Angular** s 26%

Odpovědi v tomto průzkumu jsou mírně zavádějící a potřebují drobnou interpretaci. Ač se totiž průzkum ptá na webové frameworky, tak zachycuje v odpovědích i knihovny. Ve skutečnosti jsou totiž jQuery a React knihovny (avšak s naprosto rozdílným účelem) a teprve Angular je framework, jelikož jako jediný přebírá kontrolu nad programem. Zatímco jQuery se zaměřuje na ulehčení práce vývojáře a zkrácení kódu po stránce fungování aplikací (jako například změna vzhledu prvku na stránce či zakomponování animace pomocí pár řádek kódu), tak React sebou přináší celé paradigma o tom, jak strukturovat kód do komponent (menších samostatných kusů kódu), předávat mezi komponentami data a překreslovat při změně pouze ty komponenty, u kterých je to potřeba. Strukturování kódu do menších celků je stěžejní vlastnost, která je potřeba pro psaní přehledného kódu. Tuto vlastnost sdílí React i Angular a proto budeme pro výběr rozhodovat mezi těmito dvěma technologiemi.

■ 2.4.1 React

React je „deklarativní, efektivní a flexibilní javascriptová knihovna pro vytváření uživatelských rozhraní“ [22] spravovaná firmou Facebook. Klíčovou součástí práce s reactem je členění kódu na komponenty, které by měly být samostatným a izolovaným kusem kódu. Nejedná se o plnohodnotný framework například kvůli tomu, že React samotný neobsahuje žádnou funkcionalitu na přecházení mezi jednotlivými stránkami, tedy nerozhoduje o průchodu aplikací. Když však spojíme React s dalšími balíčky jako například React-router a Redux, dostaneme již funkce, které bychom očekávali od frameworku a které má například Angular dostupné v základu. Hlavní rozdíly v používání obou technologií shrnu z pohledu vývojáře v krátkém bodovém přehledu:

- **Minimalismus** - React je minimalistický co se do complexity používání a funkcionalit týče.
 - Seznámení se s Reactem tedy programátorovi zabere méně času. Například pro zobrazování dat využívá react JSX, což je vlastně zjednodušeně možnost psaní javascriptového kódu přímo do HTML šablony. Angular umí dosáhnout stejných věcí, ale používá pro to vlastních direktiv, které se programátor musí naučit.
 - Programátor má možnost vybrat si jemu vyhovující řešení pro ostatní funkcionality, které v Reactu chybí, ale využijí se při vytváření webových aplikací. Angular zahrnuje některé dodatečné funkce napevno, takže programátor ztrácí možnost volby.
- **Oblíbenost** - Jak z průzkumu vyplývá, React je v současné době používanější než Angular [21] a používanější technologie zpravidla znamená více zdrojů k učení a více navazujících technologií.
- **Two-way binding** - React nedovoluje programátorovi narozdíl od Angularu přímý two-way data binding. Toto rozhodnutí je většinou vnímáno jako lepší rozhodnutí, neboť two-way data binding je spojen s výkonnostními problémy a ztěžuje debugování programu, neboť se dá hůře vysledovat změna stavu [23].

■ 2.4.2 Angular

Angular je sám o sobě plnohodnotný framework vydaný a spravovaný firmou Google. Důležité vlastnosti při používání angularu v kontrastu s reactem jsou:

- **Typescript** - Angular vyžaduje psaní kódu v typescriptu místo javascriptu. Typescript je nadmnožina javascriptu, která zavádí možnost přidat do kódu kontrolu typů, výsledný kód se však zkompiluje do javascriptu, takže typescript můžeme (za pomoci kompilování) používat všude tam kde javascript. Typescript obecně přidává na přehlednosti a dokumentaci kódu a umožňuje odhladit určité typy chyb před samotným během programu. V reactu je tato možnost také, ale Angular ji přímo vyžaduje, což můžeme vnímat jako praktiku podporující lépe čitelný a spravovatelný kód.
- **Kompletní řešení** - Angular obsahuje všechny funkcionality k vytvoření standardní webové aplikace již v základu, jako například routování. Programátor je sice tlačén do používání těchto součástí Angularu, na druhou stranu nemusí trávit čas vybíráním a instalací dodatečných balíčků, které bude téměř jistě potřebovat.
- **Dosah stylů** - Pokud přidáváme ke komponentě styly, tak Angular sám od sebe omezuje působení těchto stylů na danou komponentu. Pokud v Reactu přidáme ke komponentě soubor se styly a upravíme styl například pro tlačítko (jako selektor použijeme HTML tag) tak to má dosah na elementy v celé aplikaci. To může přinést nečekané styly na stránce, kde jsme s nimi nepočítali.

■ 2.4.3 Rozhodnutí

Pomocí tohoto srovnání si programátor může lépe uvědomit výhody jedné technologie oproti druhé a podniknout dodatečné kroky, které umožní vykompenzovat případné nevýhody. Ve své práci jsem se rozhodnul používat React kvůli jeho popularitě a menší učicí křivce (což je výhodné i z hlediska navazování na projekt někým jiným, neznalým frontendových technologií). Ač typescript není vyžadován povinně v Reactu, tak se dá do projektu snadno přidat. Chybějící defaultní omezení stylů na určité komponenty se dá zase kompenzovat pomocí delších názvů CSS tříd pro prvky (ve stylu „modul-popis“ tedy např. „profil-next-button“) a dále zmírnit zanořováním selektorů za pomoci preprocesorů.

Kapitola 3

Analýza

Průzkum v obchodu aplikací systému Android na téma odvykání kouření ukáže hned několik programů přesahujících milion stažení. Obvyklé funkce zahrnují přehled statistik, získané odznaky, náhodné tipy a například psaní vlastních poznámek nebo vytváření vlastních záznamů. Aplikace, kterou se zabývám v této práci (a která byla poskytnuta vedoucím práce) má jednu důležitou věc navíc, kterou se od ostatních odlišuje, a sice interaktivní terapie na denní bázi vytvořené odborníky na odvykání.

Může se zdát, že vytvářet webovou aplikaci podle aplikace mobilní je snazší než vytvářet novou aplikaci podle vlastních představ, avšak pokud chybí jakákoliv projektová dokumentace, tak tento argument ztrácí na síle. Naopak, bez dokumentace nemá programátor lepších nástrojů než opakovaně za sebou procházet aplikaci, obzvláště tehdy, kdy se možné akce větví do několika různých směrů, aby si byl jist, že na nic nezapomněl, nebo procházet kód existující aplikace, což je bez potřebné specifikace nebo komentářů leckdy ta horší varianta. Aplikace, která běží v produkci a slouží uživatelům většinou také během měsíců až let fungování posbírání nové funkce, takže obsahuje více požadavků, než běžná aplikace, která je vytvářena z nuly.

3.1 Sezení

Nejzajímavější funkcionalitou oproti konkurenčním aplikacím jsou pravděpodobně interaktivní sezení. Jednotlivá sezení se skládají ze stránek, mezi kterými uživatel přechází a vyplňuje v nich informace, ať už pomocí zadávání textu nebo výběru z možností. Systém informace ukládá a podle nich přidává uživateli získané odznaky a části uživatelského plánu. Plán se skládá z nejdůležitějších momentů posbíraných ze sezení, například proč chce uživatel skončit s kouřením a kdo je jeho oporou. Na základě vyplněných informací na každé stránce systém dokáže rozhodnout, kterou stránku uživateli ukáže příště. Jednotlivé stránky a pravidla mezi nimi vytváří příslušný odborník na adiktologii v administraci serverového řešení (úprava serverového řešení není součástí práce) pomocí skládání daných prvků (texty, podmínkové bloky, obrázky, vstupy, ...). Jednotlivé sezení jsou tedy uloženy jako grafy, kde stránky jsou uzly grafu a hrany jsou podmínky pro přechod z jedné stránky do druhé. Náhled do administrace sezení je zachycen na obrázku 3.1.

3.2 Funkční požadavky

■ Sezení

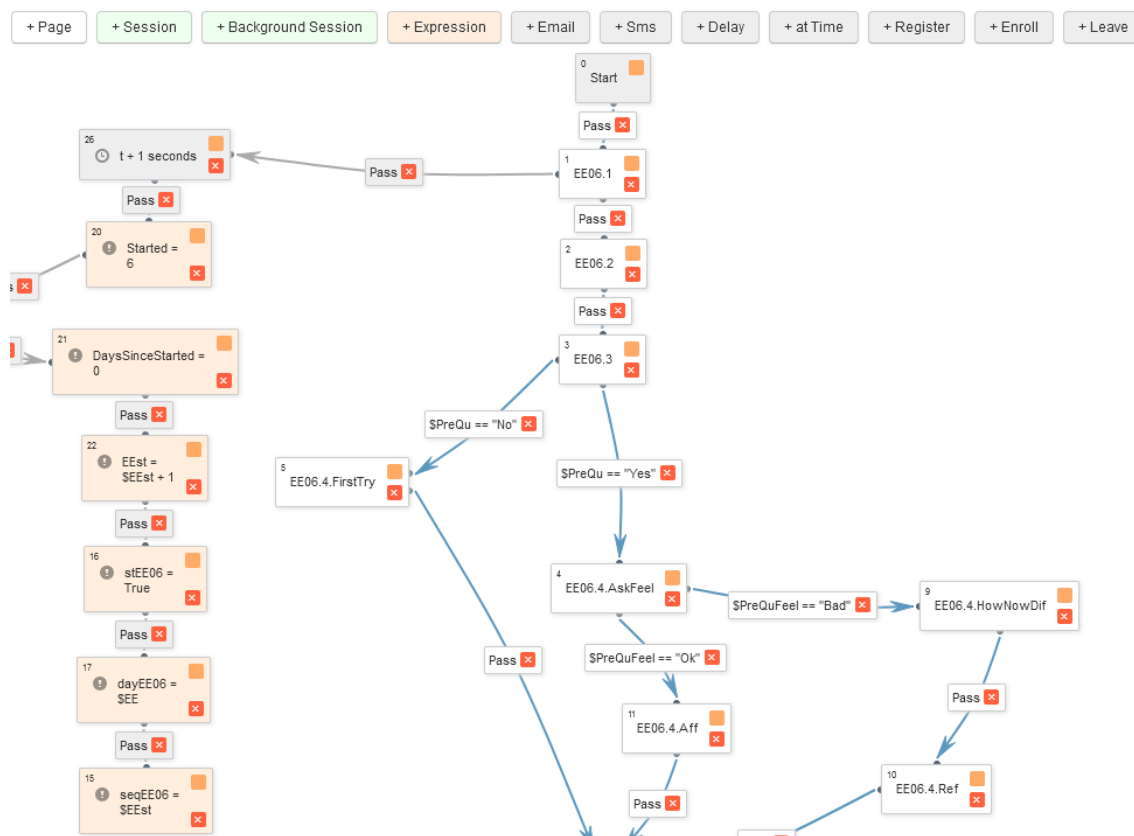
- **Notifikace ohledně dostupnosti** - Uživatel se dozví, zdali je pro něj připravené další sezení. Jednotlivá sezení od sebe mají povinný časový rozestup, který řídí server.
- **Procházení sezení** - Uživatel si může zobrazit jednotlivé stránky sezení, vyplňovat na nich potřebné informace (jednoduché vstupy a výběr z více možností), rozbalovat interaktivní prvky (toggle elementy) a přecházet na další stránky sezení.

- **Výzba k platbě** - V základním módu aplikace ukazuje prvních N sezení, poté ukáže uživateli informace k platbě (více v samostatném bodu).
- **Nově získaný odznak** - Systém ukáže po sezení uživateli získaný odznak.
- **Reakce na notifikace ohledně propadnutí cigaretám** - Ve fázi aktivního odvykání si uživatel může zobrazit speciální typ notifikací, kterými potvrzuje, jestli zvládá odvykání.
- **Ukázat obrazovky po dokončení terapie** - Po dokončení terapie se uživateli zobrazí několik obrazovek k završení odvykání.
- **Zobrazit souhrn informací** - Stránka, na které uživatel vidí souhrn základních funkcí, tedy informace o získaných odznacích, statistikách, svém plánu a zdali je pro něj připraveno další sezení, spolu s odkazem do jednotlivých sekcí.
- **Zobrazit a upravit svůj plán** - Uživateli se zobrazují spolu s krátkým popisem důležité body ze sezení doplněné o jeho vstup, které mu připomínají jeho plán k odvykání. Tyto vstupy může uživatel kdykoliv změnit.
- **Zobrazit statistiky** - Uživatel si může zobrazit statistiky ohledně ušetřených peněz, počtu absolvovaných sezení a jiné.
- **Zobrazit odznaky** - Uživatel průchodem sezení získává odznaky, které si může zobrazit. Tyto odznaky se dají rozkliknout a obsahují dodatečný motivační text. Uživatel zároveň může upravovat údaje, které v aplikaci zadal.
- **Debugovací mód** - Tester aplikace může zapnout debugovací mód.
 - **Přeskočení sezení** - Tester může přeskočit do bodu, kdy je dostupné další sezení, aby nemusel čekat vyhrazený čas, než server sám od sebe nabídne uživateli nové sezení.
 - **Restartování sezení** - Tester může restartovat sezení a projet celou aplikaci od začátku bez potřeby stále zakládat nové účty.
 - **Checkpointy** - Tester může vytvářet checkpointy, tedy jakýsi otisk aktuálního stavu uživatele. Uložené stavy pak může načíst, například pokud potřebuje otestovat stránky v pokročilé fázi odvykání několikrát za sebou, aby nemusel vše procházet znovu od prvního sezení.
- **Sekce Moje podpora**
 - **Zpězná vazba** - Uživatel může odeslat zpětnou vazbu vývojářům a specifikovat jestli se jedná o chybu.
 - **Tipy** - Uživatel si může zobrazit náhodný tip pro odolání pokušení a přepnout na další tip.
 - **O aplikaci** - Uživatel se může dozvědět více o aplikaci, společně se zobrazením informací o ochraně osobních údajích a obchodních podmínkách.
 - **Odkaz na blog** - Uživatel se může z této sekce dostat odkazem na blok s příspěvky ohledně odvykání.
 - **Poslední stránka sezení** - Tato možnost ukáže kuřákovi poslední stránku z naposled navštíveného sezení.
 - **Zavolat známému** - Uživatel může zavolat známému.
- **Sekce Můj účet**
 - **Zobrazit si sílu profilu** - Uživatel se dozvídá pomocí názorného ukazele sílu svého profilu, která odpovídá počtu vyplněných dotazníků.
 - **Vyplnit dotazníky** - Uživatel může vyplnit dotazníky ohledně svých zdravotních informací, předchozích pokusech a svém zázemí.
 - **Změnit oslovení** - Uživatel může změnit jméno, kterým jej aplikace oslovuje.

- **Registrace** - Uživatel se může registrovat do aplikace pomocí emailu a hesla. Systém u toho posbírá od uživatele dodatečné informace (oslovení, pohlaví, kolik cigaret vykouří) a zobrazí mu motivační texty.
- **Změna profilové fotky** - Uživatel si může navolit vlastní profilovou fotku.
- **Přihlášení**
- **Odhlášení**
- **Obnovení hesla**
- **Smazání účtu**

■ Platby

- **Argumenty platby** - Uživatel si po kliknutí na tlačítko může zobrazit sérii obrazovek s důvody proč přestat kouřit/proč si pořídit plnou verzi aplikace a případně vybrat důvod proč si nepřeje pokračovat.
- **Slevový kód** - Uživatel může uplatnit slevový kód na pořízení aplikace, po kterém se sníží cena (případně až na průchod aplikací zdarma).
- **Platba** - Uživatel se může prokliknout k platební bráně, přes kterou si odemkne přístup ke všem terapiím.
- **Zobrazit časově omezenou slevu** - V aplikaci se uživateli zobrazí informace o časově omezené slevě z ceny.

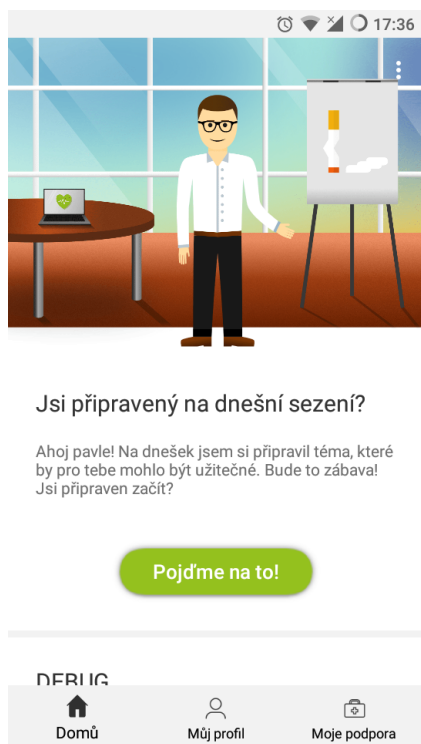


Obrázek 3.1. Ukázka sezení v administraci serverového řešení

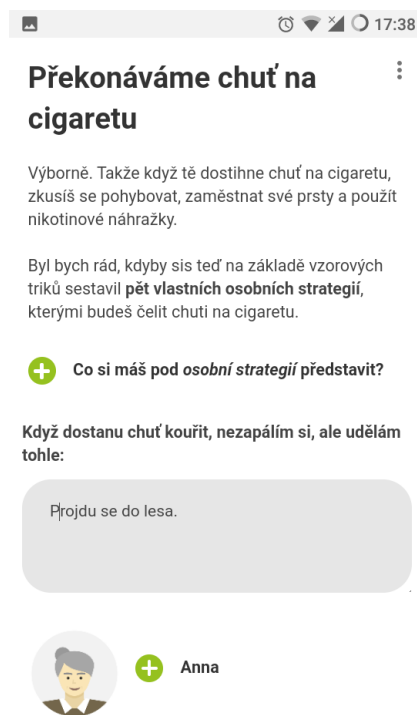
3.3 Obrazovky

Pro ilustraci uvádím také ukázky obrazovek z aplikace. Jelikož budou zajímavé změny rozhraní vyzdvihnuty dále v kapitole o návrhu webové aplikace, tak zde uvádím jenom

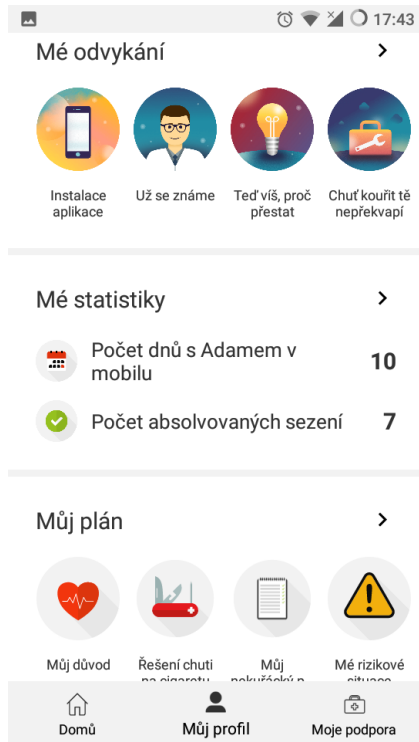
ty nejdůležitější. Aplikace oplývá ještě několika dalšími obrazovkami pro dotazníky, argumenty pro platbu, závěrečné povzbuzení po dokončení sezení, přihlášení a registraci.



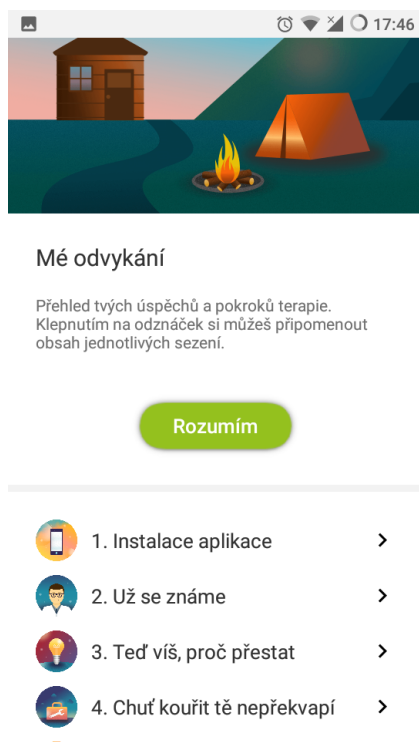
Obrázek 3.2. Úvod do sezení v mobilní aplikaci



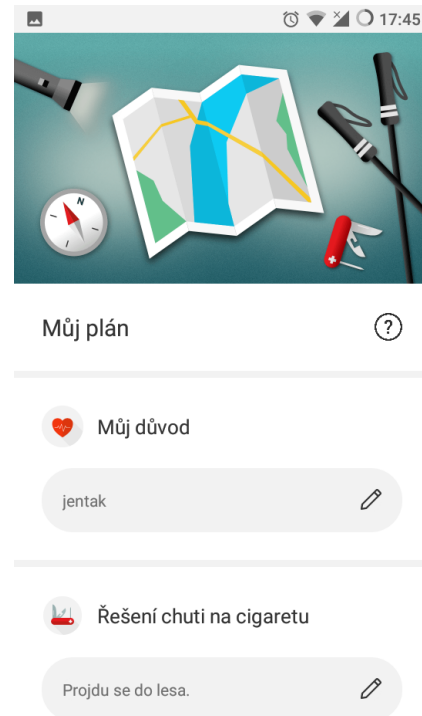
Obrázek 3.3. Jedna strana náhodného sezení v mobilní aplikaci



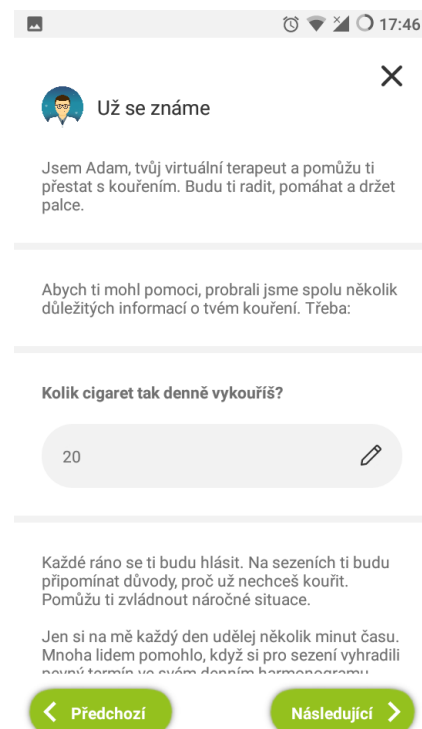
Obrázek 3.4. Hlavní stránka (profil), přístupná po přihlášení do mobilní aplikace



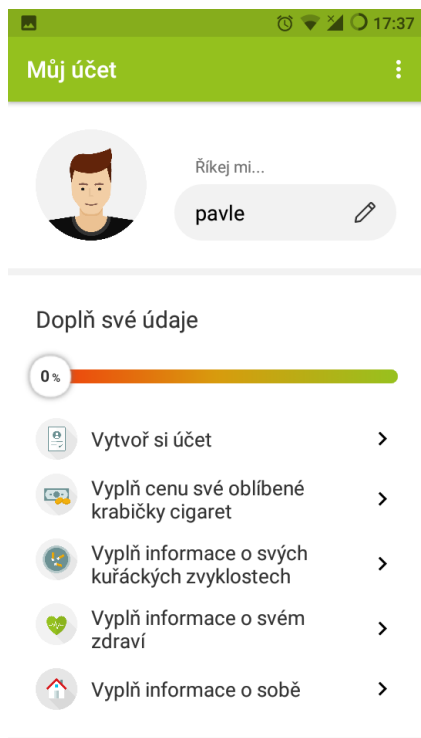
Obrázek 3.6. Přehled získaných odznaků v mobilní aplikaci



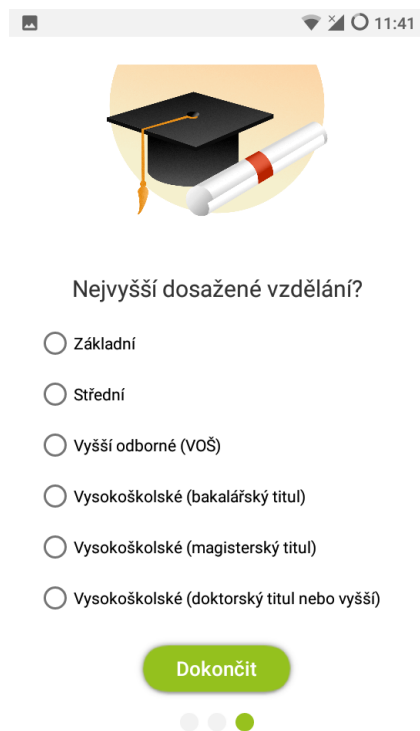
Obrázek 3.5. Plán, který si kuřák vytváří procházením sezení v mobilní aplikaci



Obrázek 3.7. Detail získaného odznaku v mobilní aplikaci



Obrázek 3.8. Možnosti v nastavení účtu, mezi nimi změna oslovení a vyplnění dotazníků.



Obrázek 3.9. Jedna stránka z dotazníku ohledně osobních informací.

Kapitola 4

Návrh a Implementace

Účelem této práce je vytvořit webovou aplikaci, která bude fungovat stejně jako existující aplikace mobilní a bude napojená na data ze stejného, již existujícího, backendového řešení. Není cílem této práce vytvořit jednu aplikaci, která by fungovala na všech platformách a nahradila tak existující mobilní aplikaci. Zároveň ani není cílem dělat změny v serverovém řešení, aby jednak nedošlo k ovlivnění funkce mobilní aplikace a jednak proto, že na serverovém řešení se pracuje a probíhá jeho přepis ze starší verze do nové. Zaběhnutá mobilní aplikace poskytnutá vedoucím práce již funguje ke svému účelu a je využívána tisíci uživateli na platformách iOS a android. Jak již bylo zmíněno v 2.3.4, mobilní aplikace mají výhody například v instalovatelnosti a push notifikacích, které zatím nejsou ve webových aplikacích podporované všemi potřebnými platformami (zejména zařízeními od Applu), zatím by tedy jednotná webová aplikace nedokázala poskytnout všem uživatelům mobilní aplikace funkcionality, na které jsou zvyklí. Přesto je ale dostatečné množství uživatelů (nejenom ze starších generací), kteří upřednostňují v digitálním světě používání počítačů oproti mobilům, viz 2.2.3. Účelem je tedy vytvoření webové aplikace, která bude sloužit jako alternativa uživatelům preferujícím počítače, neboť mobilní aplikaci nemají možnost nebo nechtějí používat.

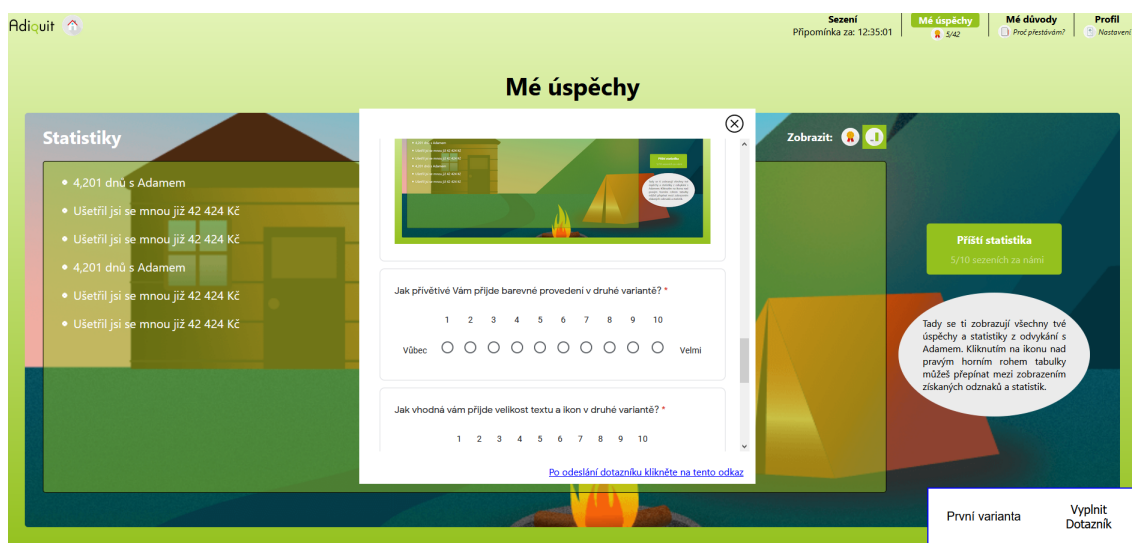
4.1 Návrh

Hlavním principem při návrhu webové aplikace bylo využití sloupového rozložení prvků, aby aplikace mohla splňovat potřebu responzivity designu. Bylo také potřeba pochopit návrhové vzory přítomné v mobilní aplikaci, aby byly zachovány i ve webové verzi. Aplikace využívá principů gamifikace aby odlákala kuřákovu pozornost od jeho závislosti a to například pomocí odemykatelných odznaků, dostupnosti nových funkcionalit při vyplnění dotazníků a členění uživatelských dat do přehledných statistik, což jsou známé vzory při návrhu uživatelských rozhraní [24]. Kromě toho ale byl samozřejmě prostor pro přicházení s inovacemi. Při vytváření vzhledu aplikace jsem mířil své návrhy k agilnímu přístupu, a tedy využití testování a evaluace už během samotného návrhu. Většina základních stránek byla vytvořena ve dvou variantách, aby následně mohlo být použito preferenčního testování, kdy uživatel může v krátké ukázce rozhodnout, kterou ze dvou variant návrhu upřednostňuje [25]. Design byl vytvořen pomocí programu Adobe XD. Použité ikonky byly poskytnuty vedoucím a pochází z mobilní aplikace.

4.1.1 První verze

Na začátku procesu návrhu jsem vytvořil většinu obrazovek ve dvou variantách s cílem, aby uživatelé mohli rozhodnout, která je pro ně přívětivější. Myšlenka byla jednoduchá, ukázat dvě varianty též stránek s jednoduchými přechody a zeptat se uživatele na jeho subjektivní preference. K testování jsem chtěl využít některou z mnoha UX testovacích platform, ale ukázalo se, že mají velmi citelná omezení ve verzi zdarma. Nejčastějšími omezeními byla časová omezení (dotazník zdarma může trvat pouze 1 minutu), omezení počtu uživatelů (maximálně 5 uživatelů) případně nebyla verze zdarma vůbec nabízena.

Rozhodl jsem se tedy pro větší komfort uživatelů vytvořit vlastní jednoduché rozhraní pro otestování prototypu, které představovalo jednoduchou grafickou nadstavbu ve spojení s Google forms. Nad nakódovaným prototypem byla vrstva rozhraní pro přepínání vrstev a zobrazení dotazníku. Na obrázku 4.1 Je vidět ono rozhraní a zároveň raná verze stránky s úspěchy, která se nedostala do finální verze. Z 3 hlavních obrazovek, u které měla každá 2 varianty, byla za pomoci výsledků uživatelských preferencí a konzultace s vedoucím vybrána výsledná verze, která bude ukázána v následující kapitole. Kromě čistých preferencí byly uživatelé tázáni ještě na drobnosti ohledně velikosti písma, barevné kombinace. Tyto odpovědi byly reflektovány, ale nezmiňuji je zde, neboť padly podobné dotazy ještě u finálního testování, které je rozebráno důkladněji



Obrázek 4.1. Ukázka jednoduchého rozhraní pro průzkum preferencí mezi dvěma variantami stejné obrazovky.

4.1.2 Implementovaná verze

Inovace v konečném návrhu se projevily například v procesu registrace, hlavní obrazovce, vynahrazení mobilních gest, zobrazení sezení a spojení sekcí. Mezi inovacemi bych vyzdvihнул zejména proces registrace. V telefonu se jedná o sérii obrazovek skládajících se z obrázků a textů. Ve webové aplikaci jsem chtěl naplno využít dostupného místa a zároveň udržet uživatelskou pozornost. Výsledkem bylo vyplňování informací, které je uživateli prezentováno pomocí systému připomínajícího konverzaci (viditelné na obrázku 4.2), jelikož návrh uživatelského rozhraní na bázi konverzace zjednodušuje uživateli jeho používání [26]. Důležité změny grafické podoby se dostalo také sezení, které záměrně nevyužívá celou plochu a záměrně má světlejší pozadí, aby obsah sezení vystupovalo z celé stránky a lépe zaujalo kuřákovu pozornost, jak je vidět na 4.3.

Mezi další menší inovace kromě využití responzivního designu pomocí sloupců patří například vyřešení chybějící možnosti ovládání pomocí gest (například k přecházení mezi obrazovkami dotazníku) tím, že jsem k vyskakujícím oknům odznaků přidal šipky po stranách. Dále jsem spojil kategorie odznaků a statistik dohromady, neboť jsou spolu velmi úzce významově spjaté. V neposlední řadě jsem přistoupil trochu jinak k domácí obrazovce, ve které se nesnažím o duplikaci obsahu jako u mobilní aplikace, ale ukazuji uživateli z každé kategorie pouze poslední získaný nebo náhodný střípek ze sezení (odznaky, součást plánu, statistiky).

AdiQuit

Můžeš mi o sobě něco říct?

Aby nám to spolu fungovalo, měl bych se o tobě leccos dozvědět. Pověz mi prosím něco o sobě.

Jsem: ☐ žena ☐ muž

Hynku, Viléme, Jarmilo? Pomoz mi se skloňováním a použij prosím 5. pád.

Říkej mi:

Abych tě příště poznal, budu potřebovat ještě tvůj email a heslo

Email: Heslo:

Kolik cigaret denně vykouříš, Pavle?

0 40

Kolik stojí krabička? Řekni mi kolik cigaret máš v krabičce a kolik za ni většinou platíš.

Obrázek 4.2. Obrazovka pro založení nového účtu, sbírání jednotlivých informací má připomínat chatování s uživatelem

AdiQuit

Sezení ☐ Spolu to zvládneme

Mé úspěchy ☐ Co jsem zvládl

Můj plán ☐ Proč přestávám?

Moje podpora ☐ Okamžitá pomoc

Můj účet ☐ Nastavení

Překonáváme chuť na cigaretu

Výborně. Takže když tě dostihne chuť na cigaretu, zkusíš se pohybovat a zaměstnat své prsty.

Byl bych rád, kdyby sis teď na základě vzorových triků sestavil pět vlastních osobních strategií, kterými budeš čelit chuti na cigaretu.

Co si máš pod osobní strategií představit?

Pokud ti je třeba blízké udělat něco příjemného, napiš, co přesně by to mělo být. Například sednout si a přečíst kus rozečtené knihy. Jestliže ti dává smysl trocha pohybu, napiš, že si pustíš hudbu a zavzpomínáš na tanečky ze studentských let. Zkrátka, ušij si své triky na tělo!

Když dostanu chuť kouřit, nezapálím si, ale udělám tohle:

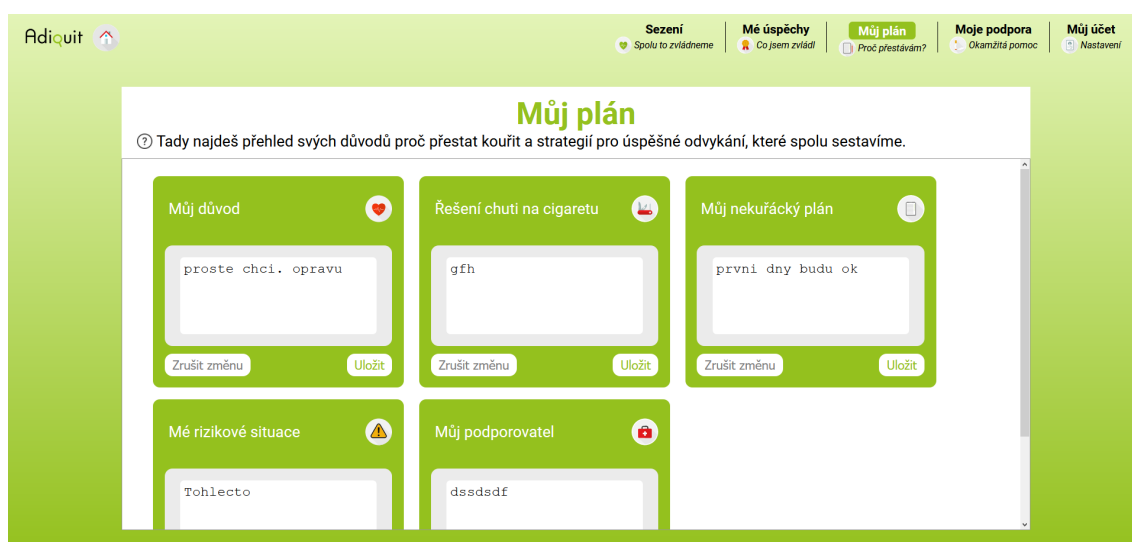
Anna

1. Udělám si čaj.
2. Vezmu si do ruky tužku.
3. Pustím si televizi.

Obrázek 4.3. Obrazovka reprezentující příklad jedné stránky sezení



Obrázek 4.4. Úvodní obrazovka webové aplikace, kombinující přehled a odkazy do ostatních částí



Obrázek 4.5. Obrazovka Můj plán, pokrývající důležité body ze sezení, ze kterých si kuřák tvoří plán



Obrázek 4.6. Obrazovka Podpora, ve které kuřák najde tipy a dodatečné informace k odvykání



Obrázek 4.7. Obrazovka Úspěchy, která kombinuje statistiky a získané odznaky



Obrázek 4.8. Obrazovka Můj účet, s dotazníky k vyplnění a základními možnostmi ke správě účtu



Obrázek 4.9. Obrazovka zachycující detail rozkliknutého odznaku.

4.2 Rozšiřitelnost a udržitelnost

Má snaha v implementaci této aplikace a výběru technologií mířila maximálně k rozšiřitelnosti a dokumentaci projektu. Tohoto cíle jsem se snažil dosáhnout pomocí tří hlavních pilířů:

- **Použití typescriptu místo javascriptu** - Lépe vysvětleno v podkapitole 4.3.1.
- **Vyhýbání se instalací knihoven, které nejsou přímo potřeba** - Nadměra používání externích knihoven vede snadno k tomu, že když převeze po nějakém čase dosavadní systém nový programátor s úlohou rozšíření systému, tak se dozví, že projekt nejde ani spustit, neboť jedna z mnoha knihoven ztratila podporu a není již kompatibilní s ostatními knihovnami.

- **Dokumentace existujícího rozhraní (API) i nově vytvořeného** - Existující rozhraní bude popsáno v příloze, samotná aplikace je popsána pomocí této bakalářské práce a komentářů v kódu.

4.3 Technologie

4.3.1 Typescript vs javascript

Prvním problémem v udržitelnosti a debugování webových aplikací je dynamičnost typování javascriptu, dominantního jazyka webových prohlížečů. Výběr jazyka podporujícího statické typování má hned dvě velké výhody:

- **Odchycení chyb** - Kompilátor může odchytit chyby, které by se v dynamicky typovaných jazycích projevy až v tu chvíli, kdy by se zrovna dotýčný kód prováděl. Takto se chyby mohou odchytit už při překladu, což má pozitivní dopad na opravování chyb spojených s typy [27].
- **Ulehčení čtení kódu** - Typy ulehčují čtení programu, obzvláště pro ty, kdo se seznamují s kódem, který vytvořil někdo jiný. Psaní v staticky typovaném jazyku tedy snižuje čas, který potřebuje nově příchozí programátor na seznámení s kódem, například pokud se jej chystá rozšířit.

Tyto výhody jdou na úkor mírného zvýšení časové zátěže při psaní kódu a při vytváření nového projektu. Občas se může stát, že nastane potřeba při snaze importovat existující knihovnu v javascriptu, u které chybí definice typů, nebo že programátor nezná přesnou podobu odpovědi API tak, aby mohl jasně říct jaké datové typy bude odpověď mít (například když byl backend napsán v dynamicky typovaném jazyku, který dovoluje měnit typy jednotlivých parametrů). Typescript má navíc tu výhodu, že je nadmnožinou Javascriptu, jakýkoliv kód v javascriptu může být použit i v typescriptu, buď pomocí nastavení kompilátoru nebo pomocí klíčového slova `any`. Používání tohoto klíčového slova není v důsledku žádanou praktikou, protože obírá programátora o výše popsané výhody, avšak taková možnost alespoň existuje. Tak jako platí, že i nedostatečná dokumentace je lepší než žádná, tak platí, že použití kontroly typů alespoň u části systému a vynechání tam, kde by typy byly příliš svazující (např. externí knihovny) je daleko lepší než nepoužívání typů vůbec.

4.3.2 Ostatní použité technologie a knihovny

- **Sass** - Jedná se o nadstavbu (zjednodušeně řečeno) nad samotným CSS, která přidává funkce pro zpřehlednění kódu a méně opakování. Mezi zlepšeními nad CSS je vhodné zmínit například možnost definovat konstanty (vhodné například pro barvy), znovu používat kód jiných tříd (podobné dědičnosti) a také zanořovat do sebe jednotlivé selektory.

```
.left-arrow {
  @extend .arrow; // zde se doplní definice CSS třídy .arrow
  transform: rotate(-90deg);
}
```

- **Create-react-app** - Jedná se o nástroj, který pomocí jednoho příkazu založí nový projekt s knihovnou React a nakonfiguruje případné použití sass a typescriptu, takže uživateli stačí pouze spustit projekt a poté případně doinstalovat balíčky, které potřebuje navíc.

- ## 4.4 Prozkoumání a dokumentace API

Nastává tedy otázka, jak přistupovat k implementaci projektu bez dokumentace. V průběhu práce byl poskytnut vedoucím práce kód mobilní aplikace i serverové části. Vcelku se tak ale jedná o tři různé technologie, tedy webová aplikace napsaná v Reactu, mobilní aplikace napsaná v Jave a serverová část napsaná v Pythonu, ve frameworku Django. Bez zkušeností s danými frameworky zabere nastudování a použití i pouze jedné platformy dost času. Bylo tedy třeba zvolit vhodný nástroj, který by ulehčil trochu práci, nástroj k analýze dostupných funkcionalit serveru a následnému popsání, a tím je Swagger, který dokáže pouze s minimální konfigurací poskytnout souhrn dostupných endpointů [29]. Bohužel z použití pythonu na backendu pramení další jisté nevýhody. Zatímco Django dovoluje rychlý vývoj a přináší sebou v základu velmi užitečnou administraci, dynamické typování komplikuje rozpoznávání parametrů požadavků a odpovědí ze serveru. Po tomto kroku jsem měl sice k dispozici seznam funkcionalit, ale nevěděl zatím jak je volat. Zbytek snažení zahrnoval po kouskách rozpoznávání kódu jak na straně serveru tak na straně aplikace, doplněnou o vzácné konzultace s původními

tvůrci. Dokumentaci API, složenou z vygenerovaných endpointů pomocí nástroje Swagger a následného ručního doplnění parametrů požadavků a následných odpovědí, jsem přiložil k této práci jako přílohu.

4.5 Zobrazování sezení

Z dříve zmíněného API a jeho použití aplikací je nejpřekvapivější sekce ohledně sezení. Místo aby android a iOS aplikace skládaly sezení z informací získaných z API, tak se pouze použilo existující zobrazení na serveru, kdy aplikace funguje pomocí technologie WebView, což znamená že pouze otvírá stránky z internetu v renderovacím jádře prohlížeče. Toto přineslo jednu hlavní výhodu tvůrcům, a sice že nemusí pro různé systémy vytvářet různá zobrazení ale pouze použijí hotové zobrazení pomocí prohlížeče. Mobilní aplikace tedy nevykresluje žádná sezení ani se nestará o jejich přepínání, pouze zjistí, jestli je sezení dostupné a poté otevře prohlížeč. Tento přístup by se dal uplatnit i u webové aplikace jako prototyp řešení, má však mnoho nevýhod, které popíši v následující podkapitole.

4.5.1 Nevýhody prototypu

Paralelou s řešením sezení po stylu mobilní aplikace by bylo vložit do stránky element představující vestavěný prohlížeč, který by pouze ukazoval zobrazení na serveru. Ušetřil by se tím sice čas, jelikož sezení je možné tímto způsobem ovládat, jak dokazuje mobilní aplikace, ale tento přístup má vážné vady:

- **Větší zátěž** - Interakce se stránkami na straně serveru představují větší zátěž pro server. V případě, že by server poslal pouze informace o jednotlivých prvcích, ze kterých se dá sezení poskládat, nemusí na klienta vůbec putovat styly a chování stránky.
- **Chybějící responzivita** - Server vykresluje sezení jedním způsobem, které je napasováno na ideální zobrazení v mobilní aplikaci. Zobrazením stránky uvnitř stránky v prohlížečovém jádru se programátor vzdává možnosti stylovat prvky na straně klienta a učinit je responzivními. Obrazovky, které vypadají dobře na mobilu, pravděpodobně nebudou beze změny vypadat dobře na 32 palcovém monitoru.
- **Přílišné provázání** - K tomu, abychom mohli upravit zobrazení klientovi najednou musíme měnit celý server. Jakmile změníme zobrazení na serveru, tak se automaticky změní na všech platformách, tedy Androidu, iOS i webu. Takové řešení svazuje programátorovi ruce pokud chce na jedné z platform udělat drobné změny a zároveň ztěžuje případné opravování chyb a diagnostiku aplikace ze strany webu.

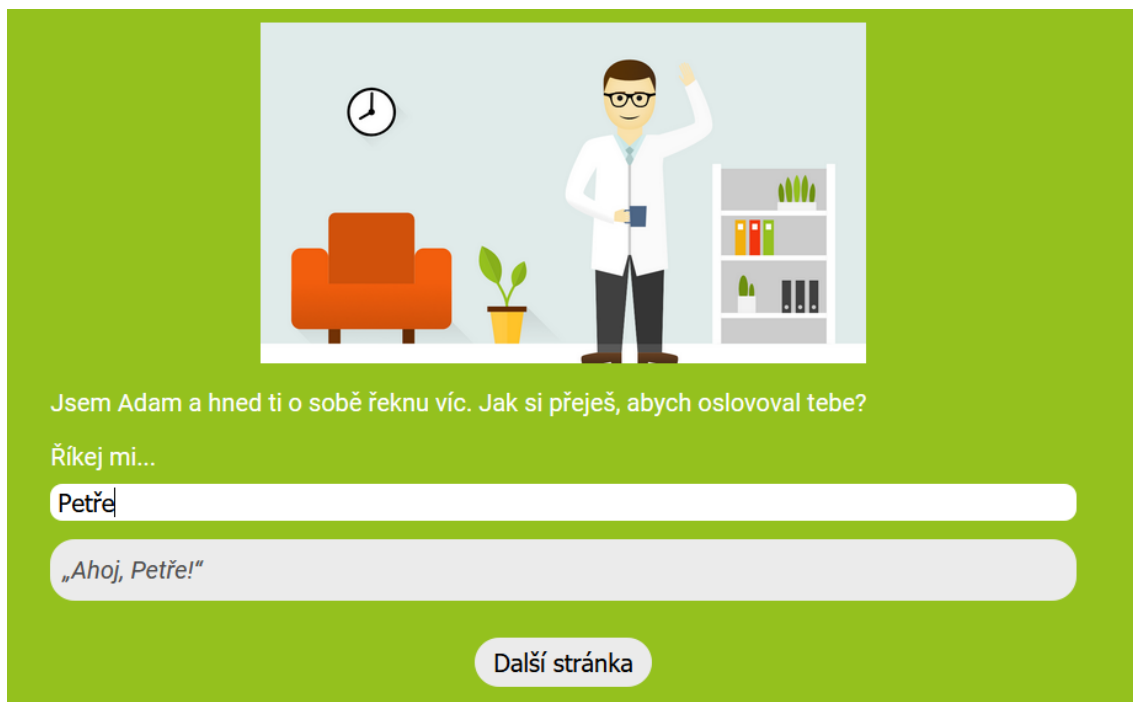
Na základě těchto nevýhod jsem se rozhodl hledat jiné řešení. Zjistil jsem, že server nabízí ještě možnost zaslat stromový výpis elementů, ze kterých se aktuální sezení skládá.

4.5.2 Zjednodušené zadání vykreslování sezení

Jak bylo vidět již při zkoumání mobilní aplikace a serverové části, jednotlivá sezení se skládají ze základních prvků, které sestavuje odborník na odvykání v administraci. Sezení prezentují uživateli informaci v textové i obrazové podobě a sbírají od něj vstup pomocí různých formulářů. Na základě odpovědí v jednotlivých sezeních server rozhodne, kterou stránku a později které sezení má být uživateli předneseno příště. Jednotlivými prvky jsou:

- Textový element, který může obsahovat značky pro doplnění vstupů ze současného nebo minulých sezení
- Tlačítko
- Formulář, skládající se z libovolného počtu těchto vstupů:
 - Jednoduchý vstup (číslo/jednořádkový vstup/více řádkový vstup)
 - Výběr jedné možnosti z mnoha
 - Výběr 0-N možností z mnoha
- Obsah podmíněný vyhodnocením výrazů, který se musí odeslat na server
- Toggle element, který kombinuje nadpis a text, jenž je skrytý/viditelný na základě klikání na prvek
- Kombinaci toggle elementů a multivýběru (uživatel vybírá z nabídky a zároveň se mu při označení možnosti objeví nový text)
- Obrázky

Abych mohl lépe ilustrovat problém, ukážu jak by mohlo vypadat sezení ve **zjednodušené podobě** na 4.10. Je důležité nepřehlédnout, že v v textu se mohou nacházet značky v podobě názvu proměnných pro informaci získaných ze vstupů (ze současného nebo předešlých sezení), jejichž data je potřeba spojit s případnými výskyty v textu a dynamicky je updatovat na základě toho, co uživatel zadává, tedy nejde jenom o prezentaci statických dat. Žádoucí výsledná podoba příkladu sezení ve formátu JSON je na obrázku 4.11.



Obrázek 4.11. Ukázka vykreslení jednoduchého sezení z formátu JSON

■ 4.5.3 Potřeba předzpracování dat

Formát sezení z API obsahuje (pouze tam, kde se očekává dynamická interpolace proměnných) v datech značky z frameworku AngularJS, který používá server k vykreslování odpovědi. Zde by se dalo namítnout, že bych tedy mohl použít k vykreslování rovnou AngularJS. Tento framework již ale není rozvíjen a je obecně doporučováno přejít na

```

"data": [
  {
    "content_type": "image",
    "content": {
      "url": "/media/image1.png",
      "class": "header-no-border"
    }
  },
  {
    "content_type": "conditionalset",
    "content": [
      {
        "expression": "$current_time < $AfternoonStart",
        "content": "<h1>Ahoj, rád tě poznávám</h1>\n",
      },
      {
        "expression": "$current_time >= $EveningStart",
        "content": "<h1>Dobrý večer</h2>",
      }
    ],
  },
  {
    "content_type": "form",
    "content": [
      {
        "field_type": "string",
        "required": true,
        "label": "<p>Říkej mi...:</p>\n",
        "value": "",
        "variable_name": "Name5p",
      }
    ],
  },
  {
    "content": "<p>„Ahoj, {{ Name5p }}"</p>\n",
    "content_type": "text",
    "variables": {
      "Name5p": "",
    }
  },
  {
    "content_type": "button text",
    "content": "",
  }
]

```

Obrázek 4.10. Zjednodušená ukázka sezení ve formátu JSON jak opole prvků

novou verzi, která by se dala považovat za úplně nový framework. Navíc, jedním z bodů této práce bylo nezasahovat do existujícího serverového kódu. Značky jsou našťestí pouze v místech, kde se očekává dynamická interpolace proměnných. V první řadě by kvůli čistotě kódu a udržitelnosti projektu odpověď ze serveru na dotaz k datům ze

sezení neměla vůbec obsahovat značky doplněné frameworkem. Zvolil jsem proto řešení pomocí Adapter, což je „návrhový vzor, který dovoluje spolupráci objektům s nekompatibilními rozhraními“ [30]. Díky tomu odpadá potřeba měnit stávající kód, pouze zařadím Adapter, který zpracuje chaotická data ze serveru, a další interpretace dat se bude zakládat na jejich ideální podobě. V případě že v budoucnu dojde k přepsání původního systému do ideální podoby, tak lze pouze smazat patřičný adaptér, a systém bude fungovat s novým API. Adaptér tak vyřeší syntaktické problémy s přijatou odpovědí, a sice naparsuje pomocí regulárních výrazů případné texty tak, aby z nich vyřadil znaky angularu.

■ 4.5.4 Další překážky

Kromě této úlohy ještě adaptér zastává funkci vyřazení nadbytečných prvků nebo naopak opravení chybějících. Nezřídka se stává, že součástí sezení přijdou platné prvky ale s prázdným obsahem, přemírá tlačítek k dokončení sezení nebo naopak nepřijde žádné tlačítko k ukončení sezení. Tyto chyby jsou způsobeny lidským faktorem při tvorbě sezení v administraci. Adapter zároveň vyřeší všechny elementy s chybějícím obsahem.

Odpověď ze sezení přidělová programátorovi práci nejenom po stránce syntaxe ale i sémantiky. Některá data by se mohla vyhodnotit rovnou na serveru a přesto předávají tuto úlohu na klienta. Jsou tím například interpolace dat, které se během sezení nemohou měnit a podmínkové bloky. Server pošle v podmínkovém bloku například 5 výrazů, u kterých se musí vyhodnotit podmínka, aby se zjistilo, který z nich se má zobrazit. Podmínku ovšem dokáže vyhodnotit pouze server.

■ 4.5.5 Architektura řešení

Konečnou architekturu pro vystavění sezení z dat z API jsem postavil pomocí návrhových vzorů Adapter, Builder a Simple Factory. Na první pohled se tato kombinace může zdát velmi zvláštní, obzvláště spojení vzoru Builder a Simple Factory. Chtěl jsem však oddělit syntaktickou opravu dat (o tu se stará Adapter) a následně sémantickou opravu dat, jelikož se musí ještě rozhodnout, které elementy je potřeba do sezení vykreslit a to platí především pro podmínkové bloky, u kterých je třeba poslat v dotazu podmínku na server a podle odpovědi určit, jestli se má daný blok vykreslit. Jedná se tedy o paralelu s čtením konfigurace dat. Builder je bod, který významově rozumí celému sezení a tedy určí, které bloky sezení se budou vykreslovat a jeho výstupem je pole prvků, ze kterých se skládá sezení. Postaví tedy z dané konfigurace validní sezení, což je pole prvků. Pole prvků následně vykreslí pomocí Simple factory. Simple factory nerozumí celému sezení, ale pouze jednotlivým elementům, takže dostane na vstupu element a převede ho jednoduchým způsobem na jeho prezentaci. Cítil jsem potřebu oddělit logiku rozhodování o vykreslování prvků od samotného vytvoření grafické reprezentace existujícího prvku.

■ 4.6 Tvorba detailu odznaků a dotazníků

Jedním z dalších důležitých bodů implementace bylo zobrazování dotazníků a odznaků. V aplikaci se nachází velké množství vyskakovacích oken na stejný způsob - zobrazit posloupnost textů, obrázků a vstupů s validací. Například jenom odznaky obsahují takových obrazovek 17. Kdyby měl programátor všechny obrazovky vytvářet jako samostatné soubory, tak by to vedlo k opakováním velké části logiky a zároveň nepřehlednosti z velkého množství dat. Místo toho jsem se snažil popsat tyto opakující se části pomocí konfigurace, která se následně postaví ve fungující odznak/dotazník. Odznak je reprezentován pomocí rozhraní, které má atribut popisující vyskakující obrazovku (detail

odznaku). Ta je tvořena polem prvků, které jsem dopředu definoval. Jednotlivé prvky jsou například texty, obrázky, vstupy a podobně. Při případném přidání nového odznaku pouze přijde programátor, podívá se na typ odznaku, zjistí že jeho detail je pole skládající se z typů například text/vstup/obrázek a napíše konfiguraci nového odznaku. Typescript byl v tomto snažení velkou pomocí, jelikož programátorovi, který ke kódu přijde, stačí jeden pohled, jak vytvořit nový blok. Bez typescriptu by se nedal systém vyjádřit tak jednoduše a elegantně.

```
{
  title: "Už se známe",
  subtitle: "Seznámi{{VerbPast}} ses s Adamem. Nyní n:",
  userVariableName: "EEcp",
  showPredicate: (val: any) => val >= 1,
  imageUrl: "adiquit_badge_ib.png",
  detailBlocks: [
    {
      type: IBadgeDetailBlockType.TEXT_BLOCK,
      text: "<p>Jsem Adam, tvůj virtuální terapeut a ;",
    } as IBadgeDetailTextBlock,
    {
      type: IBadgeDetailBlockType.TEXT_BLOCK,
      text: "<p>Abych ti mohl pomoci, probrali jsme s;",
    } as IBadgeDetailTextBlock,
    {
      type: IBadgeDetailBlockType.INPUT_NUMBER,
      text: "<h3>Kolik cigaret tak denně vykouříš?</h3>",
      varName: "NoCigi"
    } as IBadgeDetailInputBlock,
    {
      type: IBadgeDetailBlockType.TEXT_BLOCK,
      text: "<p>Každé ráno se ti budu hlásit. Na sezer",
    } as IBadgeDetailTextBlock
  ],
  order: 2,
},
```

Obrázek 4.12. Deklarativní zadávání nových odznaků v kódu

```
export enum IBadgeDetailBlockType {
  TEXT_BLOCK,
  INPUT_TEXT,
  INPUT_TEXT_FALLBACK,
  INPUT_NUMBER,
  IMAGE_BLOCK,
}

export interface IBadgeDetailTextBlock {
  type: IBadgeDetailBlockType.TEXT_BLOCK;
  text: string;
}

export interface IBadgeDetailInputBlock {
  type: IBadgeDetailBlockType.INPUT_TEXT |
  IBadgeDetailBlockType.INPUT_NUMBER;
  text: string;
  varName: string;
  disabled?: boolean;
}
```

Obrázek 4.13. Definice detailu odznaků pomocí typů

4.7 Implementace prvků PWA

- **Responzivita** - Responzivita je zajištěna především díky sloupcovému rozvržení většiny obrazovek. Aplikace je responzivní od určité míry, a sice od nezvyklé šířky okna 850 pixelů. Tento dolní limit jsem zvolil tak, že pokrývá šířky obrazovek menších laptopů a zároveň je dostatečná pro polovinu délky vůbec nejčastější varianty obrazovek (1920x1080), a sice pro lidi, kteří jsou zvyklí mít standardně dvě okna prohlížečů vedle sebe [31]. U zařízeních s menší šířkou okna se jedná už o tablety a telefony, pro které je určena existující mobilní aplikace. Mobilní aplikace již běží, má zaběhnuté uživatele a umožňuje zatím použití push notifikací pro širší publikum. Z business hlediska dává tedy větší smysl přesměrovat uživatele na mobilech, kteří chtějí odvykat od kouření, na mobilní aplikaci. S vědomím tohoto cíle tedy webová aplikace zobrazí při detekci menšího zařízení uživateli upozornění na existující mobilní aplikaci a odkaz, na kterém se dozví více.
- **Instalace** - Instalace byla naimplementována jako součást průzkumu konceptu PWA. Zhodnocení bude rozebráno v testování.
- **Podobná nativní aplikaci** - Pomocí některých vlastností (instalovatelnost) se webová aplikace přibližuje nativní aplikaci, ale jelikož nebude používána na telefonech, tak tento požadavek není důležitý.
- **Offline fungování, Aktuální** - Tyto dvě vlastnosti spolu souvisí, neboť jednou ze strategií pro lepší offline zážitek je ukazovat uživateli pouze data, která jsou v uložišti. Pro implementaci v této práci byla zvolena možnost ukazovat přednostně aktuální data pokud je dostupná síť a pokud není tak ukázat poslední dostupná data.

- ## 4.8 Co je třeba zlepšit na serveru

- **Push notifikace** - Zatím na serveru chybí možnost push notifikací pro webové aplikace, takže uživatelé na počítačích nemohou být připomínáni důležité události jako v mobilní aplikaci. Případně stačí alespoň vystavit endpoint s poslední odeslanou push notifikací na mobilní telefon, jelikož tato informace není nikde dostupná.

- **Řízení sezení** - Podobně jako odhlašování je problém i přechod mezi sezeními. Problém se mi nakonec podařilo obejít a přijít s funkční řešením, k jednomu přechodu mezi sezeními se ale posílají nyní 2 API requesty místo jednoho. To není optimální.
- **Callback z platby** - Je třeba nastavit, aby úspěšné zaplacení v platební bráně vrátilo uživatele zpět do webové aplikace, nyní je to nastavené na mobilní aplikaci.
- **Optimalizace výpisu sezení** - Spousta příchozích informací pro stavbu sezení by se dala zjednodušit, jelikož by se značná část mohla vyhodnotit již na serveru. Nyní například server posílá bloky obsahu, které si s sebou nesou podmínku. Tato podmínka se musí odeslat zpět na server, aby se rozhodlo, zdali se daný obsah vypíše. To je velmi neefektivní, jelikož se to jednak musí posílat po jednom requestu a jednak by tato povinnost neměla vůbec v odpovědi ze serveru být, jelikož v první řadě nic nebrání vyhodnocení již na serveru.
- **Dostupnost odkazu k platbě** - V tuto chvíli je odkaz k platbě dostupný teprve v momentě, kdy uživatel musí zadat kód nebo zaplatit, aby mohl pokračovat. To znemožňuje zobrazení a uplatnění slev, neboť aplikace se před tímto termínem (například hned po registraci) nedostane k odkazu na platební bránu, na kterou má uživatele přesměrovat v případě, že chce zakoupit plnou verzi.

4.9 Nasazení

Během testování byla aplikace nasazena nejdříve na AWS hostingů ve verzi, která je zdarma, z čehož ale vznikly nečekané problémy, jelikož verze zdarma neumožňuje použití SSL certifikátu bez placení. Tato skutečnost měla mírně negativní účinky na testování a sice u stránek, které běží pouze přes HTTP protokol místo HTTPS protokolu:

- Nelze zobrazovat notifikace pomocí Notification API
- Nelze vyzkoušet nainstalování aplikace
- Prohlížeč upozorňuje na nezabepečené připojení, což může mírně mást uživatele

V průběhu byla tedy webová aplikace nasazená ještě pomocí Netlify, které nabízí nejenom nasazení zdarma, ale také rovnou opatří stránku SSL certifikátem.

Problém při komunikaci s existujícím řešením nastává velmi často v blokování požadavků z neznámé domény, což řeší CORS. CORS je zjednodušeně řečeno mechanismus, který umožňuje obejít fakt, že backendové řešení odmítá z důvodu bezpečnosti přijímat a regovat na požadavky z jiné než vlastní domény. To je problém hned ve vývoji, vyvíjí-li programátor například webovou aplikaci lokálně u sebe a připojuje-li se u toho ke vzdálenému serveru. Při vývoji se dá tento problém díky create-react-app obejít nastavením proxy [33], což vytvoří mezivrstvu, která zpracuje požadavky tak, že vypadají, jako by odcházely přímo z cílového serveru. Při nasazení ale není tento přístup možný a tak je nutné změnit nastavení cílového serveru, aby přijímal dotazy z jiné domény. Tohoto jsem docílil v Django pomocí instalace balíčků django-cors-headers a django-cors-middleware na serveru a dopsání kódu do konfigurace v souboru settings.py

```
CORS_ORIGIN_WHITELIST = ('url_testovane_domeny')
MIDDLEWARE_CLASSES = ('corsheaders.middleware.CorsMiddleware', ...)
INSTALLED_APPS = ('corsheaders', ...)
```

Samotné nasazení aplikace vytvořené pomocí create-react-app funguje pomocí těchto příkazů v (linuxovém) terminálu:

```
yarn build // vytvoří z kódu produkční verzi aplikace
sudo serve -s build -l 443 & // aplikace bude dostupná na adrese serveru
```

V aplikaci je ještě třeba změnit při nasazení proměnou v souboru `environment.tsx` s názvem `currentServerUrl`, jelikož během vývoje se obchází mechanismus CORS pomocí proxy, která při nasazení nefunguje, takže je třeba proměnnou `currentServerUrl` aktualizovat na adresu backendového řešení.

Kapitola 5

Uživatelské testování

Uživatelské testování probíhalo v několika rovinách. První forma testování s uživateli proběhla již při návrhu UI a sice dotazováním na preference mezi dvěma variantami stejných stránek. Další průběžné testování bez pevnějšího řádu a zatím s uživateli bez určitého profilu probíhalo již během implementace, jelikož dnes již klasické dílo v oboru uživatelského testování tvrdí, že „testování za pomoci jednoho uživatele v rané fázi projektu je lepší než testování s 50 uživateli na konci projektu“ [34]. Poslední a nejdůležitější fází, které se věnuje tato kapitola, bylo koordinované testování s uživateli, pro které je aplikace určena.

5.1 Specifika testování

Aplikaci otestovalo v poslední fázi 5 aktivních kuřáků ve věkovém zastoupení 22 až 45 let. V ideálním případě by bylo dobré, kdyby se této fáze zúčastnilo o několik lidí více a i z generace starších 50 let. I kvůli probíhající pandemické situaci však bylo těžší domluvit se s uživateli v této věkové kategorii na testování, jelikož testování probíhalo ve virtuálním prostoru přes sdílení obrazovky a hovor. Navíc se testování této aplikace vyznačovalo vyšší časovou náročností. Tato aplikace je specifická rozložením obsahu v čase. Server nabízí kuřákovi jednotlivá sezení vždy s časovým rozestupem, aby byla terapie efektivnější. Celá náplň aplikace je tedy rozložena do terapie trvající až 2 měsíce. Tolik času jsem pochopitelně na testování neměl. Málomocný uživatel je ochotný bez finanční kompenzace testovat aplikaci periodicky. U tří z pěti uživatelů jsme díky testovací funkcionalitě přeskočení času k dalšímu sezení aplikaci projeli najednou. Dva uživatelé byli ochotni vyzkoušet odvykání a postupovali déle než týden přirozeným tempem, takže jsme ověřili i standardní funkcionalitu postupných sezení. S blížícím se termínem jsem však při posledním hovoru i s těmito uživateli vyřešil zbytek sezení najednou. I tak ale aplikace vykazovala nadprůměrnou časovou zátěž pro uživatele. Celý program se skládá z více než 25 sezení, které trvají 5 až 20 minut. Při jednotném běhu jsme zvládli s uživateli všechna sezení projít za přibližně 60 až 90 minut. Dva z uživatelů, kteří procházeli aplikaci postupně, věnovali celkově testování více než 2.5 hodiny svého času.

5.2 Dotazník

- Jaký je Váš věk a Vaše pohlaví?
 - Žena 22 let, Muž 42 let, Žena 45 let, Žena 31 let, Muž 25 let
- Chystáte se v současné době přestat s kouřením jednou a provždy?
 - 2x Ano, 3x Ne
- Trávíte denně více volného času na telefonu nebo počítači?

- 3x Počítač, 2x Telefon
- Používáte počítač ve volném čase každý nebo téměř každý den?
 - 4x Ano, 1x Ne
- Už jste někdy nainstalovat webovou aplikaci z webového prohlížeče (po vysvětlení instrukcí jak se to dělá)?
 - 5x Nikdy
- Jak často povolujete stránkám na internetu ukazovat notifikace. Pakliže ano, kterým?
 - 4x Nikdy, 1x Téměř nikdy (jediná výjimka je aplikace na měření času, nebo když se překlíknu)

5.3 Příkladný scénář

1. Přejděte na danou adresu.
2. Zaregistrujte se.
3. Přihlašte se pomocí údajů, které jste zadali do registrace.
4. Povolte aplikaci notifikace až se Vás vyskakovací okno zeptá.
5. Proklikejte všechny odkazy, které vidíte v navigaci.
6. V sekci Můj účet změňte své oslovení.
7. V sekci Můj účet vyplňte libovolný dotazník.
8. Prohlédněte si sekci Mé úspěchy.
9. Přejděte na sezení a projděte dostupné sezení.
10. Podívejte se zdali došlo v sekci Mé úspěchy po sezení k nějaké změně.
11. Projděte další sezení.
12. Podívejte se na domácí obrazovku.
13. V sekci Můj plán změňte libovolný údaj.
14. Nyní projdeme 2 sezení. Vždy když jedno sezení dokončíme, tak Vás poprosím o chvíli strpení než se Vám zobrazí další.
15. Zadejte do aplikace kód, díky kterému se Vám otevře zbytek terapie.
16. Projděte dalších 6 sezení.
17. Podívejte se do sekce Mé úspěchy a projděte všechny získané odznaky.
18. Nainstalujte si aplikaci pomocí tlačítka v řádku do kterého zadáváte url adresy.
19. Zavřete webový prohlížeč a zkuste aplikaci otevřít z ikonky, která se Vám objevila na ploše.
20. Zkuste si na chvíli vypnout internet a podívat se, jestli se Vám načtou obrazovky v aplikaci. Po té si zase internet zapněte ať můžeme pokračovat v hovoru.
21. Vyberte si jeden odznak a po rozkliknutí změňte libovolný údaj a podívejte se, jestli se po přenačtení změna zachovala.
22. Podívejte se na další sezení. Až se Vás aplikace zeptá, zdali jste vydržel nekouřit, klikněte na „Ne“.
23. Podívejte se do sekce Moje podpora a přečtěte si nějaký tip k odolání chuti na cigaretu.
24. Projděte dalších 10 sezení.
25. Podívejte se na sekci Můj plán zdali přibýly nové údaje.
26. V sekci Moje podpora si otevřete libovolný odkaz, který Vás zavede mimo testovanou aplikaci.

27. Projděte zbytek sezení, dokud nezískáte všechny odznaky.
28. Po získání všech se vraťte do domácí obrazovky a zjistěte jestli se Vám zobrazí obrazovka gratulující ke konci Vaší cesty.

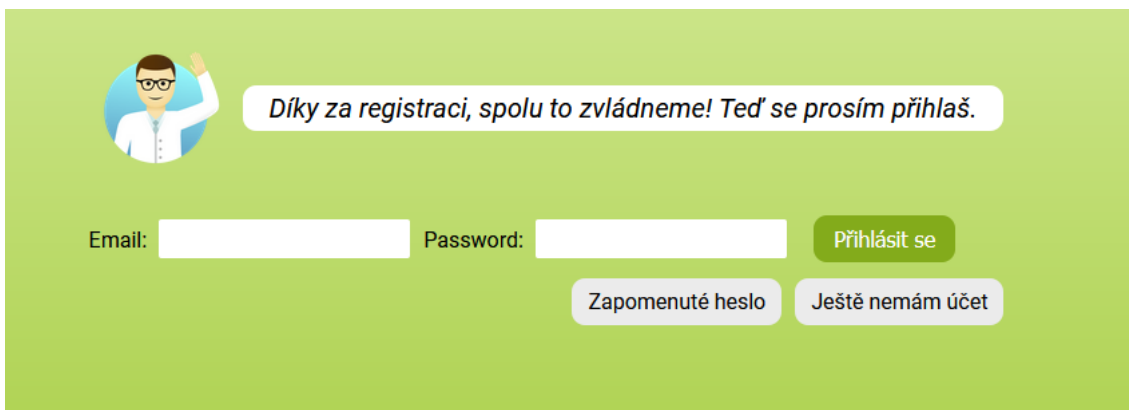
5.4 Vyhodnocení

Ani jednou během testování s uživateli se nestalo, že by aplikace spadla. Byly však samozřejmě objeveny některé skutečnosti, které se daly zlepšit. Nyní tedy uvedu změny provedené na základě zpětné vazby z testování.

5.4.1 Registrace

Ukázalo se, že registrace je pro uživatele trochu matoucí. Zjistil jsem, že zmatení působí z pocitu zahlcení příliš mnoha informacemi, obsahu na více místech a následném přesměrování na přihlášení bez upozornění. Rozhodl jsem se tedy pro následující úpravy:

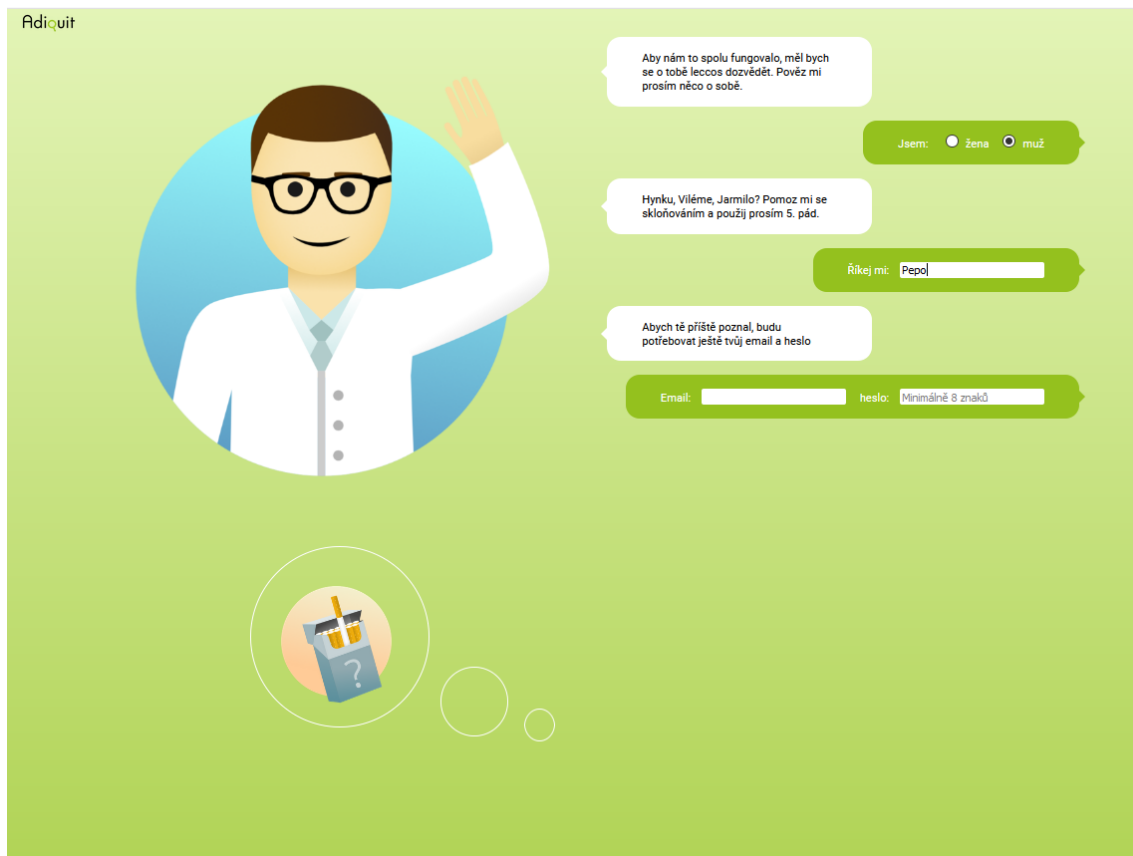
- **Texty stále na stejné straně** - Uživatelé museli v dřívější verzi přecházet očima z jedné strany obrazovky na druhou. Jako lepší řešení se ukázalo držet po celou dobu obrázky na jedné straně a texty na druhé (obrázek 5.1).
- **Postupné odkrývání zpráv** - V návrhu měla tato stránka všechny informace viditelné najednou. U uživatelů s ale ukázala zahlcenost obsahem. Jako řešení jsem tedy zvolil postupné odhalování zpráv pomocí animace a to až tehdy, kdy uživatel vyplní předešlé pole. Vstupy jsou mu tedy předkládány postupně.
- **Nápověda při přihlašování po registraci** - Uživatelé byli zmateni z přesměrování na přihlášení po registraci. Přidal jsem tedy zprávu ve formě nápovědy i do přihlašování, která uživateli odtajní další informaci a napoví mu, co se od něj očekává (obrázek 5.2).



Obrázek 5.2. Obrazovka Přihlášení s přidanou nápovědou po úspěšné registraci

5.4.2 Načítání sezení

Problémy, na které bylo třeba reagovat, se ukázaly i při sezení. Při obdržení dat reprezentujících sezení ještě bohužel probíhalo vykreslení sezení, jelikož zpráva ze serveru obsahuje části, které je potřeba poslat zpátky na server a podle odpovědi se ukáže, zdali se mají tyto části vykreslit. Nejdříve jsem chtěl zachovat co největší dostupnost sezení a začít ukazovat dostupné prvky rovnou, avšak někteří uživatelé přetočili stránku sezení dříve, než se stihly vykreslit všechny prvky. Dalším problémem bylo, že při opakovaném kliknutí na přetočení stránky existoval malý časový interval, během kterého šlo



Obrázek 5.1. Obrazovka Registrace po reflektování poznatků z uživatelského testování

pomocí opakovaného kliknutí na přetočení stránky přetočit několik sezení za sebou a tedy několik sezení minout. V zájmu autenticity obsahu jsem tedy rozhodl o ukázání obrazovky načítání ihned po přetočení stránky a to do té doby, dokud se nenačtou všechny elementy na stránce. Tímto krokem jsem vyřešil oba dva zjištěné problémy se sezeními (obrázek 5.3).



Obrázek 5.3. Obrazovka Sezení s přidanou fází načítání dokud nejsou k dispozici všechny prvky pro předcházení problémům

■ 5.4.3 Drobné změny

Po fázi testování byla změněna řada dalších menších věcí za účelem zlepšení uživatelského prožitku. Pomocí jednoho z úkolů (otázka na proklikání detailů všech odznaků) jsem například zaznamenal, že někteří uživatelé si nevšimli šipek pro rychlé procházení mezi odznaky. Šipky po stranách vyskakovacích oken byly tedy zvýrazněny. Další změny zahrnovaly například zvětšení vstupů při výběru z více možností, aby se na ně dalo snadněji klikat, zvýraznění některých odkazů, opravení textů a jiné.

■ 5.4.4 Zhodnocení prvků PWA

Překvapením při testování bylo, jak uživatelé nakládají s prvky progresivních webových aplikací. Když se na stránce objevila žádost aplikace ohledně zobrazování notifikací, tak každý uživatel instinktivně začal klikat na zamítnutí. Každý z dotazovaných uživatelů řekl, že ještě nikdy nenainstaloval aplikaci z webového prohlížeče do počítače (dokonce ani nevěděl o této možnosti). Navíc, naprostá většina uživatelů odpověděla, že vždy, když jsou na počítači, tak mají k dispozici internet. Z těchto informací mi vyšel závěr, že ač webová aplikace má naimplementované některé elementy progresivních webových aplikací, tak pokud jsou určeny k používání převážně na počítačích, zůstanou pravděpodobně naprostou většinou uživatelů nevyužity. Ikdyby tedy byly na straně serveru hotové push notifikace, tak by jich většina uživatelů pravděpodobně nevyužila, neboť by neměli aplikaci nainstalovanou, ani stále spuštěnou, případně ani povolené notifikace.

Kapitola 6

Závěr

6.1 Zhodnocení

Podařilo se mi zaznamenat požadavky existující mobilní aplikace a proměnit je ve fungující aplikaci webovou, přičemž jsem se musel vypořádat s hned několika neznámými prostředími (vyčtení API endpointů z kódu serveru, pochopení některých mechanismů z kódu mobilní aplikace, nasazení aplikace). Návrh práce jsem průběžně testoval na uživateli a získával zpětnou vazbu, z níž jsem většinu reflektoval v následných úpravách. Prozkoumal jsem nejdůležitější prvky progresivních webových aplikací a v podkapitole 4.7 popsal jejich uplatnění a omezení vzhledem k současné situaci, používání aplikace i možnostem serveru. Byly objeveny nedostatky na straně serveru, které jsem zaznamenal v podkapitole 4.8 a poskytl vysvětlení či návrh, jak je třeba rozšířit funkcionality serveru a zdali se jedná o změnu s kritickou prioritou.

Jako osobní přínos práce považuji získání solidní znalosti frameworku React, ocenění přínosu typescriptu, procvičení práce s grafickým editorem a spoustu dalších drobnějších dovedností jako například vědomosti o nasazování aplikací a povědomí o psaní mobilních aplikací, jelikož kód aplikace byl jeden z primárních zdrojů, ze kterého jsem musel čerpat.

6.2 Rozšíření aplikace

Na téměř každé aplikaci je co zlepšit a tato tomu není výjimkou. V následujících bodech shrnu drobné i větší funkcionality, které buď nebyly zprvu cílem práce a vyplynuly až později, nebo nějaká skutečnost ztížila jejich realizaci natolik, že jsem dal přednost ostatním funkcionalitám.

- **Samostatný chatovací systém** - Jeden z požadavků mobilní aplikace byl původně ohledně kontaktování blízkých osob, které by kuřákovi při zaváhání mohli pomoci. Mobilní aplikace má výhodu ve snadném přístupu k seznamu kontaktů a existujícímu způsobu vzdálené komunikace, u webové aplikace ale taková možnost ze základu přístupná není. Do budoucna by mohlo být výhodné zauvažovat o způsobu (ideálně okamžité) komunikace s podporovatelem kuřáka, možná pomocí další aplikace pro podporovatele.
- **Drobnější změny**
 - **Vyzkoušení bez registrace** - Mobilní aplikace má výhodu v možnosti režimu „na zkoušku“, kdy uživatel může vyzkoušet aplikaci bez registrace. Všechny údaje k identifikaci se přitom ukládají u uživatele do jeho telefonu. Uložiště pro webové aplikace má ale zpravidla kratšího trvání, bylo tedy riskantní ukládat identifikátor z webové aplikace, neboť hrozí ztráta dat a možnosti se přihlásit, stačí totiž aby uživatel promazal data nebo použil anonymní mód.
 - **Více barevných režimů** - Webová aplikace je laděna do zelené barvy a na některých místech upřednostňuje temné pozadí a bílý text. Při testování každý z uživatelů

odpověděl, že nemá se čtením textu problémy a že se mu barevné provedení líbí, ale přesto mohou někteří uživatelé upřednostňovat jiné barevné kombinace textu a pozadí.

- **Cizojazyčné texty** - Toto je jeden z požadavků, který nebyl v počátku zaznamenán, neboť aplikace nemá možnost změny jazyka, ale bere jej z informací dostupných z nastavení mobilního telefonu. Texty již existují, ale bylo časově náročné vytahovat jednotlivé texty a jejich překlady z kódu mobilní aplikace, tak jsem se prozatím omezil na české varianty textů. Počítám s tím, že před nasazením do provozu se ještě některé texty budou měnit. Pokud bude k dispozici vyexportovaný seznam textů v české verzi a jejich ekvivalenty, tak se do aplikace přidá velmi snadno, jde pouze o nahrazení proměnných.
- **Správa zasílaných emailů** - V kapitole 5.4.4 jsem popsal, že uživatelé často nevyužívají elementů progresivních webových aplikací, ač jsou dostupné, protože instinktivně zakazují notifikace a nejsou zvyklí instalovat na počítač aplikace z webové prohlížeče. Pokud je pro aplikaci důležité upozornit uživatele na nějakou událost a nejsou dostupné push notifikace jako na telefonech, jednou z variant může být zasílání emailů. Server by tedy mohl posílat vždy, když je sezení připraveno, uživateli email a v sekci účet by tedy mohla přibýt správa zasílaných emailů.
- **Drobné úpravy pro prohlížeč Safari** - Každé prohlížečové jádro má drobné odchylky od těch ostatních, co se vykreslování týče. Aplikaci jsem testoval s uživateli (a také sám) v prohlížečích Chrome, Firefox a Microsoft Edge. Situace se Safari je bohužel obtížná, neboť se vyznačuje silnou vazbou na produkty firmy Apple a nemožností nainstalování na jiných systémech. Různé prohlížeče většinou vykazují jen drobné změny, bez zařízení schopného spustit Safari ale nedokáží tento prohlížeč otestovat, takže je pravděpodobné, že menší části obsahu webové aplikace nebudou v tuto chvíli vypadat tak, jak bylo zamýšleno.
- **Drobné změny textů a ikonek** - Při vytváření práce jsem použil ikonky a texty z mobilní aplikace. Některé z nich (například k odznaku „Instalace aplikace“) zobrazují mobilní telefon. Nechtěl jsem pro jednu odlišnost porušit jednotnost všech ostatních ikonek, a tak jsem zanechal ty stejné. Bylo by dobré tyto ikony nahradit doplňky pro webovou aplikaci.

K nasazení do skutečného provozu by tedy měly být ještě doplněny jazykové varianty textů, zatím je však aplikace používána převážně v České Republice a chybějící překlad nebránil otestování. Jelikož je ale třeba k reálnému nasazení několika změn na serveru podle podkapitoly 4.8, tak věřím, že se na straně klienta jedná o změny malého rozsahu, oproti trochu větším změnám na straně serveru.

Literatura

- [1] World Health Organization. *Fact Sheets - Tobacco*. 2020.
<https://www.who.int/news-room/fact-sheets/detail/tobacco>. [online] (navštíveno dne 2020-12-23).
- [2] Gary A. Giovino. *The Tobacco Epidemic in the United States*. 2007.
<http://www.sciencedirect.com/science/article/pii/S0749379707005673>. National Institutes of Health State-of-the-Science Statement on Tobacco Use.
- [3] John R. Hughes, Josue Keely a Shelly Naud. *Shape of the relapse curve and long-term abstinence among untreated smokers*. 2004.
- [4] A. Kulháněk, R. Gabrhelík, D. Novák, V. Burda a H Brendryen. *eHealth Intervention for Smoking Cessation for Czech Tobacco Smokers: Pilot Study of User Acceptance*. 2019.
- [5] *Progressive web apps (PWAs)*.
https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps. [online] (navštíveno dne 2020-12-23).
- [6] Eva Králíková, Richard Češka, Alexandra Pánková, Lenka Štěpánková, Kamila Zvolská, Vladislava Felbrová, Stanislava Kulovaná a Miroslav Zvolský. *Doporučení pro léčbu závislosti na tabáku*. 2015.
www.kardio-cz.cz/data/upload/Doporuceni_pro_lecbu_zavislosti_na_tabaku.pdf.
- [7] *Závislost na tabáku: úvod*.
<https://www.nzip.cz/clanek/432-zavislost-na-tabaku-uvod>. [online] (navštíveno dne 2020-12-23).
- [8] Michael Chaiton, Lori Diemert, Joanna E Cohen, Susan J Bondy, Peter Selby, Anne Philipneri a Robert Schwartz. *Estimating the number of quit attempts it takes to quit smoking successfully in a longitudinal cohort of smokers*. 2016.
<https://bmjopen.bmj.com/content/6/6/e011045>.
- [9] *Different news audiences for mobile web and apps*.
www.americanpressinstitute.org/publications/mobile-web-app-audiences/. [online] (navštíveno dne 2020-12-23).
- [10] *Google play service fees*.
<https://support.google.com/googleplay/android-developer/answer/112622?hl=en>. [online] (navštíveno dne 2020-12-23).
- [11] *GoPay price*.
<https://www.gopay.com/en/price.html>. [online] (navštíveno dne 2020-12-23).
- [12] *Mobile vs Desktop Internet Usage*.
www.broadbandsearch.net/blog/mobile-desktop-internet-usage-statistics. [online] (navštíveno dne 2020-12-23).
- [13] Christina Bröhl, Peter Rasche, Janina Jablonski, Sabine Theis, Matthias Wille a Alexander Mertens. *Desktop PC, Tablet PC, or Smartphone? An Analysis of Use*

- Preferences in Daily Activities for Different Technology Generations of a Worldwide Sample.* In: 2018. 3-20. ISBN 978-3-319-92033-7.
- [14] Elias Papadopoulos, Michalis Diamantaris, Panagiotis Papadopoulos, Thanasis Petsas, Sotiris Ioannidis a Evangelos Markatos. *The Long-Standing Privacy Debate: Mobile Websites vs Mobile Apps.* In: 2017. 153-162.
- [15] *Desktop vs. mobile app design: how to optimize your user experience.*
<https://99designs.com/blog/web-digital/desktop-vs-mobile-app-design/>. [online] (navštíveno dne 2020-12-23).
- [16] Andreas Bjørn-Hansen, Tim A. Majchrzak a Tor-Morten Grønli. *Progressive Web Apps for the Unified Development of Mobile Applications.* 2018.
- [17] *Introduction to progressive web apps.* 2020.
developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction. [online] (navštíveno dne 2020-12-23).
- [18] *Service Workers: an Introduction.* 2020.
<https://developers.google.com/web/fundamentals/primers/service-workers>. [online] (navštíveno dne 2020-12-23).
- [19] *Cache API.* 2020.
<https://developer.mozilla.org/en-US/docs/Web/API/Cache>. [online] (navštíveno dne 2020-12-23).
- [20] *Caching Files with Service Worker.* 2020.
<https://developers.google.com/web/ilt/pwa/caching-files-with-service-worker>. [online] (navštíveno dne 2020-12-23).
- [21] *2020 Developer Survey.* 2020.
<https://insights.stackoverflow.com/survey/2020technology-web-frameworks-professional-developers2>. [online] (navštíveno dne 2020-12-23).
- [22] *Tutorial: Intro to React.* 2020.
<https://reactjs.org/tutorial/tutorial.html>. [online] (navštíveno dne 2020-12-23).
- [23] *Understanding Unidirectional Data Flow.* 2020.
<https://www.pluralsight.com/guides/understanding-unidirectional-data-flow>. [online] (navštíveno dne 2020-12-23).
- [24] *User Interface Design Patterns.* 2020.
<http://ui-patterns.com/patterns>. [online] (navštíveno dne 2020-12-23).
- [25] *Design Testing.* 2020.
<https://uxdesigntemplates.com/evidence/method/design-testing>. [online] (navštíveno dne 2020-12-23).
- [26] *What is conversation design?* 2020.
<https://designguidelines.withgoogle.com/conversation/conversation-design/what-is-conversation-design.html>. [online] (navštíveno dne 2020-12-23).
- [27] Stefan Hanenberg, Sebastian Kleinschmager, Romain Robbes, Éric Tanter a Andreas Stefk. An empirical study on the impact of static typing on software maintainability. *Empirical Software Engineering.* 2013, 19 DOI 10.1007/s10664-013-9289-1.
- [28] *NPM vs. Yarn: Which Package Manager Should You Choose?* 2020.
<https://www.whitesourcesoftware.com/free-developer-tools/blog/npm-vs-yarn-which-should-you-choose/>. [online] (navštíveno dne 2020-12-23).

- [29] *API Documentation*. 2020.
<https://swagger.io/solutions/api-documentation/>. [online] (navštíveno dne 2020-12-23).
- [30] *Adapter*. 2020.
<https://refactoring.guru/design-patterns/adapter>. [online] (navštíveno dne 2020-12-23).
- [31] *Desktop Screen Resolution Stats Worldwide*. 2020.
<https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide>. [online] (navštíveno dne 2020-12-23).
- [32] *Using HTTP cookies*. 2020.
<https://developer.mozilla.org/en-US/docs/Web/HTTP/Cookies>. [online] (navštíveno dne 2020-12-23).
- [33] *Proxying API Requests in Development*. 2020.
<https://create-react-app.dev/docs/proxying-api-requests-in-development/>. [online] (navštíveno dne 2020-12-23).
- [34] Steve Krug. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. 3. vydání. USA: New Riders Publishing, 2014. ISBN 0321965515.



Příloha A

Zkratky

API	Application Programming Interface
CORS	Cross-Origin Resource Sharing
CSS	Cascading Style Sheets
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
JSON	JavaScript Object Notation
PWA	Progressive Web App
SSL	Secure Sockets Layer
UI	User Interface



Příloha B

Obsah elektronické přílohy

- **client/** - složka obsahující zdrojový kód webové aplikace
- **design/** - složka pro exportovaný návrh prototypu webové aplikace
- **dokumentace-api.yaml** - dokumentace API, se kterým webová aplikace komunikuje