

Diplomová práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Buddy systém pro odvykání kouření

Bc. Martin Turyna

Školitel: doc. Ing. Daniel Novák, Ph.D.

Obor: Otevřená informatika

Zaměření: Softwarové inženýrství

Leden 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Turyňa** Jméno: **Martin** Osobní číslo: **466151**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Otevřená informatika**
Specializace: **Softwarové inženýrství**

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce:

Buddy systém pro odvykání kouření

Název diplomové práce anglicky:

Smoking Cessation Buddy System

Pokyny pro vypracování:

Cílem práce je problematika kouření a dalším vývoj mobilní aplikace Buddy App určené pro rodinu a přátele odvykajícího kuřáka. Buddy App by měla poskytovat cenné informace o průběhu odvykání, Součástí práce bude softwarová analýza projektu a testování výsledného produktu.

- 1) Seznamte se s problematikou odvykání kouření.
- 2) Designujte BuddyApp aplikace o nové funkcionality.
- 3) Aplikace by měla zahrnovat následující funkcionality: plán odvykacího procesu, statistiky a zasílání zpráv
- 3) Implementujte navržené funkcionality včetně propojení na server
- 4) Proveďte studii na menším počtu uživatelů a vyhodnoťte užitečnost navržené aplikace.

Seznam doporučené literatury:

- [1] Michie, S., Hyder, N., Walia, A. & West, R. Development of a taxonomy of behaviour change techniques used in individual behavioural support for smoking cessation. 2011. Addictive behaviors, 36(4), 315–319. doi: 10.1016/j.addbeh.2010.11.016
- [2] Medlock, S. & Wyatt, JC. Health Behaviour Theory in Health Informatics: Support for Positive Change. 2019. Chapter in Applied Interdisciplinary Theory in Health Informatics, 146 – 158.

Jméno a pracoviště vedoucí(ho) diplomové práce:

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

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

Datum zadání diplomové práce: **13.02.2023**

Termín odevzdání diplomové práce: _____

Platnost zadání diplomové práce: **22.09.2024**

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

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

Datum převzetí zadání

Podpis studenta

Poděkování

Rád bych tímto poděkoval vedoucímu své diplomové práce doc. Ing. Danielu Novákovi, Ph.D. za pevné nervy, vstřícný přístup a cenné rady při psaní této práce. Rád bych též poděkoval své rodině a přátelům za poskytnutí maximální podpory a v neposlední řadě děkuji svojí Kačence, která mi byla pevnou oporou ve dne v noci nejen při psaní diplomové práce, ale také během celého studia.

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a v souladu s Metodickým pokynem o dodržování etických principů pro vypracování závěrečných prací, a že jsem uvedl všechny použité informační zdroje.

V Praze, 9. ledna 2024

.....

Abstrakt

Tato diplomová práce se soustředí na analýzu, návrh a implementaci multiplatformní mobilní aplikace Buddy a příslušného backendového řešení. Hlavním účelem aplikace Buddy je poskytnout podporu rodině a blízkým osobám kuřáka, který podstupuje digitální terapii pro odvykání kouření. Aplikace je vyvinuta v jazyce React Native, zatímco backend je realizován pomocí frameworku Spring Boot v jazyce Java.

Klíčová slova: React Native, Odvykání kouření, Javascript, Expo, Mobilní aplikace, Spring boot, Java, Databáze, Testování, Analýza

Školitel: doc. Ing. Daniel Novák, Ph.D.

Abstract

This thesis focuses on the analysis, design and implementation of a multiplatform mobile application Buddy and the corresponding backend solution. The main purpose of the Buddy app is to provide support to the family and loved ones of a smoker undergoing digital therapy for smoking cessation. The app is developed in React Native while the backend is implemented using the Spring Boot framework in Java.

Keywords: React Native, Smoking Cessation, Javascript, Expo, Mobile app, Spring boot, Java, Database, Testing, Analysis

Title translation: Smoking Cessation Buddy System

Obsah

1 Úvod	1	4.3.3 Dashboard a fáze	27
1.1 Motivace	1	4.3.4 Ukázat cestu a odznaky	27
1.2 Cíl	1	4.3.5 Widget s ušetřenými penězi a obrazovka přání	28
1.3 Struktura práce	1	4.3.6 Widget a seznam sezení	29
Část I		4.3.7 Widget podpory a rychlá rada	31
Teoretická část		4.3.8 Widget s reporty a seznam reportů	32
2 Historie a statistiky kouření	5	4.3.9 Posílání zpráv	32
2.1 Stručný pohled do historie tabáku a cigaret	5	4.4 Popis technické části	33
2.2 Statistiky celosvětově	7	5 Modelování požadavků, návrh a design	37
2.2.1 Úmrtnost na kouření mezi lety 1990 až 2019	8	5.1 Návaznost na předchozí práci	37
2.2.2 Počet prodaných cigaret na osobu za den	9	5.2 Funkční požadavky	38
2.3 Statistiky v České republice	10	5.2.1 Nové funkční požadavky mobilní aplikace Buddy	38
2.4 Škodlivost kouření	12	5.2.2 Odstranění původních požadavků	39
2.5 Opatření pro redukci kouření	13	5.3 Návrh JSON zpráv pro rozhraní mezi aplikací Buddy a serverovou částí	39
2.6 Výrobky určené ke kouření	14	5.3.1 Odznaky	39
2.7 Používání tabákových a nikotinových výrobků mezi mladistvými	16	5.3.2 Reporty	40
2.8 Metody odvykání kouření	17	5.3.3 Odpověď kuřáka a historie odpovědí	40
2.8.1 Nicotine Replacement Therapy	17	5.3.4 Zpráva kuřákovi	41
2.8.2 E-Health	17	5.4 Návrh serverové části	42
2.8.3 Nikotinové vakcíny	18	5.4.1 Backendový framework - Spring Boot	42
3 Metody vývoje mobilních aplikací	19	5.4.2 Databáze	43
3.0.1 Trendy vývoje mobilních aplikací	19	5.4.3 Další technologie	43
3.0.2 Nativní vs multiplatformní vývoj aplikací	19	5.4.4 Verze technologií	43
3.0.3 Nativní vývoj mobilních aplikací	20	5.5 Architektura	44
3.0.4 Multiplatformní vývoj mobilních aplikací	20	5.6 Databázový model	44
Část II		5.7 Načítání a potvrzení odpovědi kuřáka	46
Realizační část		6 Implementace frontendové části - Mobilní aplikace	49
4 Analýza výchozího stavu aplikace Buddy	25	6.1 Aktualizace zastaralého načítání aplikace	49
4.1 Mobilní aplikace pro odvykání kouření	25	6.2 Aktualizace odznaků	50
4.2 Mobilní aplikace Buddy	25	6.3 Widget s odpovědí kuřáka	52
4.3 Popis jednotlivých funkcionalit	25	6.4 Historie odpovědí	54
4.3.1 Registrace a přihlášení	26	6.5 Mapper pro skloňování textu ze serverové části	56
4.3.2 Onboarding	26		

7 Implementace backendové části - Spring Boot aplikace	57
7.1 Odpověď kuřáka	57
7.2 Odznaky	59
7.3 Report scheduler	60
7.4 Ošetření chybových stavů	61
7.5 Návrhové vzory a principy použité při vývoji	62
8 Testování	65
8.1 Moderované testování mobilní aplikace	65
8.1.1 Příprava scénářů a dotazníku	66
8.1.2 Průběh testování	67
8.1.3 Vyhodnocení výsledků	70
8.2 Testování serverové části	71
8.2.1 Unit testy - použité nástroje	72
9 Budoucí práce	75
10 Závěr	77
Literatura	79
Přílohy	
A Seznam zkratk	87
B Odpovědi respondentů z testovacího dotazníku	89

Obrázky

2.1 Konzumace různých forem zpracovaného tabáku na osobu za rok ve Spojených státech amerických, 1880-1995, zdroj: U.S. Department of Agriculture, 1996	6
2.2 Konzumace na osobu za rok a celková konzumace cigaret ve Spojených státech amerických, 1900-1995, zdroj: U.S. Department of Agriculture, 1996	6
2.3 Úmrtnost na kouření v roce 1990, zdroj: [7]	8
2.4 Úmrtnost na kouření v roce 2019, zdroj: [7]	9
2.5 Průměrný počet prodaných cigaret mezi lety 1875 až 2015 ve vybraných zemích, zdroj: [7]	10
2.6 Kuřáci tabákových výrobků celkem (denní a příležitostní) v České republice, zdroj: [11]	11
2.7 Kuřáci tabákových výrobků podle pohlaví (denní a příležitostní) v České republice, zdroj: [11]	11
2.8 Počet kusů cigaret v průměru vykouřených za den v České republice, zdroj: [11]	12
2.9 Zavedení alespoň jednoho bodu MPOWER v letech 2007-2020, zdroj: [9]	14
2.10 Druhy tabákových výrobků a jejich členění na skupiny a podskupiny, zdroj: [12]	15
2.11 Rozdělení alternativních nikotinových výrobků, převzato z [10], původní zdroj: Společnost pro léčbu závislosti na tabáku	15
2.12 Míra rizika - rizikové kontinuum různých tabákových a nikotinových výrobků převzato z: [10], původní zdroj: Abrams et al. (2018) podle Nutt a kol. (2014)	16
3.1 Popularita mobilních frameworků v letech 2019-2022, zdroj: [19]	21
4.1 (a) Uvítací stránka, (b) Přihlašovací formulář, (c) Registrační formulář zdroj: Autor - Aplikace Buddy	26
4.2 Obrazovka cesta a úspěchy (a) 1. část, (b) 2. část, (c) 3. část, zdroj: Autor - Aplikace Buddy	28
4.3 Widget s ušetřenými penězi (a) přípravná fáze, (b) odvykací fáze, zdroj: Autor - Aplikace Buddy	28
4.4 Widget sezení, zdroj: Autor - Aplikace Buddy	30
4.5 (a) První otázka, (b) Druhá otázka, (c) Čtvrtá otázka zdroj: Autor - Aplikace Buddy	30
4.6 (a) Obrazovka rychlé rady, (b) Rada pokud je kuřák smutný, Autor - Aplikace Buddy	31
4.7 Widget podpory, zdroj: Autor - Aplikace Buddy	31
4.8 Widget s reporty, zdroj: Autor - Aplikace Buddy	32
4.9 Obrazovka pro odeslání zprávy, zdroj: Autor - Aplikace Buddy	33
5.1 Databázový model - UML diagram tříd, zdroj: Autor	45
5.2 Interakce mezi objekty - načítání odpovědi, zdroj: Autor	47
5.3 Interakce mezi objekty - potvrzení odpovědi, zdroj: Autor	47
6.1 Obrazovka s odznaky (a) 1. část, (b) 2. část, zdroj: Autor - Aplikace Buddy	52
6.2 (a) Widget s odpovědí, (b) Widget bez odpovědi, zdroj: Autor - Aplikace Buddy	53
6.3 Obrazovka historie odpovědí, zdroj: Autor - Aplikace Buddy	55
7.1 Grafický rozbor hodnot parametru cron, zdroj: [32]	61
8.1 Graf znázorňující míru výskytu chyb na počet testujících uživateli, zdroj: NN Group [31]	67

Kapitola 1

Úvod

1.1 Motivace

Motivací pro vznik této diplomové práce je paralelně probíhající vývoj související *Mobilní aplikace pro odvykání kouření*. Vývoj této aplikace v současné době probíhá pod záštitou pracoviště Analýzy a interpretace biomedicínských dat na ČVUT, FEL ve spolupráci s Klinikou adiktologie na 1. lékařské fakultě Univerzity Karlovy. *Mobilní aplikace pro odvykání kouření* pomáhá kuřákům s odvykáním formou digitální automatizované terapie. Paralelně s touto aplikací je ve vývoji mobilní aplikace Buddy, která má sloužit přátelům, známým a rodině (dále jako uživatel) odvykajícího kuřáka. Cílem aplikace Buddy je poskytnout rozhraní, které jim nabídne aktuální informace o průběhu odvykání kuřáka. Rozvojem aplikace Buddy a implementací příslušného serverového řešení se právě zabývá tato diplomová práce.

1.2 Cíl

Cílem této práce bude zanalyzovat současné řešení mobilní aplikace Buddy, která slouží pro podporu odvykajících kuřáků používajících *Mobilní aplikaci pro odvykání kouření*. Z analýzy následně provedeme návrh na implementaci nových funkcionalit a tyto funkcionality implementujeme včetně propojení na server. Implementaci poté podrobíme testování a vyvodíme závěr společně s návrhem budoucích implementačních kroků.

1.3 Struktura práce

Tato práce bude členěna do dvou částí. V první části se budeme zabývat teorií, ve které rozebereme problematiku odvykání kouření. V další kapitole se poté podíváme na moderní řešení a technologie pro vývoj mobilních aplikací. Ve druhé, realizační části nejprve v první kapitole provedeme analýzu současného řešení aplikace Buddy. Posléze se v další kapitole budeme v rámci analýzy věnovat návrhu nových funkcionalit pro frontendovou část aplikace. Naváže kapitola, ve které analyzujeme řešení a požadavky pro vytvoření backendové

části (dále také jako serverová část) systému a následně také její integrace s frontendovou částí. Další kapitola se bude zabývat konkrétní implementací řešení. Popíšeme si některé stěžejní komponenty a vysvětlíme si principy a technologie použité při vývoji. Následující kapitola poté rozebere metodiku moderovaného testování frontendové části a Unit testy pro testování serverové části. Dále definujeme budoucí implementační kroky a v poslední kapitole učiníme závěr.



Část I

Teoretická část

Kapitola 2

Historie a statistiky kouření

2.1 Stručný pohled do historie tabáku a cigaret

Na základě archeologických studií se má za to, že tabák používali již staří Mayové v prvním století před našim letopočtem. Mayové, kteří obývali střední Ameriku, používali tabákové listy ke kouření při jejich posvátných a náboženských ceremoniálech. Tato tradice se poté začala rozšiřovat do jižní Ameriky během migrace Mayské komunity mezi lety 470 až 670 našeho letopočtu. Šamané a léčitelé těchto kmenů našli pro tabákové listy mnoho využití a používali je zejména k léčbě mnoha nemocí a zranění. [1]

V roce 1492, kdy se Kryštof Kolumbus vrátil z Ameriky do Evropy se svou expedicí, přivezl s sebou tabákové listy a semínka. V druhé polovině 16. století se tabák rozšířil po Evropě, především ve Francii a Portugalsku. Později byl tabák exportován do Asie a Afriky. V roce 1577 jej evropští lékaři začali doporučovat pro léčbu bolesti zubů, odpadávaní nehtů, horečky a rakoviny. Pro léčebné účely se tabák nejen kouřil, ale také žvýkal nebo louhoval v horké vodě. [2, 3]

O škodlivosti tabáku a jeho kouření se začaly objevovat informace již v roce 1601, kdy se v Londýně začal šířit anonymní pamflet s názvem *Work for Chimney-Sweepers*, který tabák popisoval jako jed, jehož používání pro medicínské účely má poškozující účinky. [2]

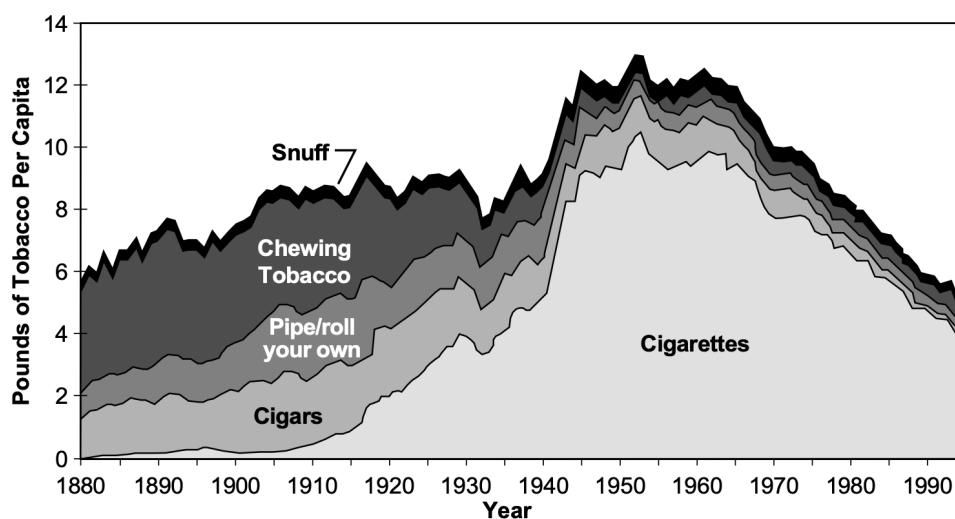
O 3 roky později, tedy v roce 1604, anglický král Jakub I. sepsal proti tabáku varování, ve kterém zmiňuje, že kouření má neblahé účinky na oči, nos, mozek a plíce. [3] Zároveň zakázal jakoukoliv kultivaci tabáku v Anglii a tabák byl zakázaný importovat do Anglie pro všechny, kdo nebyli držitelem patentu. [4]

Během 19. století postupně docházelo k úpadku využívání tabáku pro medicínské účely, a tak se nejpopulárnějším využitím tabáku stalo kouření pro vlastní potěšení. [2, 3]

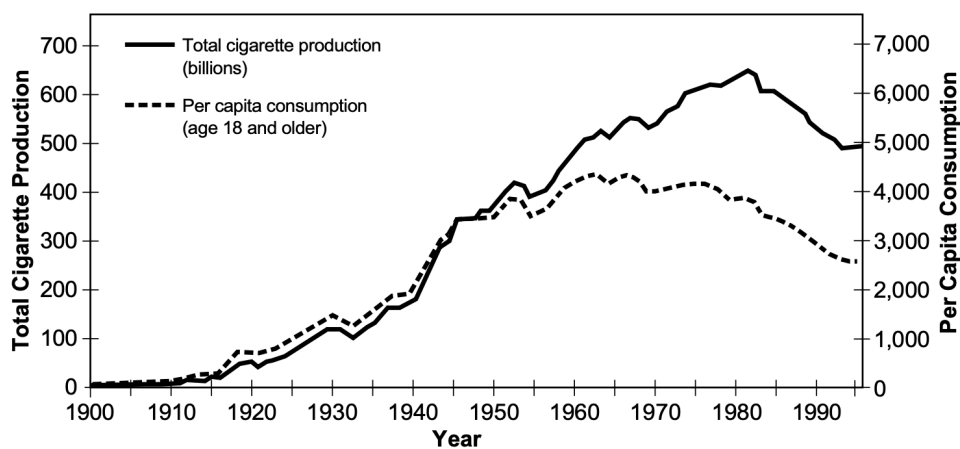
První patentovaný stroj na výrobu cigaret vymyslel Juan Nepomuceno Adorno, jenž si nechal zapsat patent 24. února 1847. [4] V témže roce začal v Londýně prodávat ručně balené turecké cigarety Phillip Morris a v roce 1854 poté začal prodávat své vlastní cigarety. [3]

V 18. a 19. století dominovalo využití tabáku především ve formě žvýkacího a šňupacího nebo prostřednictvím doutníku či fajfky. Ve 20. století však došlo

k masivnímu nárůstu a popularizaci cigaret v západním světě, především ve Spojených státech amerických. [5]



Obrázek 2.1: Konzumace různých forem zpracovaného tabáku na osobu za rok ve Spojených státech amerických, 1880-1995, zdroj: U.S. Department of Agriculture, 1996



Obrázek 2.2: Konzumace na osobu za rok a celková konzumace cigaret ve Spojených státech amerických, 1900-1995, zdroj: U.S. Department of Agriculture, 1996

Z obrázku 2.1 vyplývá, že přibližně do roku 1910 bylo kouření cigaret ve Spojených státech amerických napříč všemi formami konzumace tabáku poměrně neobvyklé. Od tohoto roku však dochází k masivní adopci kouření cigaret a od roku 1920 cigarety tvořily téměř jednu čtvrtinu z celkového množství spotřeby tabáku na osobu za rok.

Velkou roli na vzestupu konzumace cigaret hrají obě Světové války, během nichž byly cigarety distribuovány vojákům zdarma. V roce 1950 ve Spojených státech amerických kouřilo již 50 % všech dospělých osob. [2, 5]

Absolutního vrcholu bylo dosaženo podle obrázku 2.2 v roce 1963, kdy průměrná konzumace cigaret na osobu ve Spojených státech amerických dosáhla 4345 cigaret za rok, což odpovídá zhruba 11,9 cigaretám denně.

V roce 1967 Federální komise pro komunikace ve Spojených státech amerických prosadila vyvážení reklam na cigarety v televizním a radiovém vysílání s protikuřáckými reklamami. Tyto protikuřácké reklamy byly vysílány zdarma a alokovaný čas pro ně byl zhruba v hodnotě 75 milionů dolarů ročně. O tři roky později, tedy v roce 1970, byly reklamy na cigarety zcela zakázány. Masivní protikuřácká kampaň vyústila v celkový pokles kuřáků, což je patrné z obrázků 2.1 a 2.2 [5].

2.2 Statistiky celosvětově

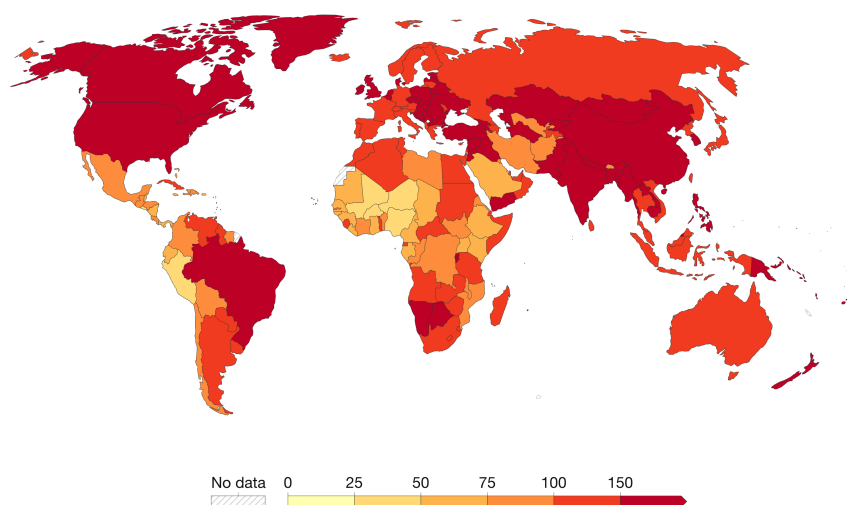
Podle rozsáhlé studie a analýzy detailně popsané v článku *Evolution of the Global smoking epidemic over the past half century: strengthening the evidence base for policy action* kouří na celém světě zhruba 32,6 % mužů a 6,5 % žen, což dohromady odpovídá 1,18 miliardám lidí. Od roku 1990 do roku 2020 byl zaznamenán významný pokles kuřáků. Procentuálně došlo k poklesu kouření u mužů o 27,2 % a u žen dokonce o 37,9 %. Nejvyšší podíl na poklesu mají především země s vysokými příjmy, kde byl celkový pokles až o 40 %. Naopak v zemích s nízkými nebo středními příjmy byl pokles poměrně malý nebo naopak dokonce došlo k nárůstu počtu kuřáků. [6]

2.2.1 Úmrtnost na kouření mezi lety 1990 až 2019

Death rate from smoking, 1990

Estimated annual number of deaths attributed to smoking¹ per 100,000 people.

Our World
in Data



Source: IHME, Global Burden of Disease (2019)

OurWorldInData.org/smoking • CC BY

Note: To allow comparisons between countries and over time this metric is age-standardized.

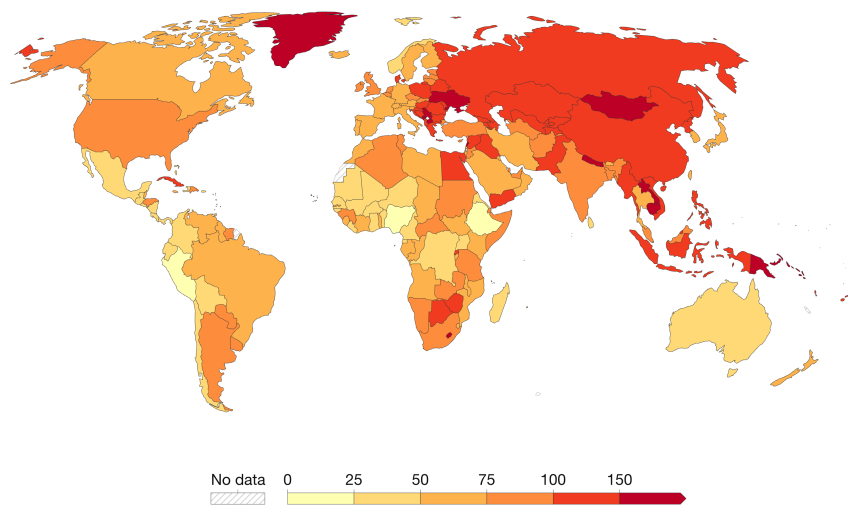
1. **Smoking:** Tobacco smoking is the practice of burning tobacco and ingesting the smoke that is produced. Smoking is a risk factor for many diseases including heart attacks, strokes and cancer. It is the leading risk factor for death in men, globally.

Obrázek 2.3: Úmrtnost na kouření v roce 1990, zdroj: [7]

Obrázek 2.3 graficky znázorňuje celosvětově průměrnou úmrtnost lidí na 100 000 obyvatel v roce 1990 v jednotlivých zemích. Nejvyšší úmrtnost byla zaznamenána v Kiribati (407,6 úmrtí na 100 000 obyvatel), druhá nejvyšší úmrtnost byla na Šalamounových ostrovech (395,3 úmrtí na 100 000 obyvatel), třetí nejvyšší úmrtnost měl Myanmar a čtvrtou Grónsko. Naopak celkově nejnižší úmrtnost byla v Nigeru a Peru (26,94 a 30,54 úmrtí na 100 000 obyvatel). Česká republika v roce 1990 byla na 20. místě s 211,73 úmrtími na 100 000 obyvatel.

Death rate from smoking, 2019

Estimated annual number of deaths attributed to smoking'per 100,000 people.



Source: IHME, Global Burden of Disease (2019)

OurWorldInData.org/smoking • CC BY

Note: To allow comparisons between countries and over time this metric is age-standardized.

1. **Smoking:** Tobacco smoking is the practice of burning tobacco and ingesting the smoke that is produced. Smoking is a risk factor for many diseases including heart attacks, strokes and cancer. It is the leading risk factor for death in men, globally.

Obrázek 2.4: Úmrtnost na kouření v roce 2019, zdroj: [7]

Na obrázku 2.4 je poté vidět, že mezi 1990 a 2019 došlo celosvětově k signifikantnímu snížení úmrtnosti. Ve 13 státech však byl trend opačný a naopak se v nich zvýšila úmrtnost ve spojení s kouřením. Nejvyšší nárůst zaznamenal Uzbekistán, kde došlo v období mezi lety 1990 až 2019 k relativnímu nárůstu o 34 % z 81,73 úmrtí na 109,83 úmrtí na 100 000 obyvatel. Nejmenší nárůst úmrtí zaznamenala Severní Korea, ve které došlo k nárůstu o pouhých 0,64 úmrtí na 100 000 obyvatel z celkových 137,95 úmrtí na 138,59 umrtí na 100 000 obyvatel. Naopak nejvyšší pokles úmrtí zaznamenal Singapur a Etiopie (o -72 % a -68 %). Česká republika se umístila na 28. místě s nejvyšším poklesem úmrtí s -54 % (z 211,73 na 96,38 úmrtí na 100 000 obyvatel).

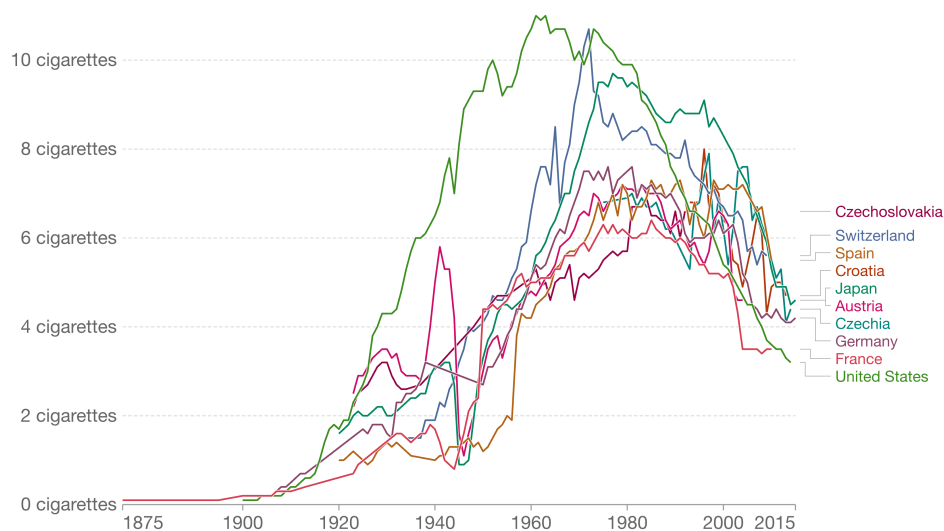
2.2.2 Počet prodaných cigaret na osobu za den

Jak již bylo zmíněno v kapitole Historie kouření, během 20. století kouření zaznamenalo signifikantní nárůst popularity následovaný poklesem během 60. let. Tento fakt přehledně ukazuje následující obrázek 2.5, na němž se nachází graf znázorňující průměrný počet prodaných cigaret dospělým lidem mezi léty 1875 až 2015 ve vybraných zemích s vysokým příjmem. [7]

Sales of cigarettes per adult per day, 1875 to 2015



Figures include manufactured cigarettes, as well as estimated number of hand-rolled cigarettes, per adult (ages 15+) per day.



Source: International Smoking Statistics (2017)

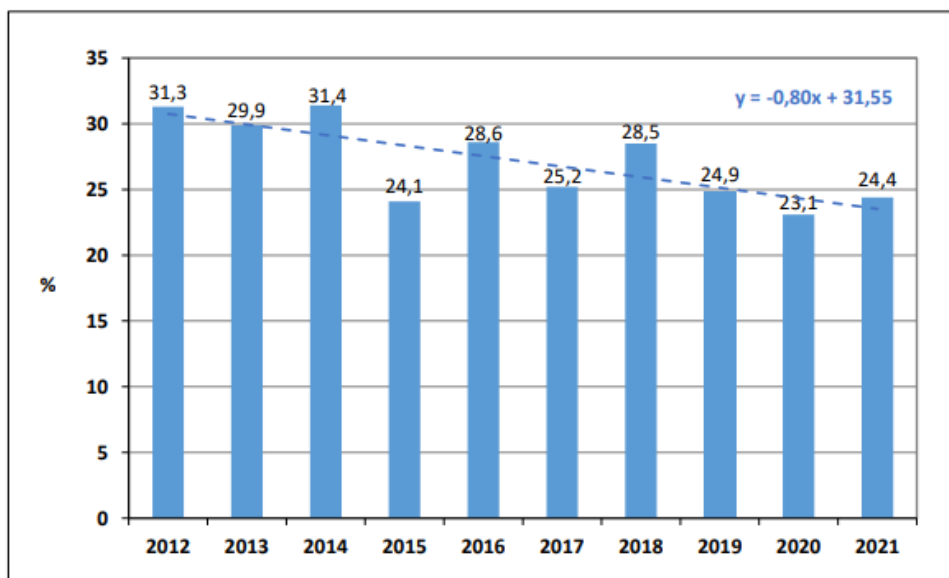
OurWorldInData.org/smoking • CC BY

Obrázek 2.5: Průměrný počet prodaných cigaret mezi lety 1875 až 2015 ve vybraných zemích, zdroj: [7]

Z grafu lze například vyčíst, že ačkoliv absolutní vrchol popularity prodeje cigaret ve Spojených státech amerických byl v roce 1963 (10,9 cigarety na osobu za den), ve Švýcarsku to bylo až v roce 1972 (10,7 cigarety na osobu za den). V Německu dokonce až v roce 1981 (7,6 cigarety na osobu za den). V tehdejší Československu byl vrchol prodeje cigaret v témže roce s průměrným počtem rovných 7 prodaných cigaret na osobu za den. V samostatné České republice pak největší popularita prodeje cigaret byla v roce 1997 a to se 7,9 cigaretami na osobu za den.

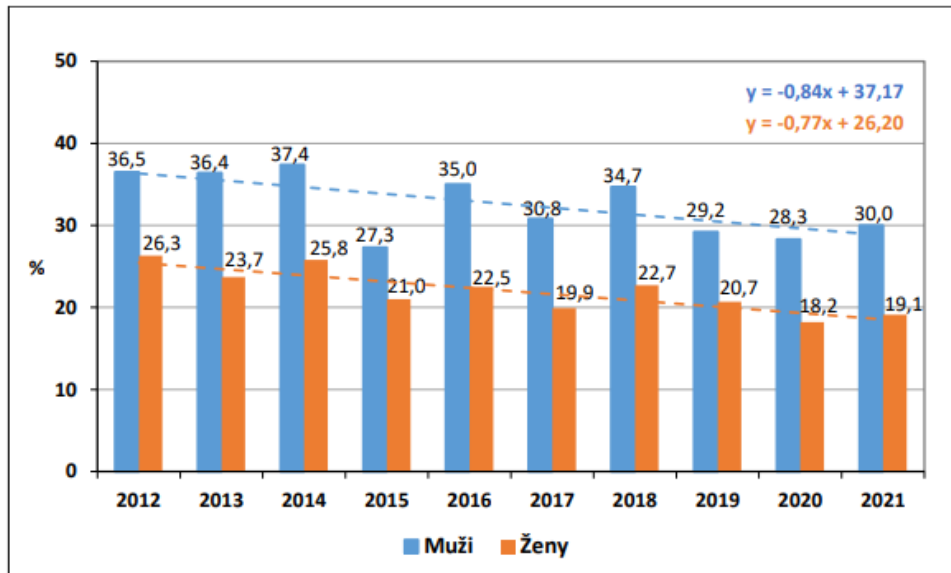
2.3 Statistiky v České republice

Z *Národního výzkumu užívání tabáku a alkoholu v České republice* (dále jako NAUTA) z roku 2021 vyplývá, že v České republice je zhruba 24,4 % všech obyvatel kuřáků (denních i příležitostných). V roce 2021 se jednalo o 23,1 %. Meziročně lze tedy sledovat mírný nárůst, který je však autorem studie zdůvodněn jako projev náhodného meziročního kolísání. Důležité je, že od roku 2012 lze pozorovat trvalý pokles celkového počtu kuřáků, viz 2.6.



Obrázek 2.6: Kuřáci tabákových výrobků celkem (denní a příležitostní) v České republice, zdroj: [11]

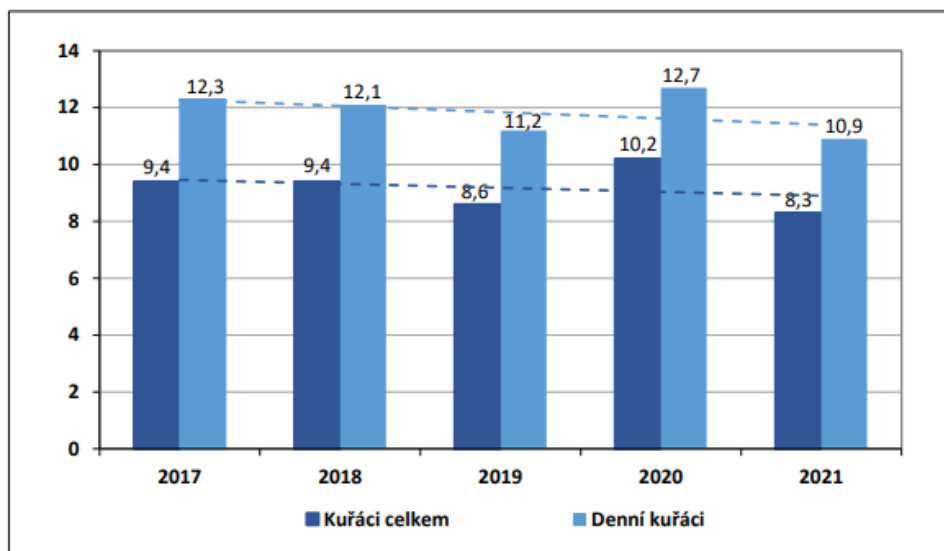
Podíl mužů a žen na celkovém počtu kuřáků je k roku 2021 přibližně 30 % mužů a 19,1 % žen. Tedy mužů, kteří jsou denními nebo příležitostnými kuřáky, je o 10,9 % více než žen. Tento rozdíl vyobrazuje graf na obrázku 2.7.



Obrázek 2.7: Kuřáci tabákových výrobků podle pohlaví (denní a příležitostní) v České republice, zdroj: [11]

Graf na obrázku 2.8 zobrazuje průměrné počty vykouřených cigaret za den v letech 2017-2021. Tento průměrný počet u denních kuřáků klesl od roku 2017 do roku 2021 o 1,4 cigarety za den na celkových 10,9 cigaret denně. Lze

si rovněž všimnout, že po započítání i příležitostných kuřáků toto číslo klesá na průměrných 8,3 vykouřených cigaret za den. Z dalších informací a dat dostupných z [11] je patrné, že je rozdíl mezi muži a ženami také v průměrném počtu vykouřených cigaret za den. V roce 2021 byl celkový počet vykouřených cigaret denně u mužů 9,4 a u žen 6,8 (zahrnuje denní i příležitostné kuřáky).



Obrázek 2.8: Počet kusů cigaret v průměru vykouřených za den v České republice, zdroj: [11]

2.4 Škodlivost kouření

Cigaretový kouř obsahuje tisíce škodlivin, mezi nimiž se vyskytují jedovaté látky, mutageny, kancerogeny nebo oxid uhelnatý. Kancerogenní látky způsobují vznik nádorů, kdežto oxid uhelnatý snižuje obsah kyslíku v krvi, čímž způsobuje nedostatečné prokrvení tkání. [53]

Nejčastěji je kouření spojované s onemocněními plic, a to zejména s nádory plic a výskytem chronické obstrukční plicní nemoci. Velkou problematikou jsou již zmíněné nádory. Z kouření mohou nádory kromě plic vznikat také v dutině ústní, v hltanu či jícnu, kde kouř přímo přichází do styku se sliznicemi. Kouření však může souviset i s výskytem rakoviny slinivky, tlustého střeva nebo močového měchýře. Nemálo je ovlivněn i kardiovaskulární systém. Kouření způsobuje poškození tepen a tím i zvýšenou krevní srážlivost, která může vést k infarktu myokardu či k mozkové mrtvici. Rovněž může dojít ke zvýšení krevního tlaku nebo ke vzniku srdečních arytmií. Kouřením je ovlivněn i celý náš metabolismus. Přispívá k rozvoji cukrovky a obezity a snížené mohou být i funkce imunitního systému. [52, 54]

Nově je znám také nepříznivý vliv na duševní zdraví. Kouření snižuje plodnost a v těhotenství významně ohrožuje plod. Jedná se o zvýšené riziko potratu nebo předčasného porodu, ale také častější výskyt vrozených vývojových vad. Obecně by se dalo říci, že kouření významně přispívá ke zkrácení

života, a to i až o 10-25 let. [52, 55]

2.5 Opatření pro redukci kouření

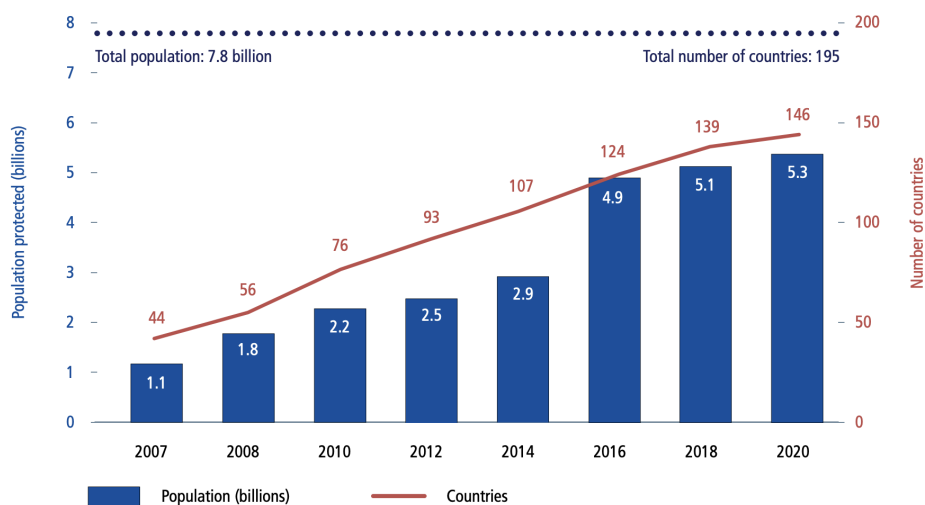
Vzhledem k obrovskému počtu kuřáků po celém světě je kouření často nazýváno globální epidemií. WHO na boj proti této epidemii ve snaze zabránit kouření jakožto vážné zdravotní hrozbě vydala soubor několika doporučení, jak spotřebu tabáku omezit. Tato doporučení se nazývají MPOWER a byla poprvé vydána v roce 2008 v dokumentu *WHO Report on the Global Tobacco Epidemic, 2008, The MPOWER package*.

Zkratka MPOWER vychází z těchto šesti bodů:

- **M**onitor tobacco use and prevention policies (monitorovat užívání tabáku a politiku prevence),
- **P**rotect people from tobacco smoke (chránit lidi před kouřením tabáku),
- **O**ffer help to quit tobacco use (nabídnout pomoc při odvykání kouření),
- **W**arn about the dangers of tobacco (varovat před nebezpečím užívání tabáku),
- **E**nforce bans on tobacco advertising, promotion and sponsorship (prosazovat zákazy reklamy na tabák a jeho propagaci a sponzorství), and
- **R**aise taxes on tobacco (zvyšovat daně na tabák).

[8]

Generální ředitel WHO, Dr. Tedros Adhanom Ghebreyesus, uvádí v reportu z roku 2021, že po celém světě je alespoň jedním bodem ze souboru doporučení MPOWER ovlivněno již 5,3 miliard lidí ze 75 % všech zemí. 50 % zemí má zavedené alespoň dva body. [9]



Obrázek 2.9: Zavedení alespoň jednoho bodu MPOWER v letech 2007-2020, zdroj: [9]

Z grafu 2.9 lze vyčíst, že mezi lety 2007 až 2020 došlo k vysokému nárůstu počtu zemí, které postupně aplikovaly alespoň jeden z bodů doporučení MPOWER. K největšímu nárůstu došlo mezi lety 2014 až 2016.

2.6 Výrobky určené ke kouření

Jestliže pomíneme kategorii výrobků určených ke kouření obsahující různé bylinné směsi, čaje a směsi obsahující kanabidiol (CBD), tak můžeme tyto výrobky rozdělit do dvou kategorií - **tabákové výrobky** a **alternativní nikotinové výrobky**. Podle vyhlášky č. 261/2016 Sb., o tabákových výrobcích se do kategorie **tabákových výrobků** řadí cigarety, doutníky, doutníčky a tabák, jež je k sehnání v několika formách - k ručnímu balení cigaret, dýmkový tabák, tabák určený do vodní dýmky, šňupací, žvýkáci a k orálnímu užití. Mezi všemi těmito formami drtivě převažuje užívání tabáku v podobě cigaret (komerčních i ručně balených). Mezi lety 2018 až 2020 sice došlo k poklesu podílu obou druhů cigaret z 98 % na 94 %, avšak používání ostatních forem jako jsou dýmky, doutníčky nebo vodní dýmky je stále velmi nízká. Přehled členění do skupin a podskupin dle vyhlášky č. 261/2016 Sb. z přílohy č. 1 lze sledovat na obrázku 2.10 [10]

Druh	Skupina	Podskupina
cigarety		
doutníky		
doutníčky		
tabák	ke kouření	určený k ručnímu balení cigaret
		dýmkový
	bezdýmný	určený do vodní dýmky
		šňupací
nový tabákový výrobek	ke kouření	žvýkací
	bezdýmný	určený k orálnímu užití

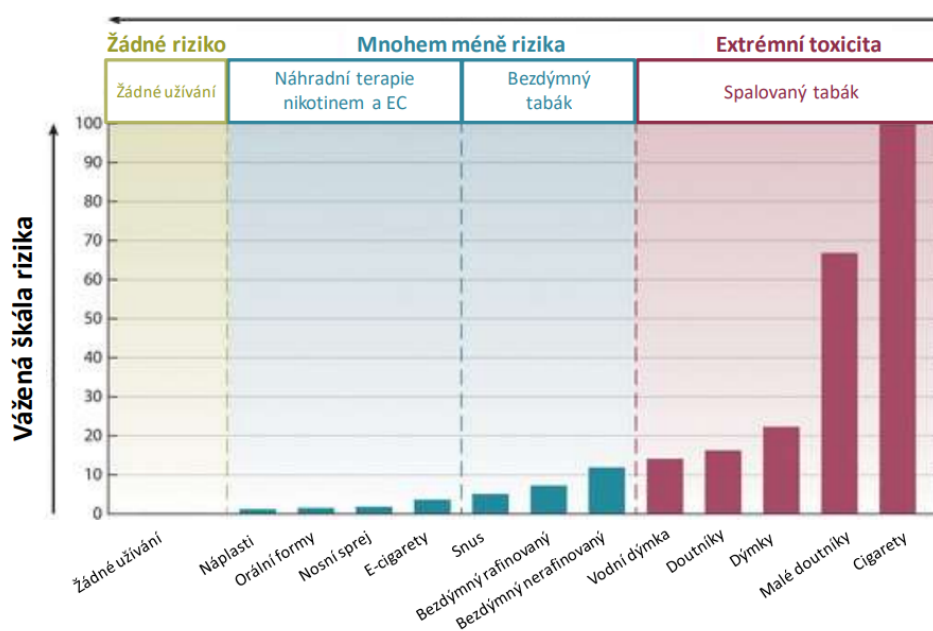
Obrázek 2.10: Druhy tabákových výrobků a jejich členění na skupiny a podskupiny, zdroj: [12]

Alternativní nikotinové výrobky, které obsahují nikotin, ale nedochází u nich ke spalování tabáku (tzv. Alternative Nicotine Delivery Systems, ANDS), lze rozdělit do dvou kategorií podle způsobu aplikace - **výrobky určené k inhalaci aerosolu** a **výrobky s jiným způsobem aplikace**. Do kategorie **výrobků určených k inhalaci aerosolu** se například řadí elektronické cigarety s nikotinem, též označovány jako ENDS (z anglického Electronic Nicotine Delivery Systems) nebo elektronické cigarety bez nikotinu, ENNDS (z anglického Electronic Non-Nicotine Delivery Systems). Do této kategorie řadíme i produkty označované jako HTP (z anglického Heated Tobacco Products), neboli výrobky s tabákem k zahřívání. Do druhé kategorie **výrobků s jiným způsobem aplikace** poté řadíme například nikotinové léčivé přípravky označované jako NRT (z anglického Nicotine Replacement Therapy), orální tabák, žvýkací tabák, šňupací tabák, případně náplasti, tablety, žvýkačky nebo orální spreje s obsahem nikotinu. Rozdělení alternativních nikotinových výrobků je vizuálně reprezentováno na obrázku 2.11. [10]

	Na bázi čistého nikotinu	Na bázi tabáku
Nezahřívané	náhradní terapie nikotinem (náplasti, žvýkačky, pastilky, ústní/nosní sprej), nikotinové sáčky	bezdýmný tabák porcovaný tabák, snus
Zahřívané	elektronické cigarety (vapování liquidů na bázi vody/alkoholu)	zahříváný tabák HTP - Heated Tobacco Products (IQOS, GLO)

Obrázek 2.11: Rozdělení alternativních nikotinových výrobků, převzato z [10], původní zdroj: Společnost pro léčbu závislosti na tabáku

Cigarety jsou pro kuřáky ze zdravotního hlediska nejvíce rizikovou formou konzumace nikotinových produktů. Nikotin sám o sobě totiž není tak škodlivý, jako tisíce látek obsažených v tabákovém dehtu, který vzniká při spalování a je inhalován spolu s kouřem do plic. Míru rizika vyobrazuje graf na obrázku 2.12, kde je vidno, že cigarety jsou zdaleka nejtoxičtější způsobem konzumace nikotinových produktů. Druhým nejnebezpečnějším produktem po cigaretách jsou malé doutníčky, též známé jako *cigarillos*. Oproti klasickým doutníkům se liší především formou konzumace. Zatímco klasické doutníčky se konzumují pomocí nasávání kouře do dutiny ústní, cigarety a malé doutníčky se tzv. šlukují, tedy kouř se vdechuje až do plic. Naopak nejnižší toxicitu podle grafu na obrázku 2.12 vykazují výrobky, které slouží pro náhradní terapii, jako jsou náplasti, orální formy nebo nosní spreje. Stále je však třeba mít na paměti, že se jedná o pouhý odhad vycházející z dosud známých vědeckých studií. O škodlivosti novějších produktů, z nichž můžeme jmenovat nahřívané tabáky, e-cigarety nebo nikotinové sáčky, se toho ví stále málo. Obecně se má za to, že při konzumaci e-cigaret je až 200krát nižší riziko výskytu rakoviny způsobené kouřením, nicméně se neví, jak škodlivá může být inhalace samotných příchutí, které e-cigarety obsahují. [10]



Obrázek 2.12: Míra rizika - rizikové kontinuum různých tabákových a nikotinových výrobků převzato z: [10], původní zdroj: Abrams et al. (2018) podle Nutt a kol. (2014)

2.7 Používání tabákových a nikotinových výrobků mezi mladistvými

ESPAD (European Survey Project on Alcohol and Other Drugs) v nejnovějším reportu z roku 2019 uvádí řadu statistik k používání tabáku mladistvými. V

průměru 60 % dotazovaných studentů uvádí, že by pro ně bylo jednoduché získat cigarety, pokud by o to stáli. Nejvyšší dostupnost byla zaznamenána v Dánsku se 79 %. V České republice to bylo přes 70 %. Naopak nejnižší dostupnost byla v Kosovu (24 %). Co se týče pohlaví, rozdíly byly minimální. Ve většině zemí měli však chlapci mírně větší přístup k cigaretám než dívky. 18 % studentů začalo kouřit cigarety ve věku 13 let nebo dříve. Nejvyšší podíl byl zaznamenán v Lotyšsku a Litvě (přes 30 %). Průměrná míra studentů, kteří začali denně kouřit před dovršením 13 let, byla 2,9 %. Více než jeden z 10 studentů uvedl, že použil elektronickou cigaretu do 13 let věku. Nejvyšší podíly byly v Litvě a Estonsku (přes 19 %). Denní používání elektronických cigaret před dovršením 13 let uvádělo průměrně 1,7 % studentů, a to s nejvyšší mírou v Kosovu, na Kypru, na Slovensku a na Ukrajině. [56]

2.8 Metody odvykání kouření

Metod odvykání kouření v dnešní době existuje celá řada, některé méně, některé více efektivní. V této části si zmíníme v souvislosti s odkvykáním kouření tři metody. Podíváme se na NRT (Nicotine Replacement Therapy) jako nejběžnější metodu v souvislosti s odvykáním kouření, dále na E-Health, metodu, která je předmětem zkoumání, ale je často skloňována pod pojmy jako revoluce ve zdravotnictví a v budoucích letech se očekává její vzestup. A poslední metodou je metoda experimentální, kterou jsou nikotinové vakcíny, jejichž budoucnost je zcela nejistá.

2.8.1 Nicotine Replacement Therapy

Jedná se o nejběžnější metodu odvykání, kdy kuřák používá alternativní výrobky s obsahem nikotinu. Takovými výrobky jsou například náplasti, nosní nebo orální spreje, žvýkačky nebo tablety. Ty fungují na principu postupného uvolňování nikotinu do těla a tím snižují chuť na cigaretu. Například nikotinové náplasti člověk střídá každý den a nosí je po dobu 24 hodin. Vedlejším účinkem náplastí však mohou být nepříjemné sny, v takovém případě se nedoporučuje náplasti nosit i přes noc. Nikotinové žvýkačky se doporučují žvýkat v množství 1-2 kusů za hodinu pro lidi, kteří s odvykáním začínají, a množství postupně snižovat. Odvykající by neměl přesáhnout 20 žvýkaček denně. Cílem je omezit počet žvýkaček během 12 týdnů na nulu. Jedná se však o obecná doporučení a konkrétní použití jednotlivých přípravků vždy podléhá příbalovému letáku. Kombinace dvou forem NRT může odvykajícímu též pomoci se snížením chuti na cigaretu. Některé přípravky NRT mohou být v určitých zemích dotované nebo jiným způsobem na doporučení lékaře podporované. [57, 58]

2.8.2 E-Health

V souvislosti s příchodem moderních technologií a vysokou dostupností mobilních telefonů a počítačů je v současné době na vzestupu takzvaný E-Health

(elektronické zdravotnictví), tedy použití informačních technologií ve zdravotnickém sektoru pro účely léčby a podpory zdraví. S tím souvisí i využití této metody pro odvykání kouření. Protože jsou tradiční terapeutické a poradenské služby často finančně i časově nákladné, jedná se o skvělou a levnou alternativu. Odvykání kouření je tak v dnešní době možné snadno podpořit tvorbou různých interaktivních webových stránek nebo mobilních aplikací. V minulosti bylo elektronické zdravotnictví předmětem zkoumání mnoha studií a bylo zjištěno, že tento přístup v souvislosti s odvykáním kouření může zvyšovat pravděpodobnost úspěšné abstinence. Vývoj mobilní aplikace Buddy, která je předmětem této diplomové práce, spadá právě pod kategorii E-Health. [60]

■ 2.8.3 Nikotinové vakcíny

Jedná se o experimentální způsob léčby závislosti na nikotinu, který je stále předmětem zkoumání. Nikotinové vakcíny fungují na principu spojení nikotinové molekuly s nosným proteinem, který v těle člověka vyvolává imunitní odpověď. Imunitní systém rozpozná nikotin jako cizorodou látku a spustí proti ní reakci. Jedním z problémů spojených s touto metodou je kompenzace nízkého účinku nikotinu s větší spotřebou tabáku. Potenciálním problémem je také krátká životnost vakcíny. Nejznámější výrobek tohoto typu je NicVAX od společnosti Nabi Bipharmaceuticals. [59]

Kapitola 3

Metody vývoje mobilních aplikací

V této kapitole se seznámíme s trendy ve vývoji mobilních aplikací. Podíváme se na aktuálně nejpožívanější frameworky a technologie a popíšeme rozdíly mezi nativním a multiplatformním vývojem.

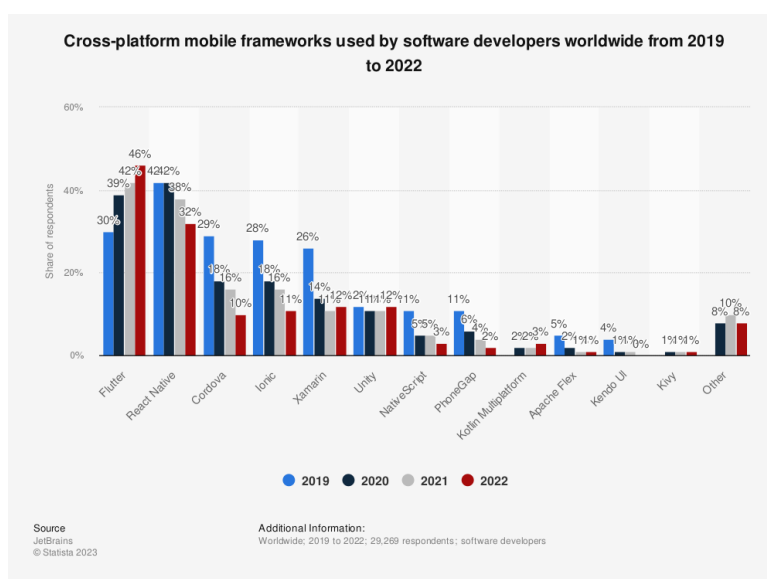
3.0.1 Trendy vývoje mobilních aplikací

V současné době celosvětově sledujeme nárůst počtu vývojářů aplikací. V roce 2023 se tento počet odhaduje na 27,7 milionů. Predikce pro rok 2024 jsou dokonce 28,7 milionů. [14] Celkový počet vývojářů mobilních aplikací se odhaduje na 8,7 milionů, z toho 5,9 milionů vývojářů pro Android a 2,8 milionů pro iOS. [15] Masivní nárůst též potvrzují statistiky a odhady, které udávají, že v roce 2023 byl roční výnos z mobilních aplikací celosvětově 467 milionů dolarů. Očekává se, že v roce 2024 to bude zhruba 522 milionů dolarů a odhady na rok 2027 udávají dokonce 673 milionů dolarů. [16]

3.0.2 Nativní vs multiplatformní vývoj aplikací

V současné době je drtivá většina mobilních aplikací vyvíjená na mobilní zařízení s operačními systémy Android a iOS. Vývojáři mají dnes na výběr, zda využijí technologie nativní nebo multiplatformní. Nativní technologie disponují lepší výkonností, bezpečností a optimalizovatelností. Nevýhoda však spočívá ve vysokých nákladech na vývoj. Multiplatformní technologie na druhou stranu umožňuje levný vývoj a možnost přepoužitelnosti programovacího kódu napříč mobilními operačními systémy. Jejich nevýhoda však spočívá v časté potřebě aplikaci spouštět přes dodatečné runtime ¹ rozhraní, čímž se vytváří další vrstva, což posléze způsobuje nižší výkon aplikací. Zároveň je nutné myslet na specifické implementační detaily a že některé hlubší problémy je tak nutné řešit pro každý operační systém zvlášť. U multiplatformních aplikací je též mnohem náročnější vývoj specifických funkcionalit požadující integraci s hardwarovým rozhraním. V zásadě nelze obecně říci, který z těchto přístupů je vhodnější, a vždy je před zahájením vývoje mobilní aplikace zásadní analyzovat její specifické vlastnosti, funkcionality a chování a na jejich základě poté vyhodnotit volbu vhodné technologie. [17]

¹Runtime - označení doby běhu počítačového programu



Obrázek 3.1: Popularita mobilních frameworků v letech 2019-2022, zdroj: [19]

Jak je patrné z grafického znázornění na obrázku 3.1, distribuce popularity multiplatformních vývojových prostředí pro mobilní aplikace v roce 2022 ukazuje, že framework *Flutter* dominuje s podílem 46 %. Následuje *React Native* s 32% podílem, což ho činí druhým nejpreferovanějším frameworkem. Další v pořadí jsou *Xamarin* s 12%, *Ionic* s 11% a *Cordova* s 10% podílem na trhu. Tato data reflektují aktuální trend v oblasti vývoje multiplatformních mobilních aplikací, kde je zřejmá rostoucí preference pro framework *Flutter*, jež byl ještě v roce 2020 na druhém místě za frameworkem *React Native*. Naopak trend vykazuje známky poklesu popularity frameworku *React Native*, který v roce 2019 dominoval na prvním místě se 42 % o 12 % před frameworkem *Flutter*.

Flutter

První stabilní verze frameworku *Flutter* vyšla v prosinci 2018. Tento framework je postavený na programovacím jazyce Dart od společnosti Google. Flutter poskytuje bohatou sadu předdefinovaných widgetů, ze kterých lze snadno vyvíjet aplikaci bez nutnosti vytvářet vlastní komponenty. Flutter je též postavený na vlastním unikátním vykreslovacím enginu, který je psaný v jazycích C/C++. Při vývoji aplikace Flutter poskytuje vývojářům just-in-time kompilaci pro snadné testování a vizualizaci změn. V produkčních buildech Flutter uplatňuje kompilaci ahead-of-time, díky které dosahuje optimalizovaného výkonu, často velmi blízkého výkonu nativně vyvíjených aplikací. [43]

React Native

Framework *React Native* je vytvořený společností Facebook cílící na vývoj pro více platform zároveň. Využívá jazyky Javascript a Typescript a frameworku



Část II

Realizační část

Kapitola 4

Analýza výchozího stavu aplikace Buddy

V této kapitole se budeme zabývat analýzou stávajícího řešení mobilní aplikace Buddy. Nejprve zařadíme aplikaci Buddy do kontextu odvykacího plánu kuřáka. Poté si popíšeme jednotlivé obrazovky a funkcionality aplikace. Nakonec provedeme rozbor z technického pohledu. Vytvoříme výčet hojně využívaných technologií a knihoven a podíváme se na jejich konkrétní využití. Nakonec popíšeme existující řešení z technického pohledu, kdy se podíváme na využívané knihovny a technologie.

4.1 Mobilní aplikace pro odvykání kouření

Kuřák pro účely odvykání používá existující řešení Mobilní aplikace pro odvykání kouření. Tato aplikace byla navržena odborníky z Kliniky adiktologie 1. lékařské fakulty Univerzity Karlovy. Kuřáka provází řadou automatizovaných sezení, ve kterých se nejprve seznamuje se základními informacemi a průběhem odvykání a poté jej tato aplikace navádí, jak se správně v určitých situacích chovat a jak přemýšlet, aby eliminoval riziko neúspěšného odvykání. Kuřák při používání aplikace také sbírá odznaky jako stvrzení dosažení určitých výsledků, čímž je motivován.

4.2 Mobilní aplikace Buddy

Smyslem mobilní aplikace Buddy je zajištění podpory známých, přátel a rodiny pro kuřáka, který se rozhodl podstoupit digitální terapii pro odvykání kouření. Tato práce vychází z existujícího řešení mobilní aplikace Buddy. Aplikace prošla v minulosti již několika iteracemi vývoje z řad studentů v rámci semestrálních projektů a závěrečných prací a nyní se nachází ve stavu funkčního prototypu. Aplikace disponuje řadou implementovaných funkcionalit, na které v návrhové fázi navážeme.

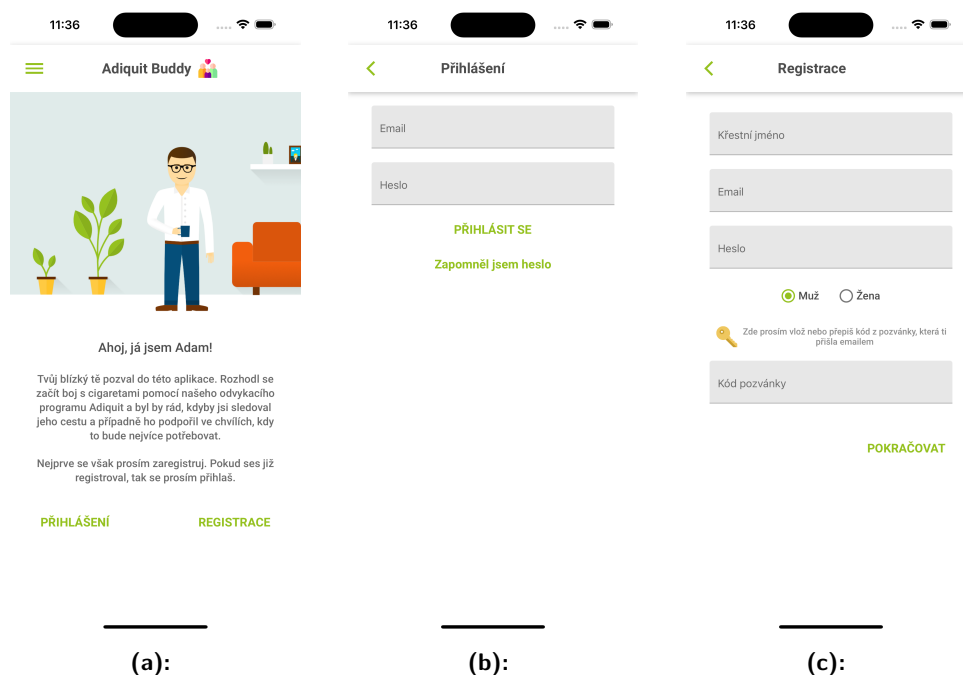
4.3 Popis jednotlivých funkcionalit

V této části analytické kapitoly bude předmětem zkoumání popis vlastností a schopností aplikace Buddy v jejím výchozím provedení. Detailně prozkoumáme

jednotlivé komponenty aplikace. Tato analýza bude následně využita pro návrh nových vylepšení nebo modifikací stávajících prvků aplikace.

4.3.1 Registrace a přihlášení

Při prvním spuštění aplikace Buddy je uživateli prezentována úvodní stránka, která obsahuje popis zaměření aplikace Buddy. Uživateli je vysvětleno, že byl pozván svým blízkým k používání této aplikace, a je vyzván k přihlášení nebo registraci. Zelený text PŘIHLÁŠENÍ a REGISTRACE slouží jako tlačítka a po jejich stisknutí je uživatel přesměrován na příslušné formuláře.



Obrázek 4.1: (a) Úvítací stránka, (b) Přihlašovací formulář, (c) Registrační formulář zdroj: Autor - Aplikace Buddy

Obrázek 4.1b znázorňuje standardní přihlašovací formulář s možností přihlásit uživatele nebo obnovit zapomenuté heslo. Registrační formulář znázorněný na obrázku 4.1c vedle standardních polí pro registraci, jako je křestní jméno, email, heslo a výběr pohlaví, obsahuje také pole pro zadání kódu pozvánky. Kód pozvánky je unikátní kód, který kuřák posílá svým známým, čímž je zve k používání aplikace Buddy. Bez tohoto kódu registraci nelze provést.

4.3.2 Onboarding

Po přihlášení uživatele do aplikace a ještě před zobrazením hlavní stránky předchází takzvaný onboarding. Smysl onboardingu je seznámení uživatele s principy a fungováním aplikace. K uživateli promlouvá textovou formou virtuální terapeut Adam, který se v prvním kroce představí a vysvětlí uživateli důležitost jeho role pro kuřáka v celém procesu odvykání. Onboarding se

skládá ze tří po sobě jdoucích obrazovek, mezi kterými se uživatel pohybuje pomocí tlačítek „další“ a „zpět“. Na třetí stránce v pořadí je poté tlačítko „další“ nahrazeno tlačítkem „dokončit“, což potvrzuje ukončení onboardingu uživatele. Úspěšné dokončení onboardingu je poté v aplikaci uloženo na lokálním uložišti zařízení a uživateli při opakovaném otevření aplikace není znovu nabízen. Uživatel může následně kdykoliv během používání aplikace v hamburger menu zvolit možnost „Opakovat onboarding“, což spustí tento proces znovu. To slouží pro případ, že by si uživatel potřeboval onboarding zopakovat z důvodu nejasností během používání aplikace.

4.3.3 Dashboard a fáze

Po přihlášení uživatele do aplikace a případného projití onboardingu se uživatel dostane na dashboard (hlavní stránku). Tato stránka obsahuje několik prvků a komponent, které se zobrazují v posloupnosti vertikálně pod sebou. Hlavní stránku lze tedy posouvat směrem dolů. Nejprve se v horní části nachází text, který stručně informuje uživatele v jedné větě o fázi, v níž se současně odvykající kuřák nachází. Pod tímto textem se nachází tlačítko „Ukázat cestu“, které vede na obrazovku s odznaky a procentuálním vyobrazením postupu kuřáka fázemi. Více kapitola 4.3.4.

V případě, že uživatel má dostupná ještě některá sezení, která neabsolvoval, a zároveň kuřák došel do fáze, ve které se tato sezení zpřístupní, následuje widget s názvem sezení, výzvou k absolvování a tlačítkem „Jdeme na to“. To uživatele přenesse na obrazovku sezení.

Další částí hlavní stránky je roadmapa. Jedná se především o grafický element tvořený z řady barevných obrázků. Roadmapa je tvořena cestou, která se postupně vyplňuje modrou barvou podle toho, jak kuřák postupuje fázemi odvykání. Na cestě se nachází vyobrazené jednotlivé fáze v podobě tlačítek. Po jejich rozkliknutí se uživateli zobrazí detailní popis vybrané fáze.

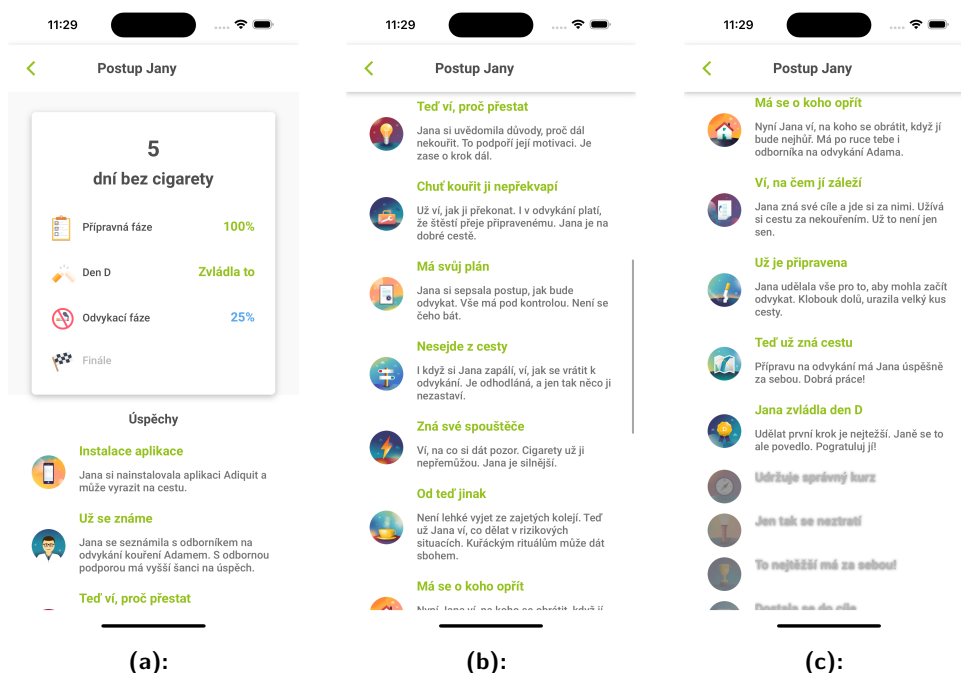
Spodní část hlavní obrazovky je tvořena několika widgety¹. Jsou to widget s ukazatelem ušetřených peněz, widget pro zopakování sezení, widget pro podporu kuřáka a widget s reporty a statistikami. Jednotlivé widgety jsou popsány v kapitolách 4.3.5, 4.3.6, 4.3.7 a 4.3.8.

4.3.4 Ukázat cestu a odznaky

Tlačítko „Ukázat cestu“ přesměruje uživatele z hlavní obrazovky na obrazovku obsahující dvě části. První z nich je menší widget s vyobrazeným procentuálním postupem fázemi odvykání. Druhá část zobrazuje úspěchy kuřáka. Úspěchy reprezentují kontrolní body, kterými kuřák při procesu odvykání prochází. Jakmile některým bodem projde, získá ocenění, které se poté zobrazí čitelným a obarveným textem. Úspěchy, které kuřák ještě nezískal, jsou rozmazané a vyobrazené šedou barvou - viz obrázky 4.2a 4.2b a 4.2c.

¹Widget - uživatelská a grafická komponenta, která slučuje více grafických prvků dohromady a tvoří tématicky ucelený element

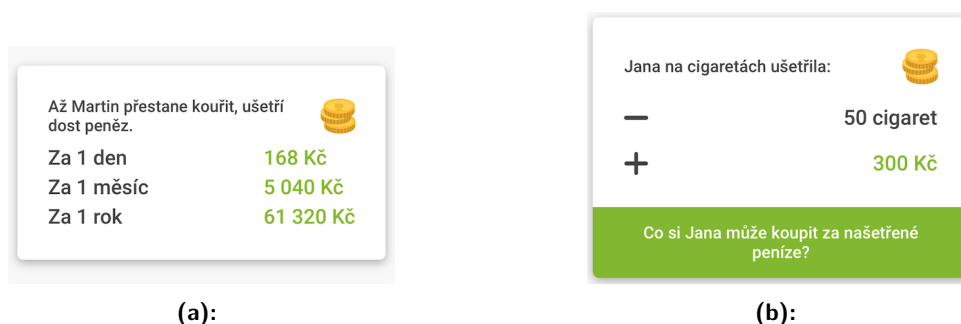
4. Analýza výchozího stavu aplikace Buddy



Obrázek 4.2: Obrazovka cesta a úspěchy (a) 1. část, (b) 2. část, (c) 3. část, zdroj: Autor - Aplikace Buddy

4.3.5 Widget s ušetřenými penězi a obrazovka přání

Widget s ušetřenými penězi se nachází ve spodní části hlavní obrazovky a zobrazuje se uživateli ve dvou různých variantách podle toho, v jaké fázi se kuřák nachází. Pokud je kuřák ve fázi přípravy na odvykání a samotné odvykání ještě nezačalo, zobrazuje se varianta na obrázku 4.3a. Ta vyobrazuje odhad, kolik kuřák ušetří za jednotlivé časové úseky - den, měsíc a rok. Pokud bylo odvykání již zahájeno, změní se widget na počítadlo ušetřených peněz za cigarety - obrázek 4.3b. Tato komponenta počítá s cenou za krabičku cigaret, kterou kuřák zadává v rámci sezení v přípravné fázi v Mobilní aplikaci pro odvykání kouření.



Obrázek 4.3: Widget s ušetřenými penězi (a) přípravná fáze, (b) odvykáací fáze, zdroj: Autor - Aplikace Buddy

Varianta na obrázku 4.3b má navíc podpurný text, který slouží jako tlačítko,

které uživatele přeneše na obrazovku se seznamem přání. Na této obrazovce se nachází seznam předdefinovaných přání, které reprezentují běžné odměny. Jedná se o:

- „oběd v dobré restauraci“,
- „měsíční permanentka do fitka“,
- „1 hodina v privátním wellness“,
- „prodloužený víkend na horách“,
- „týdenní dovolená u moře“.

Ve spodní části obrazovky je tlačítko „přidat přání +“ umožňující uživateli přidat do seznamu jakékoliv vlastní přání navíc.

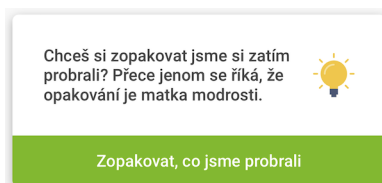
4.3.6 Widget a seznam sezení

Sezení jsou velmi významnou součástí aplikace Buddy. Jejich cílem je postupně informovat uživatele v 8 po sobě navazujících sezeních o důležitých informacích spojených s procesem odvykání kuřáka. Forma těchto sezení je opět založena na vysvětlování a popisu ze strany virtuálního terapeuta Adama, který vysvětluje jednotlivá témata.

Sezení jsou následující:

- „**Jak to bude probíhat?**“ - vysvětlení plánu odvykání,
- „**Výhody života bez cigaret**“ - probírá se, co život bez cigaret kuřákovi přinese,
- „**Nikotinová závislost**“ - sezení, při kterém Adam vysvětluje, jak funguje nikotinová závislost, a zároveň je uživatel podroben Fagerströmovu testu nikotinové závislosti,
- „**Abstinenční příznaky**“ - popis abstinenčních příznaků v průběhu odvykání,
- „**Podpora**“ - vysvětlení významu a role uživatele pro odvykajícího kuřáka a popis, jak kuřáka podporovat,
- „**Zvládání chuti na cigaretu**“ - ukázání tipů uživateli, jak radit kuřákovi zahnat akutní chuť na cigaretu,
- „**Náhradní nikotinové přípravky**“ - seznámení s různými přípravky, které mohou kuřákovi pomoci při odvykání,
- „**Zmírnění abstinenčních příznaků**“ - sezení, při kterém uživatel vybere příznaky, které kuřáka tíží. Terapeut Adam poté tyto příznaky rozebere detailněji i s doporučením, jak je snížit.

Uživatel má také možnost prohlédnout si kompletní seznam sezení i s jejich příslušnými stavy. Tento seznam se nachází na obrazovce, na niž se uživatel dostane po kliknutí na podpůrný text u widgetu (obrázek 4.4), který vyzývá uživatele k zopakování si toho, co již bylo probráno.



Obrázek 4.4: Widget sezení, zdroj: Autor - Aplikace Buddy

Fagerströmův test nikotinové závislosti

Součástí třetího sezení „Nikotinová závislost“ je takzvaný Fageströmův test nikotinové závislosti. Účelem testu je zjištění úrovně závislosti kuřáka na nikotinu na základě šesti dotazníkových otázek. Každá odpověď je bodově ohodnocena. Výsledkem je součet bodů dle konkrétních odpovědí, který určuje výši závislosti kuřáka na nikotinu. Hodnocení závislosti je následující. 7-10 bodů odpovídá silné závislosti, 4-6 bodů střední závislosti a méně než 4 body slabé závislosti. [20]



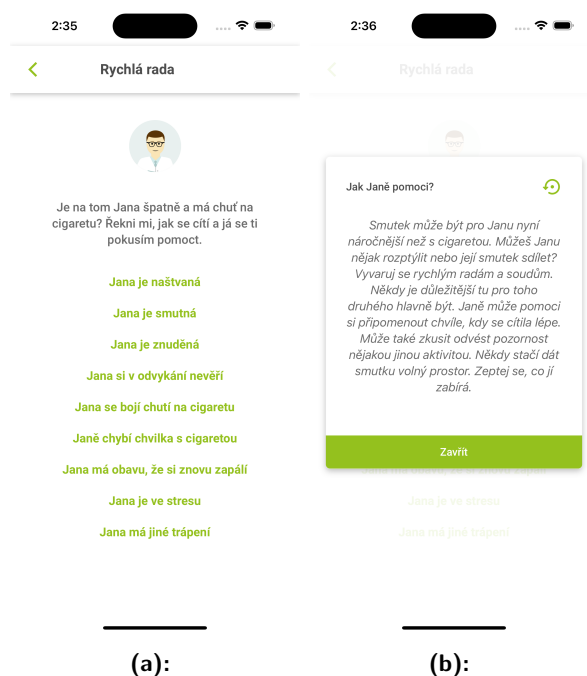
Obrázek 4.5: (a) První otázka, (b) Druhá otázka, (c) Čtvrtá otázka zdroj: Autor - Aplikace Buddy

Na obrázcích 4.5a, 4.5b, 4.5c si lze prohlédnout implementované obrazovky s dotazníkovými otázkami přímo v aplikaci Buddy. Jednotlivé formulářové

obrazovky jsou v navazující posloupnosti za sebou a uživatel se jimi posouvá výběrem odpovědi na otázku.

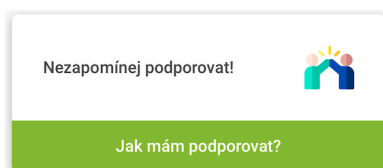
4.3.7 Widget podpory a rychlá rada

Na obrázku 4.7 je znázorněný widget podpory. Tento widget obsahuje pouze stručný text „Nezapomínej podporovat!“ a podpůrný text „Jak mám podporovat?“, který uživatele přenesou na obrazovku podpory. Zde je popsáno, jak se má uživatel zachovat, pokud chce kuřáka aktivně podpořit. Ve spodní části obrazovky je tlačítko „Rychlá rada“, které je aktivní ve chvíli, kdy kuřák přejde do fáze odvykání. Do té doby je tlačítko zašedlé a nachází se u něj upozornění, že rychlá rada se zpřístupní, až kuřák začne s odvykáním. Tlačítko „Rychlá rada“ přenesou uživatele na obrazovku, kde je seznam nejčastějších negativních pocitů, se kterými se kuřák může na své cestě při odvykání setkat.



Obrázek 4.6: (a) Obrazovka rychlé rady, (b) Rada pokud je kuřák smutný, Autor - Aplikace Buddy

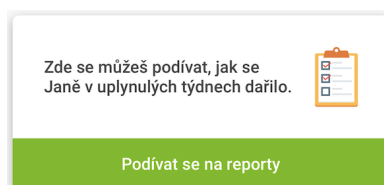
Po výběru jedné z těchto emocí je uživateli poskytnuta rada, jak se zachovat a kuřáka v této situaci podpořit. Každá emoce se váže na více variant rad a ty jsou zobrazovány náhodným výběrem.



Obrázek 4.7: Widget podpory, zdroj: Autor - Aplikace Buddy

4.3.8 Widget s reporty a seznam reportů

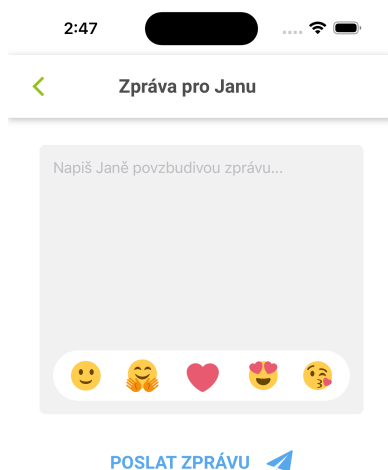
Nejnižší na hlavní obrazovce se nachází widget s jednotlivými týdenními reporty (obrázek 4.8). Podpůrný text „Podívat se na reporty“ přesměruje uživatele na obrazovku se seznamem reportů. Obrazovka disponuje horizontálním posunem mezi reporty a je řazena vzestupně. Na předposlední pozici zprava je zobrazován nejaktuálnější report z posledního týdne. Na poslední pozici je poté upozornění, které udává, kolik dní zbývá do vygenerování nového reportu.



Obrázek 4.8: Widget s reporty, zdroj: Autor - Aplikace Buddy

4.3.9 Posílání zpráv

V rámci průchodu aplikací Buddy se uživatel setká s řadou možností, jak odeslat kuřákovi zprávu. Nejvýznamnější a nejintuitivnější možností je kliknutí na tlačítko „Poslat <jméno kuřáka> zprávu“. To se nachází na hlavní obrazovce s pevnou pozicí vpravo dole. Uživatel je však vyzván ke stejné akci poslání zprávy i v dalším případě. Stejně tlačítko, akorát v modifikovaném znění, se nachází pod nejaktuálnějším reportem s textací "Pochválit <jméno kuřáka>".



Obrázek 4.9: Obrazovka pro odeslání zprávy, zdroj: Autor - Aplikace Buddy

Na obrazovce pro odeslání zprávy je poté pole pro vložení textu a předpřipravený seznam vybraných emotikonů.

4.4 Popis technické části

React Native

Mobilní aplikace Buddy je vyvíjena pomocí multiplatformní technologie React Native a využívá platformy Expo. V poslední verzi aplikace je využíván React Native verze 0.68.2. Aplikace je psaná v programovacím jazyce Javascript. Aplikace též využívá nástroje ESLint verze 6.8.0 pro statickou analýzu kódu.

React Hooks

V aplikaci se hojně využívá funkcionalit sady nástrojů React Hooks. Tyto nástroje významně usnadňují vývoj aplikace v prostředí React Native. Jednou z často používaných React Hooks je metoda `useEffect()`. Ta slouží k vytvoření akce ihned po vyrenderování komponenty. Příklad využití v aplikaci Buddy je komponenta **Notification**, kde byla využita pro správu délky zobrazení notifikace. Dalším příkladem může být použití React Hook `useState()`, který

přidává funkcionálním komponentám stav. Příkladem využití v aplikaci Buddy je komponenta **ScrollViewWithRefresh**, která jej využívá pro uložení informace o tom, zda má uživateli nabídnout možnost znovunačtení obrazovky či nikoliv. [28, 29, 45]

■ Expo

Aplikace Buddy je postavená na platformě Expo, která je založena na frameworku React Native a usnadňuje vývoj a nasazení mobilních aplikací pro platformy iOS a Android. Expo poskytuje sadu nástrojů a služeb, které umožňují vývojářům vytvářet aplikace v JavaScriptu a React, zatímco Expo se stará o kompilaci do nativního kódu pro každou z platforem. Pro efektivní a rychlé testování Expo nabízí mobilní aplikaci Expo Go, umožňující spouštění a testování vyvíjené aplikace přímo na mobilním zařízení během vývoje bez potřeby předchozího sestavení nativního kódu. [46]

■ Async Storage

Při předchozím vývoji aplikace Buddy pro účely lokálního uložení byla využita knihovna Async Storage, která je součástí balíčků frameworku React Native. Účelem této technologie je poskytnout vývojářům rozhraní řešení lokálního uložení na mobilních zařízeních. Async Storage poskytuje přístup k datům kdekoli v rámci mobilní aplikace. Funguje asynchronně, tedy čtení ani zápis neblokuje vlákno prezentační vrstvy. Data ukládá na principu klíč-hodnota. Nejedná se o šifrované uložení, není tedy bezpečné skrzes tuto formu uložení ukládat lokálně citlivá data. V aplikaci Buddy je toto uložení například využíváno pro ukládání vlastních přání nebo absolvovaných sezení uživatele. [47]

■ i18next a czech-inflection

Aplikace Buddy využívá funkcí internacionalizační knihovny i18next, která slouží pro vizualizaci textů v různých jazykových mutacích. Tato knihovna je využívána především v kombinaci s knihovnou czech-inflection, vyvinutou bývalým studentem Erikem Cupalem pro účely aplikace Buddy. Ta obohacuje aplikaci o možnosti skloňování českých jmen a přechylování českých slov. Při tvorbě statických textů je poté možné kuřáka nebo uživatele reprezentovat konstrukty jako například `'smoker5p'`, který je ve výsledném textu nahrazen jménem kuřáka v 5. pádě. [48, 49]

■ Ramda

Často se v programovacím kódu aplikace Buddy také vyskutují nástroje z knihovny Ramda. Jedná se o Javascriptovou knihovnu, která podporuje čistě funkcionální přístup k vývoji a nabízí širokou paletu funkcí. Častým výskytem využití rozhraní knihovny Ramda v aplikaci Buddy je funkce **pipe()**. Ta nabízí vývojáři rozhraní pro řetězení funkcí. Příkladem využití přímo z aplikace

Buddy je ukázka ze souboru **deriveCurrentBuddySession.js**, kde byla využita pro návrat následujícího nesplněného sezení uživatele.

```
const nextSession = pipe(
  filter(session => !finishedBuddySessions[session]),
  head,
)(BUDDY_SESSIONS_ALL);
```

Zde předchozí vývojář aplikace Buddy implementoval návrat dalšího sezení použitím funkce **filter()**, která provede iteraci nad všemi sezeními a vrátí list, který obsahuje pouze sezení, která nebyla absolvována. Následně pomocí další funkce knihovny Ramda - **head** je vrácen první prvek tohoto listu. [50]

■ Redux

Redux knihovna slouží pro správu stavu aplikace. Důležitou vlastností stavů z Redux je jejich neměnnost - stavy se při změně nemodifikují, ale vytváří se zcela nové. Změna stavu je prováděna na základě kombinace takzvaných **actions** (akce) a **reducers**. Objekty **actions** popisují, co se má stát, **reducers** určují, jak se má stav na základě konkrétní akce změnit.

Uvést si můžeme opět příklad z původní verze aplikace Buddy. Zde v souboru **submitLoginForm.js** je implementované vyvolání akce přihlášení, kdy aplikace zavolá funkci z knihovny Redux **useDispatch()** a ta následně po výběru konkrétní akce **ACTIONS.login()** vezme vstupní parametry a zavolá funkci **login()** ze souboru **authActions.js**, která potom provede logiku přihlášení uživatele.

```
const {value} = await dispatch(
  ACTIONS.login({
    email: values[FORM_FIELD_LOGIN_EMAIL],
    password: values[FORM_FIELD_LOGIN_PASSWORD],
    expoPushToken: undefined,
  }),
);
```


Kapitola 5

Modelování požadavků, návrh a design

V této kapitole nejprve vysvětlíme návaznost na výchozí stav aplikace a na předchozí práci. Poté definujeme nové funkční požadavky a nepotřebné odstraníme. Součástí návrhu je také část zabývající se definicí parametrů pro komunikaci mezi mobilní aplikací a serverovou částí. Podíváme se na framework Spring boot a databázi Postgre SQL, které využijeme pro implementaci řešení. Ve zbytku kapitoly provedeme návrh architektury, databázového modelu a sekvenčních diagramů.

5.1 Návaznost na předchozí práci

Tato diplomová práce přímo navazuje na bakalářskou práci *Buddy aplikace pro odvykání kouření* [21], ve které se student věnoval mimo jiné také zpětné kompletní analýze aplikace Buddy. Pro účely analýzy a modelování funkčních požadavků pro tuto diplomovou práci budeme vycházet z předchozí práce. Bude tedy nutné některé požadavky rozšířit, zrušit nebo jinak modifikovat. Pro funkcionality, které budeme zavádět nově, budeme požadavky definovat. Je důležité zmínit, že v rámci této diplomové práce bude implementováno serverové řešení pro aplikaci Buddy. Tím u některých požadavků dojde k delegaci za jejich zodpovědnost z mobilní aplikace Buddy na backendovou část.

Implementační kroky v rámci této diplomové práce lze bodově rozdělit na:

- Rozšíření roadmapy o udržovací fázi
- Aktualizace seznamu a struktury odznaků na základě nejnovější verze Mobilní aplikace pro odvykání kouření
- Načítání odznaků ze serverové části
- Vytvoření informačního widgetu pro zobrazení nejnovější odpovědi kuřáka ze sezení s Adamem
- Vytvoření historie potvrzených odpovědí kuřáka uživatelem
- Načítání historie potvrzených odpovědí ze serverové části

- Načítání týdenních reportů ze serverové části
- Odesílání zprávy kuřákovi na serverovou část
- Vytvoření serverové části systému pro snadnější správu a ukládání dat a pro poskytnutí větší flexibility při vývoji nových funkcionalit mobilní aplikace Buddy

■ 5.2 Funkční požadavky

Funkční požadavky budou zpracované pomocí následující šablony:

`<FR číslo požadavku> <název požadavku> <Systém umožňuje <co> (<komu>) (<za podmíněk>) (<proč>)>`

Číslování funkčních požadavků, pokud explicitně nereferencujeme na původní analýzu, bude pro účely této diplomové práce oddělené.

■ 5.2.1 Nové funkční požadavky mobilní aplikace Buddy

FR01 - Zobrazení postupu kuřáka v roadmapě. Systém uživateli umožňuje zobrazení postupu kuřáka formou progress baru všemi fázemi odvykání.

FR02 - Zobrazení odznaků kuřáka. Systém uživateli umožňuje zobrazení všech získaných odznaků kuřáka z Mobilní aplikace pro odvykání kouření.

FR03 - Načítání odznaků ze serverové části. Systém načítá získané odznaky kuřáka ze serverové části.

FR04 - Zobrazení nejnovější odpovědi kuřáka ze sezení. Systém umožňuje uživateli zobrazit nejnovější odpověď kuřáka ze sezení z Mobilní aplikace pro odvykání kouření.

FR05 - Potvrzení přečtení nejnovější odpovědi kuřáka ze sezení. Systém umožňuje uživateli potvrdit přečtení nejnovější odpovědi kuřáka ze sezení z Mobilní aplikace pro odvykání kouření.

FR06 - Zobrazení historie odpovědí kuřáka ze sezení. Systém umožňuje uživateli zobrazit historii všech přečtených odpovědí kuřáka ze sezení z Mobilní aplikace pro odvykání kouření.

FR07 - Načítání reportů ze serverové části. Systém načítá všechny týdenní reporty kuřáka ze serverové části.

FR08 - Odeslání zprávy na serverovou část. Systém umožňuje uživateli odeslat zprávu kuřákovi na serverovou část.

5.2.2 Odstranění původních požadavků

Smazání FR14 - Systém nadále neodemyká odznaky, ale načítá seznam odznaků ze serveru.

~~FR14 - Odznaky - Systém uživateli po dokončení sezení odemkne odznak v podobě ikony [21]~~

Smazání FR26 - Systém nadále nevytváří reporty, ale deleguje zodpovědnost za načítání reportů na serverovou část, která je připravena pro integraci na serverovou část Mobilní aplikace pro odvykání kouření.

~~FR26 - Vytváření reportů - Systém po minimálně každých 7 dnech vytvoří report zobrazující... [21]~~

Smazání FR27 - Systém nadále neukládá reporty do lokálního úložiště, ale načítá aktuální data ze serverové části, která je ukládá do databáze.

~~FR27 - Uložení reportu - Systém ukládá nově vytvořené reporty do lokální [21]~~

5.3 Návrh JSON zpráv pro rozhraní mezi aplikací Buddy a serverovou částí

Pro komunikaci mezi mobilní aplikací Buddy a serverovou částí bylo potřeba navrhnout rozhraní, pomocí kterého budou tyto systémy komunikovat. Pro účely této diplomové práce jsme zvolili REST API (RESTful API) rozhraní. Datovým formátem, který bude obsahovat potřebná data pro komunikaci mezi mobilní aplikací Buddy a serverovou částí, bude JSON. JSON je plně podporován jak na straně mobilní aplikace Buddy - React Native, tak na serverové straně, která bude implementována za pomoci frameworku Java Spring Boot.

5.3.1 Odznaky

Odznaky pro aplikaci Buddy budou rozděleny do tří kategorií tak, jak jsou definované v nejnovější verzi Mobilní aplikace pro odvykání kouření. Tyto tři kategorie zahrnují odznaky z přípravné fáze, odznaky z fáze odvykací a dále odznaky, jež reprezentují čas, jak dlouho kuřák vydržel bez cigarety. Návrh počítá s tím, že tyto odznaky jsou v rámci Mobilní aplikace pro odvykání kouření ukládány do databáze, a serverová část bude následně schopna tyto odznaky vystavit pro externí systém, kterým v tomto případě bude serverová část aplikace Buddy.

```
"properties": {  
  "preparationBadges": [1, 2, 3, 4, ...],  
  "cessationBadges": [101, 102, 103, 104, ...],  
  "noCigBadges": [201, 202, 203, 204, ...]  
}
```

5.3.2 Reporty

Mobilní aplikace Buddy zobrazuje na obrazovce seznam reportů (viz 4.8). Reporty se skládají z několika parametrů, které hodnotí kuřákov postup v týdenních intervalech. Každý report obsahuje stejné parametry. Těmi jsou datum vygenerování reportu, počet dnů od zahájení odvykací terapie, počet dokončených sezení od posledního vygenerovaného reportu, počet dokončených sezení celkem a fáze, ve které se kuřák nachází.

```
"allReports": [
  {
    "id": 1,
    "date": "01.01.2023",
    "daysSinceStarted": 7,
    "finishedSessionsSinceLast": 5,
    "finishedSessionsTotal": 5,
    "phase": "EE"
  },
  {
    "id": 3,
    "date": "08.01.2023",
    "daysSinceStarted": 14,
    "finishedSessionsSinceLast": 7,
    "finishedSessionsTotal": 12,
    "phase": "FU"
  },
  {
    "id": 5,
    "date": "15.01.2023",
    "daysSinceStarted": 21,
    "finishedSessionsSinceLast": 6,
    "finishedSessionsTotal": 18,
    "phase": "WR"
  }
],
"daysToNextReport": 7
}
```

V rámci JSON zprávy je nutné přidat ještě parametr počet dnů do dalšího reportu. Zprvu se může zdát zbytečné tento parametr posílat, jelikož si lze jednoduše odvodit generování reportu na základě data posledního. Problém však spočívá v tom, že bude-li nutné z byznysového pohledu generování reportu v budoucnu upravit, například na generování každé 3 dny místo původních 7 dnů, tak bude pro konzistenci dat nutné toto implementovat jak na straně mobilní aplikace Buddy, tak v serverové části. Systém navrhne tak, aby se toto měnilo pouze na jednom místě - na serverové části.

5.3.3 Odpověď kuřáka a historie odpovědí

Strukturu zprávy, kterou potřebujeme k načtení potřebných parametrů pro zobrazení odpovědi kuřáka ze sezení, lze složit ze tří parametrů. Textace otázky, odpověď kuřáka a nepovinné doporučení. Doporučení je nepovinný parametr a pokud je vynechán, widget zobrazí uživateli pouze znění otázky a odpověď na otázku.

```
{
  "question": "zneni otazky",
  "answer": "zneni odpovedi",
  "recommendation": "zneni doporuzeni"
}
```

Obdobně potřebujeme stejné parametry i u jednotlivých odpovědí historie přečtených odpovědí. Zpráva pro načítání historie obaluje tyto odpovědi do pole.

```
[
  {
    "question": "zneni otazky",
    "answer": "zneni odpovedi",
    "recommendation": "zneni odpovedi"
  },
  {
    "question": "zneni otazky",
    "answer": "zneni odpovedi",
    "recommendation": "zneni odpovedi"
  },
  {
    "question": "zneni otazky",
    "answer": "zneni odpovedi",
    "recommendation": "zneni odpovedi"
  }
]
```

Potvrzení odpovědi vytvoří zprávu, která bude posílána opačným směrem, tedy pomocí POST metody z mobilní aplikace Buddy na serverovou část. Vzhledem k návrhu logiky a databázového modelu lze tuto zprávu sestavit jednoduše dvěma parametry - ID uživatele a ID kuřáka.

```
{
  "buddyId": 1,
  "smokerId": 302
}
```

5.3.4 Zpráva kuřákovi

Poslání zprávy od uživatele kuřákovi lze definovat snadno pomocí jednoduché JSON zprávy obsahující 3 parametry. Text samotné zprávy, ID uživatele, který zprávu odesílá a ID kuřáka, pro kterého je zpráva určena. Žádné další parametry nejsou v souvislosti s touto funkcionalitou potřebné.

```
{
  "messageText": "Testovací text zpravy",
  "fromBuddyId": 1,
  "toSmokerId": 302
}
```

5.4 Návrh serverové části

Při návrhu serverové části je v prvotní fázi třeba zohlednit několik důležitých aspektů a vybrat vhodné serverové a databázové řešení. Aspekty, na základě kterých tyto technologie vybíráme, mohou mít technický i byznysový pohled. Jak si nyní v dalších podkapitolách rozebereme, zvolené technologie odůvodníme na základě obou pohledů.

5.4.1 Backendový framework - Spring Boot

V dnešní době existuje rozsáhlé množství backendových frameworků, ze kterých lze vybírat v případě, že potřebujeme vytvořit nové řešení od začátku. Podle webu <https://statisticsanddata.org/> jsou pěti nejpopulárnějšími frameworky pro tvorbu backendového řešení v roce 2023 od nejpopulárnějšího Laravel (PHP ¹), Django (Python ²), Spring, Flask (Python) a Express.js (Node.js - Javascript ³, Typescript ⁴). Spring se však v posledních letech těší významnému nárůstu popularity. Počátkem roku 2022 byl Spring ještě na pátém místě, kdežto počátkem roku 2023 se již nacházel na třetím místě. [22]

Při výběru vhodného frameworku pro vývoj backendových řešení je důležité si uvědomit, že neexistuje jednoznačně nejlepší framework univerzálně aplikovatelný na všechny projekty. Každý framework přináší specifické vlastnosti a funkce, které mohou být více či méně přínosné v závislosti na konkrétních požadavcích a specifikacích projektu.

Při posuzování potenciálního frameworku je klíčové zvážit řadu faktorů, které mohou výrazně ovlivnit jeho vhodnost pro daný projekt. Tyto faktory zahrnují zpětnou kompatibilitu, velikost a aktivitu vývojářské a uživatelské komunity, míru technické a uživatelské podpory, výkonnostní charakteristiky, kompatibilitu s existujícím technologickým stackem vývojového týmu a také finanční aspekty spojené s jeho implementací a údržbou.

V naší diplomové práci, v části zaměřené na vývoj serverové části aplikace Buddy, byla volba Spring Bootu strategickým rozhodnutím. Spring Boot poskytuje integrovanou podporu pro vytváření REST API, které jsou nezbytné pro komunikaci s mobilní aplikací. Jeho schopnost snadné integrace s JPA a široká kompatibilita s databázemi činí tento framework ideálním pro náš projekt. Mezi další výhody použití frameworku Spring Boot se řadí nasaditelnost na všech platformách, kterou zajišťuje JVM (Java Virtual Machine).

Vývoj aplikace Buddy je realizován primárně studenty Českého vysokého učení technického v Praze (ČVUT) z Fakulty elektrotechnické z oborů Softwarové inženýrství a technologie a Otevřené informatiky. Tyto studijní programy kladou značný důraz na výuku Javy, což je jeden z klíčových programovacích jazyků v moderním softwarovém inženýrství. Tento jazyk je nejen součástí

¹PHP - Skriptovací programovací jazyk pro vývoj dynamických webových aplikací

²Python - Vysokourovňový multiparadigmatický programovací jazyk

³Javascript - Multiparadigmatický skriptovací programovací jazyk

⁴Typescript - Typovaná nadstavba programovacího jazyka Javascript. Zpětně kompatibilní s Javascriptem.

standardního studijního plánu, ale je také uplatňován v souvislosti při návrhu a vývoji serverových řešení v mnoha předmětech. Volba frameworku Spring Boot je tedy v kontextu této diplomové práce a v návaznosti na práci budoucí zcela opodstatněná.

■ 5.4.2 Databáze

Volba technologie pro vytvoření databáze je poměrně přímočará. Logickou volbou je v tomto případě použití relační databáze, protože je nutné udržovat data strukturovaně a organizovaně na jednom místě. PostgreSQL tyto znaky splňuje. Jedná se navíc o open-source ⁵ technologii s širokou podporou, na které pracují vývojáři z celého světa. PostgreSQL navíc používá a rozšiřuje databázový jazyk SQL ⁶ a kombinuje jej s mnoha funkcionalitami, které se zaměřují na škálovatelnost, bezpečnost, spolehlivost a integritu dat. PostgreSQL je navíc snadno integrovatelné s Javou skrze ORM framework Hibernate, který je ve Spring Boot frameworku vestavěný.

■ 5.4.3 Další technologie

Pro vývoj serverového řešení jsme si vybrali několik dalších technologií, které umožňují snadný vývoj ve frameworku Spring Boot. Pro správu projektu a jednoduché sestavování závislostí jsme zvolili technologii Apache Maven. Ta nám umožňuje jednoduše do projektu přidávat závislosti na externí knihovny vkládáním do souboru **pom.xml**. Maven také zajistí jejich stažení a import. V moderním vývoji v jazyce Java se hojně používá knihovna *Lombok*, která slouží pro redukci zbytečného „boilerplate“ ⁷ kódu.

■ 5.4.4 Verze technologií

Zvolené technologie a jejich verze	
Technologie	Verze
Java	21 LTS
Spring boot	3.1.5
PostgreSQL	15
Maven	4.0.0
Lombok	1.18.30

Verze technologií jsou nejnovější v době zahájení implementace serverové části (k říjnu 2023).

⁵Open-source technologie je veřejná technologie, na které se podílí široká veřejnost a kterou může využít každý zdarma

⁶SQL - Structured query language - databázový dotazovací jazyk pro práci s daty v relačních databázích

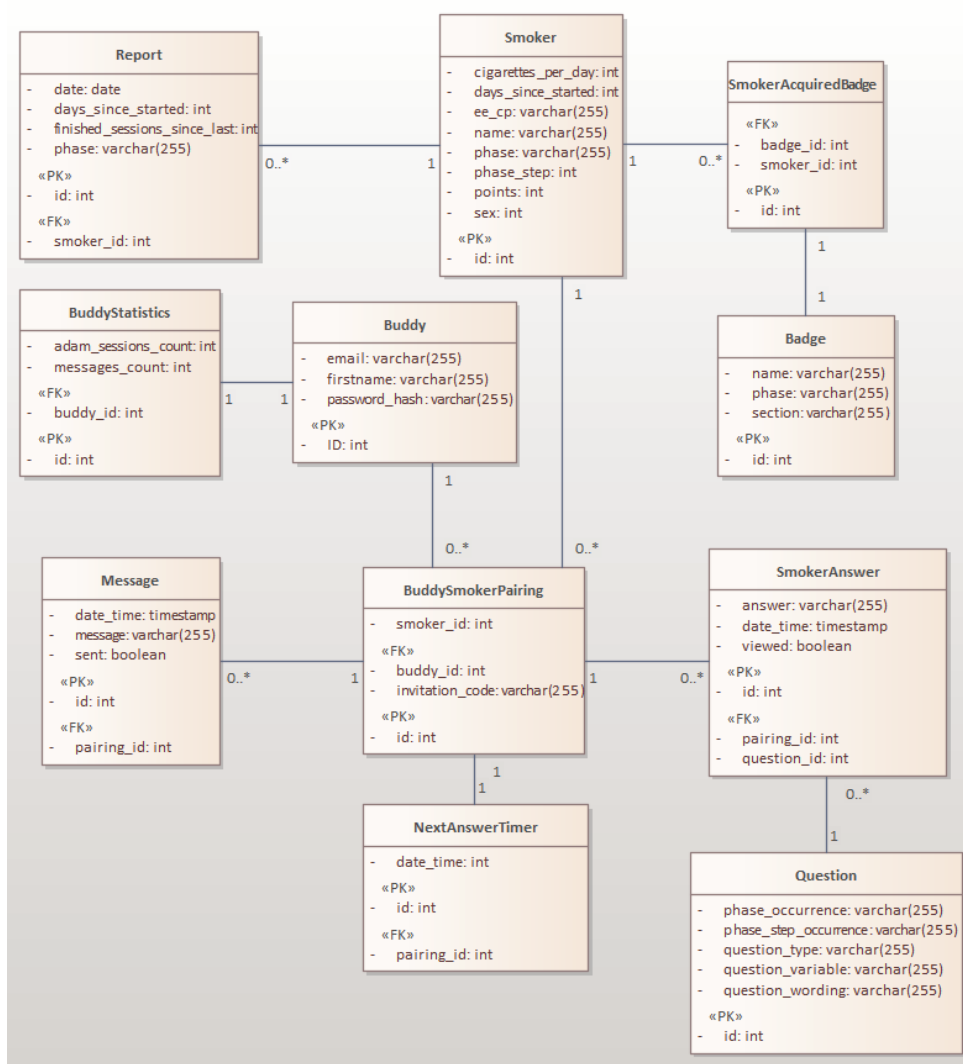
⁷Boilerplate kód - kód, který se v aplikaci často opakuje

5.5 Architektura

Architektura serverové části bude postavena na takzvané *vícevrstvé architektuře*. To znamená, že využijeme principů *Service layer pattern* a *Repository pattern*, kdy byznysová logika bude oddělená od logiky přístupu k datům. V této architektuře budou třídy rozdělené podle jejich zodpovědnosti na *Controllery*, *Service* a *Repository* - využijeme tedy tři vrstev (odtud vícevrstvá architektura, někdy též jako 3-vrstvá nebo n-vrstvá architektura). Pro přenos dat mezi vrstvou Controllerů a Service použijeme *Data Transfer Object pattern*, tedy vzor, který odděluje databázovou strukturu dat od byznysové, což umožňuje vyšší flexibilitu při vývoji a zajišťuje větší bezpečnost systému (vnějším systémům není odhalena struktura uložených dat v databázi). Struktura souborů a balíčků (package) bude tento princip dodržovat - přístup package by layer (narozdíl od opačného přístupu package by feature).

5.6 Databázový model

Abychom mohli v naší aplikaci pracovat s daty, musíme si nejprve definovat jejich strukturu a závislosti na databázové úrovni. K tomu nám poslouží UML diagram tříd, pomocí kterého si navrhne model databáze.



Obrázek 5.1: Databázový model - UML diagram tříd, zdroj: Autor

Klíčovou entitou v celém modelu je **BuddySmokerPairing**, jejímž účelem je agregovat entity **Smoker** a **Buddy**. Agregace poté využíváme v dalších entitách, jako jsou **Message**, **NextAnswerTimer** a **Question**. V tabulce 5.1 je popsán věcný význam jednotlivých entit.

Tabulka 5.1: Význam entit databázového modelu

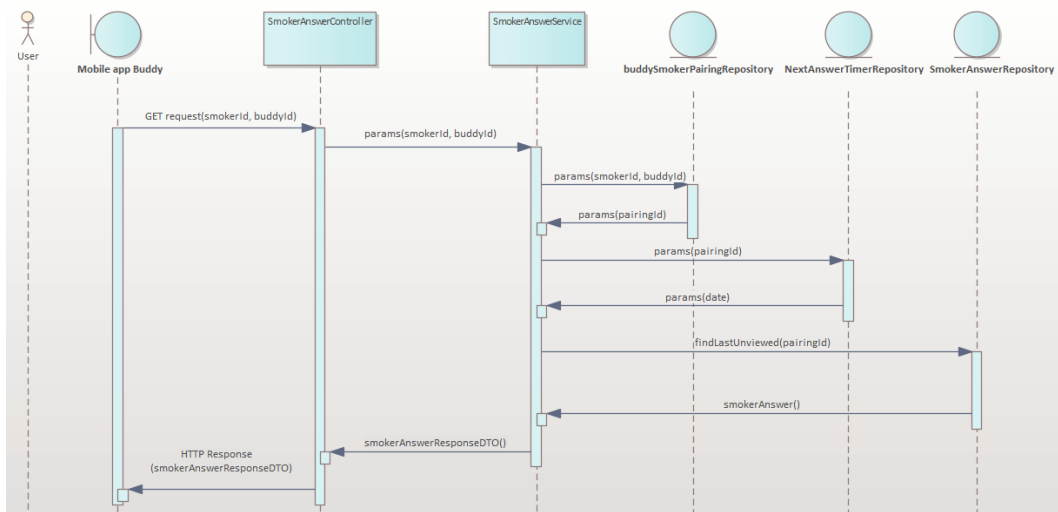
Entita	Význam
Buddy	Udržuje základní informace o uživateli aplikace Buddy. Má navržené parametry pro přihlašování. Tyto parametry v této diplomové práci nevyužíváme.
Pokračuje na další stránce	

Tabulka 5.1 pokračování z předchozí stránky

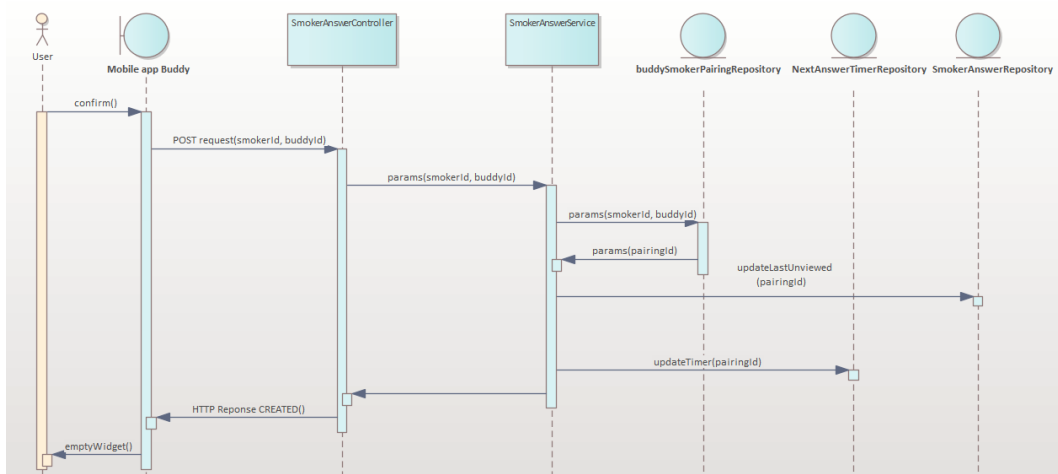
Otázka	Odpověď
Smoker	Udrží informace a parametry odvykajícího kuřáka ze serverového systému Mobilní aplikace pro odvykání kouření.
BuddySmokerPairing	Slouží k agregaci dvojice Smoker a Buddy. Je využita jako cizí klíč v entitách, které logicky nahlíží na entity Smoker a Buddy jako na ucelený pár.
BuddyStatistics	Entita slouží pro ukládání statistik uživatele mobilní aplikace Buddy. Statistika budou předmětem implementace mimo rozsah této diplomové práce.
Message	Entita pro ukládání zprávy, kterou může odeslat uživatel z mobilní aplikace kuřákovi.
NextAnswerTimer	V této fázi raného vývoje aplikace slouží jako časovač pro odeslání nové odpovědi kuřáka ze sezení. Tato entita se s největší pravděpodobností rozšíří o další časovače vázané na ucelený pár entit Smoker a Buddy.
SmokerAcquiredBadge	Entita, jejímž účelem je držet informaci ohledně získaných odznaků kuřáka během jeho postupu odvykací terapií.
Badge	Entita pro ukládání statických informací o odznacích.
SmokerAnswer	Entita pro ukládání odpovědí z kuřákových sezení.
Question	Entita pro ukládání statických informací o jednotlivých otázkách

5.7 Načítání a potvrzení odpovědi kuřáka

Pro vizualizaci interakce mezi jednotlivými objekty při načítání odpovědi kuřáka a také potvrzení přečtení odpovědi kuřáka uživatelem jsme vytvořili dva sekvenční diagramy UML, které jsou na obrázcích 5.2 a 5.3.



Obrázek 5.2: Interakce mezi objekty - načítání odpovědi, zdroj: Autor



Obrázek 5.3: Interakce mezi objekty - potvrzení odpovědi, zdroj: Autor

Z diagramu 5.2 je zřejmé, že načítání odpovědi z mobilní aplikace Buddy probíhá automaticky, konkrétně při otevření aplikace. Požadavek přijme controller, který následně zavolá servisní vrstvu. Servisní vrstva postupně ověří existenci páru kuřák - uživatel, načte dobu odeslání poslední odpovědi a poté si načte data z databáze. Provede konverzi mezi databázovou entitou a DTO objektem a předá controlleru pro následné odeslání odpovědi na požadavek.

Na diagramu 5.3 poté můžeme pozorovat akci, kdy uživatel potvrdí na hlavní obrazovce přečtení odpovědi. Pomocí metody POST je poté na server poslán požadavek na změnu stavu odpovědi kuřáka. Servisní vrstva provede kontrolu párování, změni stav odpovědi na „přečteno“ a aktualizuje datum časovače pro odeslání nové odpovědi. Controller poté vrací potvrzení pomocí HTTP Response CREATED a nakonec se uživateli změni vzhled widgetu s

upozorněním, že v současné chvíli není k načtení žádná nová odpověď.

Kapitola 6

Implementace frontendové části - Mobilní aplikace

V této kapitole si popíšeme jednotlivé kroky při implementaci funkcionalit mobilní aplikace Buddy. Nejprve se zaměříme na aktualizaci zastaralého načítání aplikace. Následně se zaměříme na popis implementace aktualizace odznaků a widgetu s odpovědí kuřáka. Poté se budeme zabývat vývojem funkcionality historie odpovědí a v poslední části stručně probereme implementaci přizpůsobení textu ze serverové části pro mobilní aplikaci Buddy.

6.1 Aktualizace zastaralého načítání aplikace

V rámci starších verzí Expo bylo doporučeným standardem načítání aplikace pomocí komponenty `<AppLoading>`, která dědí z komponenty `React.Component<AppLoadingProps>`. Toto načítání se však od vyšší verze stalo zastaralé a oficiální dokumentace platformy Expo v současné chvíli doporučuje konfigurovat načítání aplikace pomocí knihovny `expo-splash-screen`. Funkcí této knihovny je poskytnout vývojářům kontrolu nad načítacím procesem aplikace. Při startu aplikace je zpravidla nutné u rozsáhlejších projektů prvotně načíst řadu podpůrných souborů jako jsou fonty, obrázky, ikony nebo načíst potřebná data z provolání externích API. Nově implementovaná funkce `App()` tedy nyní vypadá takto:

```
SplashScreen.preventAutoHideAsync();
setupI18n();
function App() {
  const [appIsReady, setAppIsReady] = useState(false);
  useEffect(() => {
    async function prepare() {
      try {
        await Promise.all([
          Asset.loadAsync([
            ...
          ]),
          Font.loadAsync({
            ...
          }),
        ]);
      } catch (e) {
```

```

        ...
    } finally {
        setAppIsReady(true);
    }
}
prepare();
}, []);

if (!appIsReady) {
    return null;
}
SplashScreen.hideAsync();
return <LoadedApp/>;
}

```

6.2 Aktualizace odznaků

Abychom mohli implementovat funkcionalitu načítání odznaků ze serverové části, musíme implementovat funkci, která bude provolávat příslušný endpoint. V rámci Javascriptu existuje metoda **fetch()**, která vytváří asynchronně síťově požadavky na straně klienta. Tato metoda je součástí knihovny *Fetch API*, která poskytuje rozhraní pro manipulaci s řadou protokolů. [22] Zde je ukázka ze souboru **badgesEndpoints.js**, v němž je definovaná metoda **getBadges(smokerId)**, kterou voláme z příslušné komponenty pro načtení odznaků ze serveru.

```

export const getBadges = (smokerId) => {
    const apiEndpoint = LOCAL_NETWORK_ADDRESS + '/api/badges?
    smokerId=${smokerId}';
    const config = {
        method: "GET",
        headers: {
            'Content-Type': 'application/json',
        }
    };
    return fetch(apiEndpoint, config)
        .then(response => response.json())
        .then(data => {

            const preparationBadges = data.preparationBadges;
            const noCigBadges = data.noCigBadges;
            const cessationBadges = data.cessationBadges;

            return {
                preparationBadges,
                noCigBadges,
                cessationBadges
            };
        })
        .catch(error => {
            console.error('Error getting badges endpoint:',
                error);
            throw error;
        });
};

```

```
};
```

Metoda `fetch(url, options)` přijímá na vstupu dva parametry. Prvním z nich je URL odkaz volaného endpointu, druhý parametr poté přijímá objekt, který se skládá z definice parametrů jako metoda požadavku (GET, POST, PUT, DELETE) nebo header (hlavička). V našem případě při volání endpointu `/api/badges?smokerId=${smokerId}` voláme metodu GET s obsahem ve formátu JSON - `'Content-Type': 'application/json'`. Návratovou hodnotou metody `fetch()` je poté objekt typu `Promise<Response>`, který obsahuje informace o HTTP statusu a data, která nám volaný endpoint vrátil.

Odznaky jsou vykreslené v komponentě `ScreenDashboardProgressSmoker`. To jednak zařizuje volání endpointu pro načtení odznaků, jednak textaci jednotlivých odznaků. Texty jsou staticky uloženy v příslušných konfiguračních souborech, odznaky konkrétně v souboru `messagesDashboard.js`. Zde je stručná ukázka.

```
preparation: "Příprava",
  ['title-${BADGE_PREPARATION_01}']: "Už se zname",
  ['title-${BADGE_PREPARATION_02}']: "Vi, proc prestat",
  ['title-${BADGE_PREPARATION_03}']: "Chut kourit [[smoker:ho/
    jil]] neprekvapi",
  .
  .
  .
```

Komponenta `ScreenDashboardProgressSmoker` mimo jiné obsahuje řadu pomocných komponent jako potomků, ze kterých se skládají jednotlivé elementy UI.¹

```
<ScreenView>
  <HeaderWithBackButton title={t('title')} />
  <ScrollViewVertical>
    <ContainerWidgets>
      ...
    </ContainerWidgets>

    <ContainerCenter>
      <ContainerContent>
        <DividerVertical space={20} />
        <TitlePrimary4>{t('preparation')}</TitlePrimary4>
      >
        <DividerVertical space={20} />
        <FragmentSmokerProgressAchievements badgesType={
          BADGE_PREPARATION_TYPE} badgesList={badges.
            preparationBadges}/>
      </ContainerContent>
      <ContainerContent>
        ...
      </ContainerContent>
      <ContainerContent>
        ...
      </ContainerContent>
    </ContainerCenter>
  </ScrollViewVertical>
</ScreenView>
```

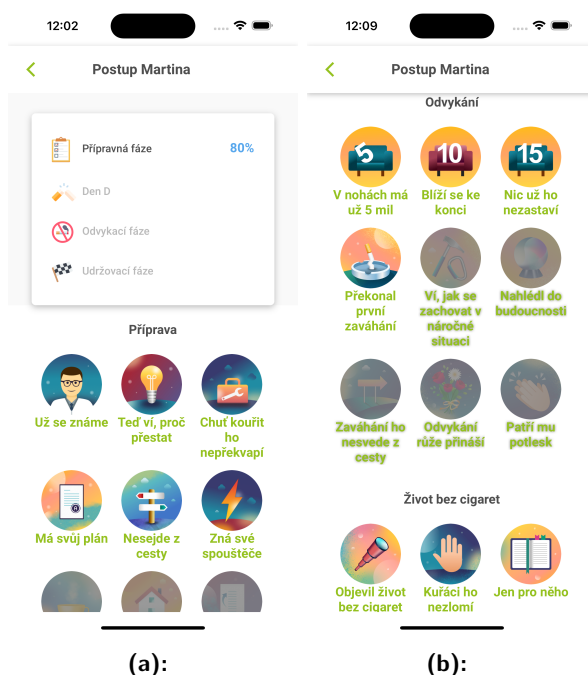
¹UI - User interface (uživatelské rozhraní)

```

</ContainerCenter>
</ScrollViewVertical>
</ScreenView>

```

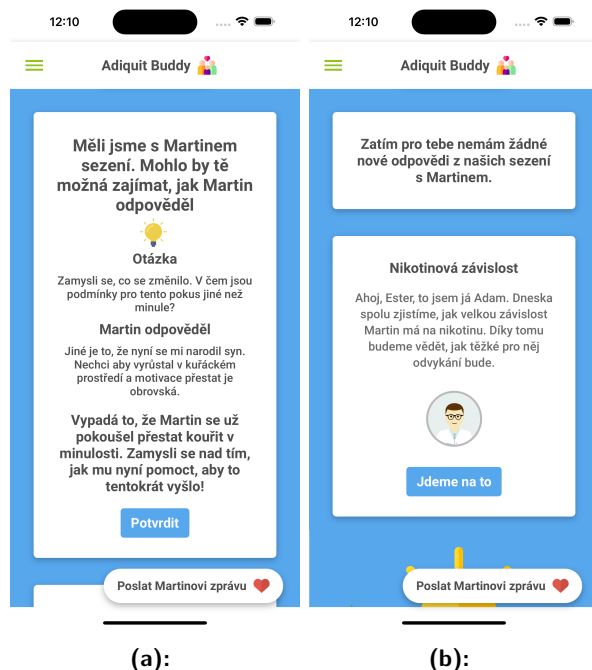
Komponenta `<FragmentSmokerProgressAchievements>` poté vykresluje jednu sekci odznaků seřazených do tabulkového formátu o třech sloupcích tak, jak je tomu v související Mobilní aplikaci pro odvykání kouření. To si lze prohlédnout na obrázcích 6.1a a 6.1b.



Obrázek 6.1: Obrazovka s odznaky (a) 1. část, (b) 2. část, zdroj: Autor - Aplikace Buddy

6.3 Widget s odpovědí kuřáka

V souboru `smokerAnswerEndpoint.js` jsme vytvořili tři metody `getAnswer(buddyId, smokerId)`, `confirmAnswer(buddyId, smokerId)` a `getAnswerHistory(buddyId, smokerId)`. Tyto metody volají rozhraní serverové části a plní jednotlivě veškeré funkce, které potřebujeme pro správné fungování komponenty pro odpověď kuřáka. Samotná komponenta, která má zodpovědnost za volání rozhraní a vykreslení widgetu, je v souboru `ScreenDashboard.js`. Jedná se současně o komponentu, která vykresluje celou hlavní obrazovku.



Obrázek 6.2: (a) Widget s odpovědí, (b) Widget bez odpovědi, zdroj: Autor - Aplikace Buddy

Na obrázku 6.2a je vykreslená komponenta poté, co aplikace přijala úspěšně data ze serverové části. Obrázek 6.2b znázorňuje vzhled po potvrzení přečtení odpovědi, když se data nepodaří načíst a nebo serverová část nemá připravená další data, která by vrátila.

```
<ScrollViewVertical
  refresh={true}
  onRefreshCallback={refreshCallback}
>
```

Komponenta **ScollViewVertical**, která je v hierarchii potomkem komponenty **ScreenDashboard**, navíc nabízí možnost nastavení parametru **refresh={true}**. Tento parametr zapíná na obrazovce možnost uživateli stránku znovu načíst. Volání endpointu jsme tedy propojili i s touto možností přes druhý parametr **onRefreshCallback={refreshCallback}**, který přijímá odkaz na metodu, jež nám zavolá inkriminovaný endpoint opakovaně. Uživatel má tak možnost manuálně aktualizovat hlavní obrazovku v případě potřeby. Asynchronní metoda **fetchAnswerData** volá metodu **smokerAnswerEndpoint.getAnswer(buddyId, smokerId)**, která nám v případě úspěšného zavolání endpointu data ze serverové části.

```
const fetchAnswerData = async () => {
  try {
    const data = await smokerAnswerEndpoint.getAnswer(
      buddyId, smokerId);
    setAnswerResponse({
      question: data.question,
```

```

        answer: data.answer,
        recommendation: data.recommendation
    });
  } catch (error) {
    ...
  }
};

```

V této ukázce kódu si lze zároveň všimnout operátoru **await** před zavoláním **getAnswer()**. Operátor **await** slouží k zachycení objektu **Promise**. Ten v Javascriptu slouží pro vyjádření úspěchu či neúspěchu daného volání. Použití objektu **Promise** v souvislosti s voláním API je tak standardní přístup při vývoji síťové komunikace mezi systémy, protože zde může snadno dojít k přerušení komunikace či návratu chybového stavu. Pokud došlo k úspěšnému načtení dat ze serverové části, tak objekt **Promise** vrátí data, která byla načtena. Pokud se uvnitř metody **getAnswer()** vyvolá chyba, tak ji zachytí konstrukce **try...catch**, kde uvnitř **catch()** je poté definováno specifické chování pro případ, že nastala chyba. [25, 26, 27]

6.4 Historie odpovědí

Komponenta **ScreenDashboardAnswersHistory** zajišťuje načítání seznamu odpovědí ze serverové části a zobrazuje je uživateli ve vzestupném pořadí podle toho, kdy bylo potvrzeno její přečtení. Uživatel tak má přístup k celé historii a může se k jednotlivým odpovědím vracet. Součástí této komponenty je také zobrazení načítacího kolečka, které se na obrazovce objeví po dobu, kdy metoda **smokerAnswerEndpoint.getAnswerHistory()** čeká na načtení dat ze serverové části. Samotná komponenta **<LoaderRing>** již byla v aplikaci implementována a my tak pouze přepoužíváme její vlastnosti. Zobrazení načítacího kolečka v době, kdy se načítají data ze serverové části, docílíme tím, že přidáme komponentně stav pomocí React Hook **useState(default_value)**, který následně využijeme pro rozhodování, kdy chceme kolečko zobrazovat a kdy již můžeme komponentu vykreslit. Před příchodem **useState()** se při nutnosti implementace stavu v komponentách ve frameworku React musela bezstavová funkce převést na třídu. Dnes lze tímto způsobem stav přidávat do funkcionálních komponent. [28]

```

import React, {useEffect, useState} from "react";
...
const [isLoading, setIsLoading] = useState(true);
...
useEffect(() => {
  const fetchData = async () => {
    setIsLoading(true);
    try {
      setIsLoading(false);
      ...
      setAnswersObject(fetchedData);
    } catch (error) {
      ...
      setIsLoading(false);
    }
  };
});

```

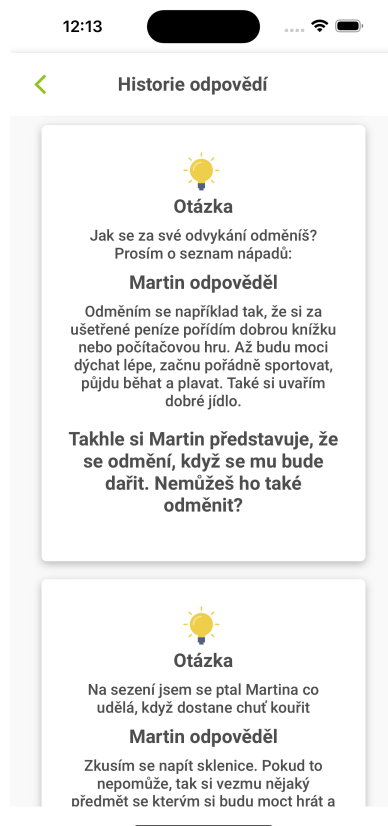


```

    ...
    dispatch(ACTIONS.notify("Doslo k chybe pri nacistani
                             dat ze serveru."));
  }
};
fetchData();
}, [dispatch]);
...
if (isLoading) {
  return (
    <ContainerLoader>
      <LoaderRing/>
    </ContainerLoader>
  );
}
}

```

V této komponentě používáme i druhý React Hook **useEffect**, kterým docílíme automatického spuštění efektu volání endpointu při renderování komponenty. [29] Současně uvnitř funkce **fetchData()** nastavíme proměnnou **isLoading** na **false** a tím zrušíme vykreslování kolečka a necháme vykreslit celou komponentu. Obrazovku historie odpovědí si lze prohlédnout na obrázku 6.3.



Obrázek 6.3: Obrazovka historie odpovědí, zdroj: Autor - Aplikace Buddy

6.5 Mapper pro skloňování textu ze serverové části

Součástí aplikace Buddy je knihovna **i18n**, která slouží pro překlad textů do různých světových jazyků. V rámci aplikace je však především využívána v kombinaci s knihovnou **czech-inflection**, která zajišťuje skloňování a přechylování slov pomocí příkazů, které se do textu zakomponují. V rámci komponent se potom zavolá již existující funkce **useComponentTranslation** a ta daný text upraví podle konfigurace. Například pokud chceme vyobrazit text s jménem kuřáka v 5. pádu, můžeme jednoduše napsat `{{smoker5p}}`. Funkce tento text nahradí skutečným jménem kuřáka v 5. pádu. Problém však nastává při načítání textu ze serverové části. Text se vrací jako jeden řetězec znaků a je žádoucí, aby v něm byla obsažena personalizovaná jména kuřáka a uživatele aplikace Buddy. Pro tyto účely jsme implementovali jednoduchý mapper, který dokáže v textu detektovat definované placeholdery a nahradit je personalizovanou textací. To umožňuje do budoucna snadným způsobem dynamicky načítat text ze serveru aplikace Buddy nebo Mobilní aplikace pro odvykání kouření, aniž bychom museli tyto texty ručně pro mobilní aplikaci Buddy editovat.

```
function replaceTextPlaceholders(text) {

    const t = useComponentTranslation(NS_DASHBOARD,
        SN_DASHBOARD_ANSWERS);

    text = text.replace(/\{\{smoker1p\}\}/g, t('smoker1p'));
    text = text.replace(/\{\{smoker2p\}\}/g, t('smoker2p'));
    text = text.replace(/\{\{smoker3p\}\}/g, t('smoker3p'));
    ...
    text = text.replace(/\[\[smoker:\a\]\]/g, t('smokerA'));
    text = text.replace(/\[\[smoker:on\ona\]\]/g, t('
        smokerOn0na'));

    return text;
}
```

Pokud bude v budoucnu nutné přidat další formy skloňování či jiné modifikace na základě pohlaví a jména, lze je přidat v souboru `messagesDashboard.js`.

Kapitola 7

Implementace backendové části - Spring Boot aplikace

V této kapitole, zabývající se implementační částí serverového řešení, se zaměříme nejprve na popis implementace funkcionality odpověď kuřáka. Posléze si popíšeme implementační kroky k řešení načítání odznaků. Navážeme částí zabývající se problematikou třídy ReportScheduler pro automatické generování reportů a v poslední části kapitoly poté rozebereme implementaci ošetření chybových stavů a s tím související třídu ApplicationExceptionHandler.

7.1 Odpověď kuřáka

Jedná se o funkcionalitu, ve které serverová část vrací na dotaz mobilní aplikace Buddy JSON objekt definovaný v kapitole 5.3.3. V rámci implementované vícevrstvé architektury se tak o tuto funkcionalitu stará několik tříd. Začneme nejprve u nejnižší vrstvy architektury, entit. Výsledný objekt, který vracíme mobilní aplikaci, skládáme na servisní vrstvě ze dvou entit. Jsou to třídy **SmokerAnswer** a **Question**, které reprezentují JPA ¹entity. JPA entity označujeme anotací **@Entity**, která třídu definuje jako objekt, na nějž se v rámci ORM mapuje databázová entita.

```
@Entity
@Table(name = "smokerAnswers")
@Getter
@Setter
@NoArgsConstructor
public class SmokerAnswer extends BaseEntity {

    @ManyToOne
    @JoinColumn(name = "pairing_id", nullable = false)
    private BuddySmokerPairing pairing;

    @ManyToOne
    @JoinColumn(name = "question_id", nullable = false)
    private Question question;

    private boolean viewed;
```

¹JPA - Java persistence api

```

    private LocalDateTime dateTime;

    private String answer;
}

```

Na tuto entitu ve vztahu **@ManyToOne** navazuje entita **Question**. Ta drží definice jednotlivých otázek i s doporučeními. Ve třídě **SmokerAnswerService** máme nadefinované tři metody, ve kterých je implementovaná logika sestavení objektu. Ten je poté skrze **SmokerAnswerController** vrácen mobilní aplikaci Buddy na dotaz. Zde si uvedeme ukázkou z třídy **SmokerAnswerService**, konkrétně metody **getFirstUnviewedSmokerAnswer()**, která vrací DTO objekt ² **SmokerAnswerResponseDTO**.

```

public SmokerAnswerResponseDTO getFirstUnviewedSmokerAnswer(
    Integer buddyId, Integer smokerId) {

    Optional<BuddySmokerPairing> buddySmokerPairingOptional
        = buddySmokerPairingRepository.
        findByBuddy_IdAndSmoker_Id(buddyId, smokerId);

    if (buddySmokerPairingOptional.isEmpty()) {
        throw new EntityNotFoundException("Smoker with ID "
            + smokerId + " or Buddy with ID " + buddyId + "
            not found.");
    }

    Optional<NextAnswerTimer> nextAnswerTimerOptional =
        nextAnswerTimerRepository.findByPairing_Id(
            buddySmokerPairingOptional.get().getId());

    if (nextAnswerTimerOptional.isEmpty()) {
        throw new EntityNotFoundException("Smoker with ID "
            + smokerId + " or Buddy with ID " + buddyId + "
            not found.");
    }

    if (nextAnswerTimerOptional.get().getDateTime().isAfter(
        LocalDateTime.now())) {
        throw new SmokerAnswerNoContentException("There is
            currently no unviewed smoker answer");
    }

    Optional<SmokerAnswer> smokerAnswerOptional =
        smokerAnswerRepository.
        findFirstByPairing_IdAndViewedFalseOrderByDateTimeAsc(
            buddySmokerPairingOptional.get().getId());

    if (smokerAnswerOptional.isPresent()) {
        return convertSmokerAnswerEntityToSmokerAnswerDTO(
            smokerAnswerOptional.get());
    }

    throw new SmokerAnswerNoContentException("There is
        currently no unviewed smoker answer");
}

```

²DTO - Data transfer object, objekt zpravidla využívaný pro transfer dat mezi vrstvami aplikace

```
}

```

V principu se logika skládá ze tří kroků. V prvním nejprve provedeme kontrolu, zda dvojice parametrů **smokerId** a **buddyId**, které přišly s požadavkem z mobilní aplikace Buddy, tvoří existující pár. Tedy že odpovídající kuřák má skutečně svého podporovatele. Poté voláme na **NextAnswerTimerRepository** databázový dotaz pro navrácení timeru. Timer potřebujeme, abychom mohli ověřit, že uživatel má nárok na novou odpověď. Pro případ, že existuje dostupná nepřčtená odpověď, je implementována logika, kdy se dá odpověď vrátit pouze jednou za určitý časový úsek. Tento časový úsek je držen ve třídě **Constants** jako statický parametr **NEXT_ANSWER_INTERVAL_IN_DAYS**.

Pokud Timer pro ověření nároku další odpovědi existuje, je využit pro kontrolu, zda definovaná doba již uplynula. To je v kódu reprezentováno pomocí metody **isAfter()** volané nad datovým typem **LocalDateTime**. V poslední části implementované logiky se poté dotazujeme pomocí **SmokerAnswerRepository** a její vytvořené derivované metody na vrácení první odpovědi, která přísluší páru **smokerId** a **buddyId** a je zároveň nepřčtená. Tato entita je posléze konvertována v metodě **convertSmokerAnswerEntityToSmokerAnswerDTO** na objekt **SmokerAnswerResponseDTO** a ten je vrácen třídě **SmokerAnswerController**, která objekt wrapuje na **ResponseEntity** a následně posílá mobilní aplikaci Buddy.

Součástí **SmokerAnswerService** jsou ještě metody **getAllViewedAnswers()** a **confirmViewOfLastAnswer()**. Metoda **getAllViewedAnswers()** funguje na stejném principu, jako výše popsané chování v metodě **getFirstUnviewedSmokerAnswer()** s tím rozdílem, že zde není nutné kontrolovat Timer, protože vracíme již přečtené odpovědi a zároveň vracíme pole **List<SmokerAnswerResponseDTO>**. Metoda **confirmViewOfLastAnswer()** však implementuje jinou logiku. Zde naopak updatujeme databázovou entitu odpovědi poté, co uživatel potvrdí její přečtení v mobilní aplikaci Buddy.

7.2 Odznaky

Třída **BadgeController**, která vystavuje REST API endpoint pro načítání odznaků do mobilní aplikace, nese anotace **@RestController** a **@RequestMapping**. **@RestController** je anotace, která má vlastnosti standardní anotace komponenty a vytváří z třídy beanu. Slouží obecně pro vystavování rozhraní pro externí systémy. Anotace **@RequestMapping** mapuje webové požadavky na konkrétní metody. [35, 35]

```
@Slf4j
@RestController
@RequiredArgsConstructor
@RequestMapping("/api/badges")
public class BadgeController {
```

```

private final BadgeService badgeService;
@GetMapping
public ResponseEntity<SmokerBadgesAcquiredResponseDTO>
    getAllBadges(@RequestParam Integer smokerId) {
    log.info("Api GET /badges was invoked with smokerId='{}'",
        ", smokerId);
    return ResponseEntity.ok(badgeService.getAllBadges(
        smokerId));
}
}

```

V našem případě máme vystavené webové rozhraní s url adresou `/api/badges`. Anotace `@GetMapping` nad metodou `getAllBadges()` nám přímo mapuje requesty typu GET, které přijdou na zmíněnou url adresu. Zároveň se očekává parametr `smokerId`.

`BadgeService` se poté stará o logiku načítání odznaku z databáze. Uvnitř metody `getAllBadges()` je volaná `SmokerAcquiredBadgeRepository`, která pomocí metody `findBySmoker_Id(smokerId)` vrátí seznam všech dosažených odznaků daného kuřáka. Seznam je poté akorát nutné filtrovat podle typu odznaku, který vychází z enumerace `Section`. Hodnoty jsou `PREPARATION`, `CESSATION` nebo `NO_CIG`, podle toho, do které z kategorií odznak zapadá.

7.3 Report scheduler

Pro implementaci logiky reportů je v aplikaci připravená třída `ReportScheduler`, která slouží k opakovanému generování reportů jednou za definovaný časový úsek. V současné chvíli se jedná o 7denní iteraci, což je nastaveno ve třídě `Constants` ve statickém parametru `REPORT_CREATION_REPETITION_IN_DAYS`. Protože aplikace se v současné chvíli neintegruje na serverovou část Mobilní aplikace pro odvykání kouření, je tento scheduler implementován pro testovací účely a simulaci situace, kdy volá jednou za týden hypotetický endpoint pro stažení aktuálních dat o kuřákovi.

```

@Component
@RequiredArgsConstructor
public class ReportScheduler {

    private final ReportService reportService;

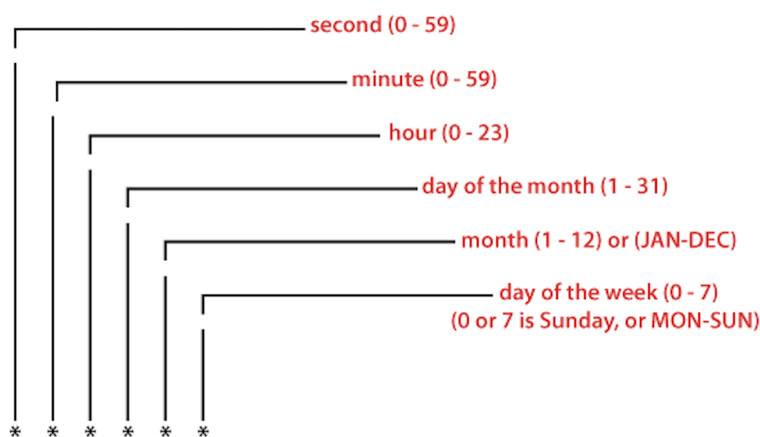
    @Scheduled(cron = "0 0 0 * * MON")
    public void fetchReportData() {

        // MOCKED DATA for simulation
        ...

        reportService.createReports(List.of(report01, report02,
            report03, report04));
    }
}

```

Třída **ReportScheduler** disponuje anotací **@Component**. Ta zařídí, že bude třída automaticky při skenování všech tříd zařazena mezi Beany, tedy třídy se samostatným životním cyklem, které se mohou pomocí Dependency injection patternu vkládat na místa, kde potřebujeme využít jejich rozhraní. [40] Pro automatické periodické spouštění metod existuje anotace **@Scheduled**. Vstupním parametrem je poté atribut typu **cron**, **fixedDelay**, **fixedRate**. V našem případě používáme pro nastavení periodického volání **cron** a to konkrétně s hodnotou **"0 0 0 * * MON"**, tedy volání každé pondělí v 0:00:00. [32, 33, 34]



Obrázek 7.1: Grafický rozbor hodnot parametru cron, zdroj: [32]

7.4 Ošetření chybových stavů

Součástí vývoje kvalitního a přepoužitelného kódu je také správné ošetření chybových stavů kódu. V Javě je tento problém řešen pomocí výjimek (Exception). Ty nám buď při vytvoření chyby při běhu programu vyhodí JVM, nebo je můžeme vyhodit ručně pomocí **throw**. [37] Při práci s daty v servisní vrstvě naší aplikace se můžeme setkat s chybovými stavy na mnoha místech. Je proto žádoucí, abychom všechny tyto stavy měli podchycené tak, aby nic nenarušovalo chod našeho programu. Protože v servisní vrstvě chybové stavy kódu přímo ovlivňují chování controlleru, který je zodpovědný za komunikaci s externím systémem, využili jsme vlastnosti anotace **@RestControllerAdvice**, jež nám ručně z anotované třídy **ApplicationExceptionHandler** vytvoří typ třídy, která zpracovává vyhozené výjimky. Jednotlivým metodám třídy jsme přidali anotace **@ExceptionHandler** a **@ResponseStatus**, jak lze vidět v ukázce kódu.

```
@RestControllerAdvice
public class ApplicationExceptionHandler {

    @ExceptionHandler(value = Exception.class)
    @ResponseStatus(value = HttpStatus.INTERNAL_SERVER_ERROR)
    public ErrorMessage commonException(Exception ex) {
        return new ErrorMessage(ex.getMessage(), Instant.now());
    }
}
```

```

    }

    @ExceptionHandler(value = {
        SmokerNotFoundException.class,
        SmokerAnswerNotFoundException.class,
        BuddyNotFoundException.class,
        EntityNotFoundException.class})
    @ResponseStatus(value = HttpStatus.NOT_FOUND)
    public ErrorMessage resourceNotFoundException(Exception ex)
    { return new ErrorMessage(ex.getMessage(), Instant.now())
      }; }
    ...
    ..
    .
}

```

@ExceptionHandler anotace přijímá jako parameter literály tříd, které dědí z obecné třídy **Exception**. V našem případě dědíme výjimky od třídy **RuntimeException**, která ve své definici z třídy **Exception** dědí, tím tedy máme podmínky pro vstupní parameter splněné. Anotace **@ResponseStatus** přidává navíc k metodě značku HTTP statusu. V naší ukázce kódu chceme, aby se v případě vyhození všech vypsanych výjimek v **@ExceptionHandler** vracel návratový HTTP kód **NOT_FOUND**, tedy číslo kódu 404. [38, 39] V souvisejícím controlleru, který poté zachycuje výjimky vyhozené v případě, že nastane chybový stav v servisní vrstvě, jsou chyby z těchto výjimek vypsane ve třídě **ErrorMessage**. Tu controller vrátí externí aplikaci jako zprávu. Zde se nám poté v případě, že při běhu aplikace nastane chybový stav, objeví konkrétní chyba. Příkladem může být zavolání endpointu s neexistujícím smokerId `/smoker-answer/history?buddyId=1&smokerId=-9999`. V následující ukázce JSON objektu je odpověď, která byla vrácena s chybovým kódem 404.

```

{
  "message": "Smoker with ID -9999 or Buddy with ID 1 not
    found.",
  "time": "2023-12-09T21:11:18.921456Z"
}

```

7.5 Návrhové vzory a principy použité při vývoji

Ve snaze napsat jednoduchý, čitelný a přepoužitelný kód jsme se při vývoji řídili řadou doporučených principů a návrhových vzorů. Jedním z těchto principů je DRY (*Don't repeat yourself*), tedy princip, který nám říká, že se nemáme při implementaci opakovat, což vede ke srozumitelnějšímu a čistějšímu kódu. Dalším principem, kterým jsme se při vývoji řídili, je KISS (*Keep it simple stupid*). Při vývoji jsme se tak snažili vyhnout zbytečným složitostem a psát udržitelný a srozumitelný kód.

Z návrhových vzorů, které se v serverové části aplikace vyskytují, jsou například *Inversion of Control* a *Dependency Injection*. Na těchto návrhových vzorech stojí princip Spring Boot frameworku. Dalším takovým návrhovým

vzorem, který používáme, je *Singleton* - všechny Beany, které v programu definujeme jsou instanciovány pouze jedenkrát. Pro izolaci přístupu k datům od byznysové logiky jsme využili *Repository pattern*. Přenos dat mezi jednotlivými vrstvami zajišťují DTO třídy čímž také využíváme takzvaný *Data Transfer Object Pattern*. V neposlední řadě můžeme zmínit využití návrhového vzoru *Service Layer Pattern*, který slouží pro oddělení byznysové logiky aplikace od prezentační a datové.

Kapitola 8

Testování

V kapitole Testování se nejprve zaměříme na moderované testování mobilní aplikace. Vysvětlíme důvod, proč jsme zvolili tuto metodiku a zdůvodníme její návaznost na testování z původní práce. V souvislosti s moderovaným testováním si popíšeme přípravu scénářů a dotazníku a k průběhu testování uvedeme konkrétní report k jednomu z respondentů. V druhé části poté popíšeme detaily z implementace jednotkových testů.

8.1 Moderované testování mobilní aplikace

Součástí předchozích iterací vývoje mobilní aplikace Buddy jsou provedené testy datové konzistence i uživatelské testování. Uživatelské testování však probíhalo vzdálenou formou, což může mít vliv na přístup a aktivitu uživatelů. [21] Pro účely této diplomové práce tak po konzultaci byla zvolena metoda moderovaného testování. Ta spočívá v jednorázových sezeních s uživatelem, který podrobí aplikaci testům na základě snadných a jasně definovaných scénářů. Důraz bude kladen především na otestování funkcionalit, které byly nově implementovány. Uživatel bude mít možnost aplikaci projít celou a předat zpětnou vazbu.

Celosvětově uznávanou autoritou v teorii uživatelského testování je Steve Krug, který se proslavil svou knihou *Don't make me think* (česky *Nenuťte mě přemýšlet*). Steve Krug ve své knize klade důraz především na jednoduchost a selský rozum při návrhu uživatelského rozhraní a samotných testů. Steve Krug testy použitelnosti definuje ve své knize takto: *„Testy použitelnosti jsou o sledování, jak se jedna osoba snaží používat něco (ať už je to webová stránka, prototyp nebo náčrty nového designu) k provádění typických úkolů, abyste mohli detekovat a opravit věci, které ji matou nebo frustrují.“* Ve své knize dále zmiňuje a klade důraz na to, že *„testování jedním uživatelem, je lepší, jak žádné testování“* a také že *„testovat aplikaci jedním uživatelem v rané fázi vývoje je lepší, jak testování padesáti uživateli v konečné fázi vývoje“*. [30] Tyto myšlenky jsme pro účely testování této diplomové práce převzali a implementovali.

Důvodem volby moderovaného testování je návaznost této práce na existující řešení. V předchozí práci se student věnoval uživatelskému testování, kdy vybral 10 respondentů, jimž poskytl mobilní aplikaci Buddy. Respondenti

byli vyzváni k týdennímu používání této aplikace. Výsledky byly shrnuty na základě diskuze a vyplnění formulářového dotazníku. Náš přístup k testování skrze metodu moderovaného testování si klade za cíl na tyto testy navázat a poskytnout nová data, ze kterých lze v dalších iteracích vývoje mobilní aplikace Buddy vycházet.[21]

8.1.1 Příprava scénářů a dotazníku

Pro moderované testování jsme nejprve sestavili několik scénářů, kterými se uživatelé budou řídit. Drželi jsme se doporučení Steva Kruga o jejich jednoduchosti a srozumitelnosti pro uživatele. Důraz byl kladen především na nově implementované funkcionality, ale průchod celé aplikace ve zbytku času testovací schůzky nebyl opomenut.

Scénář	Popis
1	Otevřete aplikaci a zobrazte si seznam odznaků, které kuřák získal během procesu odvykání kouření. Všimněte si, jaké odznaky byly získány.
2	Najděte a přečtěte si poslední odpověď kuřáka z jeho sezení. Po přečtení odpovědi použijte tlačítko k potvrzení, že jste odpověď přečetli.
3	Přejděte do sekce historie a prohlédněte si předchozí odpovědi kuřáka.
4	Zobrazte si statistiky kuřáka za uplynulé týdny. Podívejte se na jednotlivé parametry.
5	Na hlavní stránce aplikace najdete roadmapu kuřákovy odvykacího procesu. Zkontrolujte jeho současnou pozici a poté si přečtěte detailní informace o udržovací fázi odvykání.
6	Použijte funkci pro odeslání zprávy a napište kuřákovi povzbudivou zprávu, ve které vyjádříte svou podporu. Ve zprávě použijte alespoň jeden emoticon.

Tabulka 8.1: Scénáře uživatelského testování

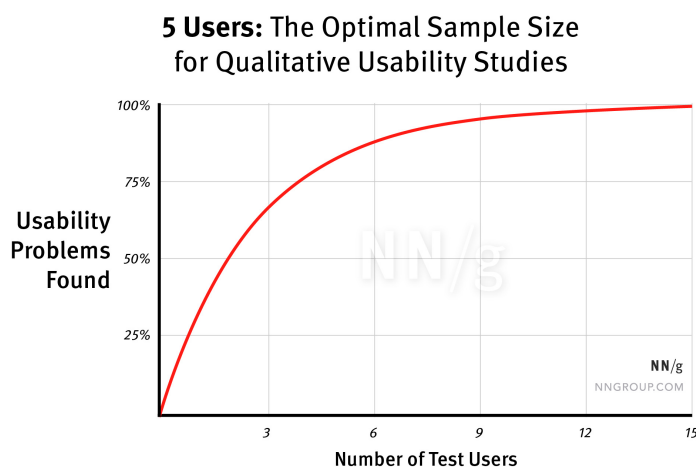
Následně jsme tyto scénáře zakomponovali do dotazníkové formy a přidali ještě doplňující otázky, které nám pomohou lépe analyzovat celkový pohled uživatelů na aplikaci.

Č.	Otázka
1	Zdá se Vám aplikace Buddy přehledně řešená?
2	Popište vlastními slovy pozitivní poznatky k přehlednosti aplikace.
3	Popište vlastními slovy nedostatky v přehlednosti aplikace.
4	Jaké funkcionality se Vám na aplikaci líbily nejvíce a proč?
5	Jaké funkcionality se Vám na aplikaci nelíbily a proč?
6	Pokud by Vás kuřák oslovil s žádostí o pomoc s jeho odvykáním a poslal Vám pozvánku do aplikace Buddy, přijal/a byste ji a aplikaci pravidelně používal/a?

Tabulka 8.2: Dotazníkové otázky

8.1.2 Průběh testování

Pro moderované testování jsme oslovili celkem pět respondentů. Tento počet respondentů je zcela v souladu s výzkumem, který provedla skupina NN Group (Nielsen Norman Group), jež je velmi uznávanou autoritou v oblasti uživatelského testování. Jakob Nielsen ve svém článku z roku 2000 uvádí graf zobrazující míru vyskytu problémů v použitelnosti vůči počtu testujících uživatelů. [31]



Obrázek 8.1: Graf znázorňující míru vyskytu chyb na počet testujících uživateli, zdroj: NN Group [31]

Naši respondenti zvolení pro moderované uživatelské testování jsou uvedeni v tabulce 8.3.

Číslo	Pohlaví	Věk	Kuřák	Povolání
1	muž	26	bývalý	student medicíny
2	muž	27	ano	softwarový inženýr
3	muž	53	ne	podnikatel
4	žena	55	ano	lékař
5	žena	25	ne	lékař

Tabulka 8.3: Seznam respondentů

Každý z respondentů byl uveden do kontextu před tím, než se do testování pustil. Úvodní text:

„Představte si, že máte dva kamarády. Martina a Janu. Martin i Jana se snaží zbavit závislosti na kouření. K tomu jim pomáhá Mobilní aplikace pro odvykání kouření. Martin i Jana Vás požádali, abyste jim s odvykáním pomohli. K tomu slouží mobilní aplikace Buddy, kterou dostanete do rukou k otestování. Aplikaci jste si nainstalovali na svůj mobilní telefon a prošli jste registrací a přihlášením a nyní máte aplikaci před sebou otevřenou. Teď Vám budu postupně zadávat scénáře a požádám Vás, abyste je splnili. Pokud během testování narazíte na problém, neváhejte se mě zeptat.“

Respondentům jsou poté postupně zadávány scénáře a jsou požádáni o jejich splnění. Pro ukázkou si uvedeme zápis dialogu a chování respondenta z jednoho testování, konkrétně respondenta číslo 1 z tabulky 8.3.

Tabulka 8.4: Zápis z moderovaného uživatelského testování uživatele č. 1

Scénář	Dialog
1	<p>Respondent: Scrolluje po hlavní obrazovce, snaží se nalézt, kde by mohly odznaky být. Často promlouvá řečnickými otázkami a hledá na spodní části obrazovky. Poté kliká na reporty. Vrací se na hlavní obrazovku.</p> <p>Moderátor: „S čím podle Vás souvisí odznaky?“</p> <p>Respondent: „S postupem v druhé aplikaci.“</p> <p>Moderátor: „Dobrý nápad, tak se na to zaměřte.“</p> <p>Respondent: Prohlíží si roadmapu. Scrolluje na horní část obrazovky a kliká na tlačítko „Ukázat cestu“. „To je ono, ne?“</p> <p>Moderátor: „Vypadá to jako odznaky?“</p> <p>Respondent: „Ano.“ Prohlíží si odznaky. „Takže tyhle odznaky jsou ty získané a tyhle budou asi ještě neodemčené.“ Ukazuje na rozmazané odznaky.</p> <p>Moderátor: „Přesně tak.“</p>
2	<p>Respondent: Vrací se na hlavní obrazovku. Ihned si všimá odpovědi. Čte. „Mám to potvrdit?“</p> <p>Moderátor: „Jistě.“</p>
Pokračuje na další stránce	

Tabulka 8.4 pokračování z předchozí stránky

Scénář	Dialog
3	Respondent: Scrolluje po hlavní obrazovce. Po chvíli kliká na widget historie odpovědí. Scrolluje po widgetu. „Tady je ta odpověď, kterou jsem potvrdil.“ Moderátor: „Ano.“
4	Respondent: Scrolluje po hlavní obrazovce. Kliká na tlačítko Ukázat cestu. „To je tohle?“ ptá se a ukazuje na procentuální vývoj fází. Moderátor: „Myslíte, že to vypadá jako statistiky?“ Respondent: Přemýšlí. „Spíše ne.“ Vrací se na hlavní obrazovku. Scrolluje. „Je to tohle?“ Ukazuje na widget s ušetřenými penězi a scrolluje dál. „Tak tohle.“ Ukazuje na reporty. „To mě zmátlo, ale má chyba. Hledal jsem statistiky a neuvědomil si, že tím jsou myšlené reporty.“ Prohlíží si týdenní reporty. „Tady by se mi líbilo přidat ještě kolik vykouřil cigaret za ten týden.“
5	Respondent: Vrací se na hlavní obrazovku a scrolluje nahoru přímo na umístění roadmapy. „Ta se mi moc líbí.“ Vybírá tlačítko Udržovací fáze a čte popis. „Jasně, trvá 10 týdnů.“
6	Respondent: Vrací se na hlavní obrazovku. Chvíli scrolluje. Poté vybírá tlačítko Poslat zprávu Martinovi v pravém dolním rohu. Píše text a přidává emoticon. Kliká na tlačítko odeslat.

Uživatel se poté ještě podrobil dotazníkovému formuláři, ve kterém doplnil poznatky z testovacích scénářů a v druhé části poskytl celkový pohled na aplikaci.¹

Tabulka 8.5: Odpovědi uživatele č. 1 dotazníkový formulář

Otázka	Odpověď
Scénář 1: Zobrazte si seznam odznaků, které kuřák získal během svého odvykání	Ocenuji rozdělení na 3 části. Líbí se mi grafická stránka jednotlivých odznaků. Chybi mi možnost rozkliknutí jednotlivých odznaků, kde bych očekával jejich přesnou definici (za co je exkurak získal).
Scénář 2: Přečtete si poslední odpověď z kuřákovy sezení a potvrďte její přečtení	Přístup jednoduchý a intuitivní, nic bych neměl. Tlačítko "potvrdit" považuji za vhodný krok. K uvážení bych dal, zda si "exkurak Martin" nezaslouží kromě pochvaly a podpory i radu protistrany.
Pokračuje na další stránce	

¹Poznámka autora: Pro zachování autenticity odpovědi testovaného uživatele jsme ponechali odpovědi v původním stavu, tedy bez diakritiky i s překlepy

Tabulka 8.5 pokračování z předchozí stránky

Otázka	Odpověď
Scénář 3: Zobrazte si historii předchozích odpovědí kuřáka	Kategorie "historie" se mi ze své podstaty líbí, připadá mi důležité mít možnost vrátit se k předchozím "sezením".
Scénář 4: Zobrazte si statistiky za uplynulý týden	Kategorie statistiky se mi líbí, je jednoduchá a přehledná. Ke zvažení bych dal, zda tam není kolonka pro počet vykouřených cigaret za dané období.
Scénář 5: Podívejte se na kuřákovův postup na roadmapě na hlavní stránce aplikace a poté si přečtěte detail o udržovací fázi	Roadmapa se mi líbí, je přehledná a jednotlivé fáze jsou po rozkliknutí jednoznačně definovány.
Scénář 6: Pošlete kuřákovi pozitivní zprávu, že mu držíte palce a že ho plně podporujete v jeho odvykání. Použijte alespoň jeden emoticon	Průchod byl snadný a intuitivní, mám rozporuplné pocity z výzvy k použití emotikonu.
Zdá se Vám aplikace Buddy přehledně řešená?	Ano
Popište vlastními slovy pozitivní poznatky k přehlednosti aplikace	Velmi oceňuji Roadmapu, celkově i rozdělení do kategorií historie odpovědi, reporty, opakování sezení mi připadá přehledné.
Popište vlastními slovy nedostatky v přehlednosti aplikace	Do kategorie historie odpovědi bych přidal dataci jednotlivých odpovědi.
Jaké funkcionality se Vám na aplikaci líbily nejvíce a proč?	Nejvíce se mi líbí Roadmapa. Vysvětluje vše podstatně o procesu odvykání, jaký typ podpory "Martin" v které fázi nejvíce potřebuje a o čem vlastně ta která fáze je. Navíc je hezká a jednoduše zpracována po grafické stránce.
Jaké funkcionality se Vám na aplikaci nelíbily a proč?	Nemožnost rozkliknutí jednotlivých odznaků za účelem získání jejich přesné definice

8.1.3 Vyhodnocení výsledků

Moderované testování proběhlo bez větších problémů. Všichni respondenti se úspěšně podrobili testování a prošli všechny nadefinované scénáře. Aplikace během testování ani jednomu z respondentů nespadla a napojení na serverový systém spuštěný pro účely testování v lokální síti bylo stabilní. Někteří

respondenti během testování měli problém v aplikaci nalézt místa, kde se nachází elementy, které byly ve scénáři popsány. Nejvíce problematické bylo mezi respondenty hledání odznaků a statistiky kuřáka za uplynulé týdny. To však nutně neznamená chybně definované uživatelské rozhraní. Odznaky jsou spíše doplňková funkcionality a uživatel se při prvním používání aplikace často chová dezorientovaně a netuší, co si pod pojmem odznaky představit. Problematické hledání statistik, bylo spíše způsobeno textací při definici scénáře. Poslední 2 uživatelé již místo slova statistiky měli uvedené ve scénáři slovo reporty a to výrazně snížilo dobu hledání tohoto prvku v aplikaci.

Po analýze odpovědí z vyplněných formulářů a samotného moderovaného testování lze odvodit následující nedostatky k budoucí konzultaci. V závorce uvádíme respondenty, u kterých se nedostatek projevil.

- U odznaků chybí popis za co jsou získané (respondent č. 1, 4)
- Tlačítko „Ukázat cestu“ je nevýrazné (respondent č. 1, 2, 3, 5)
- Změna textace odznaků na ocenění (respondent č. 4, 5)
- Text ve widgetu odpovědi kuřáka je rozsáhlý a nabádá k přeskočení (respondent č. 2, 3)
- Matoucí název reporty (respondent č. 1, 3, 4, 5)
- V reportech a na jiných místech aplikace je použit malý font (respondent č. 3)
- Zbytečnost zobrazení emoticonů (respondent č. 1, 2)
- Hlavní obrazovka je dlouhá a nabádá k přílišnému scrollování (respondent č. 1, 2, 3, 4, 5)

8.2 Testování serverové části

V rámci testování serverové části aplikace se primárně zaměříme na Unit testy (jednotkové testování). Serverová část zatím není dostatečně velká a komplexní, aby pro nás měly přínos například integrační testy. Ty dávají smysl spíše při větším množství komponent a vývoji ve větším vývojářském týmu, kde se vyskytuje vyšší míra nekonzistence v implementaci logických úkolů. Zároveň nám současně databáze slouží primárně jako uložisko namockovaných dat, která nahráváme do databáze při spuštění serveru pomocí skriptu **data.sql**. Současné serverové řešení je určeno k budoucí integraci na serverové řešení Mobilní aplikace pro odvykání kouření. To však vyžaduje další implementační kroky ze strany vývojářů. Testování chování serverové části aplikace Buddy pomocí Unit testů je tak pro účely této diplomové práce dostatečné.

8.2.1 Unit testy - použité nástroje

Pro vytváření Unit testů používáme v současné chvíli nejnovější verze nástrojů JUnit Jupiter 5.10.1 a Mockito 5.3.1. JUnit Jupiter je součástí frameworku JUnit 5, který poskytuje sadu nástrojů pro testování v jazyce Java. Mezi klíčové vlastnosti tohoto frameworku patří anotační model, který obsahuje sadu anotací, jako jsou například `@Test`, `@BeforeEach`, `@AfterEach` a `@BeforeAll`. Ty slouží ke konfiguraci životního cyklu testů s větší flexibilitou. JUnit Jupiter také podporuje lambda výrazy, vnořené testovací třídy, parametrické testy nebo dynamické testy. [41]

Framework Mockito poskytuje vývojářům sadu nástrojů pro vytváření jednotkových testů. Zaměřuje se především na takzvané mockování² objektů. Důležitou vlastností frameworku Mockita je možnost integrace s frameworkem JUnit, kdy můžeme mockované objekty využít pro testování za pomoci frameworku JUnit, v našem případě JUnit Jupiter. Chování mockovaných objektů můžeme pomocí Mockita definovat řetězením poskytovaných metod s určitým typem chování. [42]

Unit testy servisní vrstvy

Pro jednotkové testování jsme zvolili třídy, které reprezentují v našem serverovém řešení servisní vrstvu. Jsou jimi `BadgeService`, `MessageService`, `ReportService` a `SmokerAnswerService`. Servisní vrstva obsahuje většinu logiky naší aplikace a je vhodné ji podrobit jednotkovému testování. Ostatní třídy jako Entity, DTO objekty, Exceptiony nebo Controllery jsou buď pouhými nositeli informací, nebo jsou navrženy tak, že přes jejich jednoduchost pro ně jednotkové testování nedává smysl.

Testování servisní vrstvy pro načítání odznaků se nachází v testovací třídě `BadgeServiceTest`. Zde jsme si vytvořili metody, které pokrývají všechny možné kombinace výstupů. Naším cílem je otestovat všechny možnosti, abychom tak pokryli veškeré průchody. Jednou z testovacích metod je `testGetAllBadges_WhenSmokerExistsAndHasBadges()`. Tato metoda definuje chování dvou namockovaných tříd, kterými jsou `SmokerRepository` a `SmokerAcquiredBadgeRepository`. Ty jsou v testovací třídě instanciované pomocí anotace `@Mock`, která z nich vytváří fiktivní třídy. Poté můžeme použít funkce frameworku Mockito `when()` a `thenReturn()`, abychom nadefinovali jejich chování.

```
@Test
public void testGetAllBadges_WhenSmokerExistsAndHasBadges() {

    Integer smokerId = 1;
    when(smokerRepository.existsById(smokerId)).thenReturn(true)
    ;
    when(smokerAcquiredBadgeRepository.findById(smokerId)
        ).thenReturn(Arrays.asList(
            mockBadge(1, Section.PREPARATION),
```

²Mockování - technika, při které se vytváří fiktivní data reprezentující skutečné objekty nebo data

```

        mockBadge(2, Section.CESSATION),
        mockBadge(3, Section.NO_CIG)
    ));

    SmokerBadgesAcquiredResponseDTO response = badgeService.
        getAllBadges(smokerId);

    assertEquals(List.of(1), response.preparationBadges());
    assertEquals(List.of(2), response.cessationBadges());
    assertEquals(List.of(3), response.noCigBadges());
}

```

Řetězec `when(smokerRepository.existsById(smokerId)).thenReturn(true)` konkrétně simuluje stav, kdy `SmokerRepository` vrátí hodnotu `true`, tedy že se objekt `Smoker` s příslušným `smokerId` v databázi nachází. Druhý řetězený konstrukt poté definuje případ, když `SmokerAcquiredBadgeRepository` z databáze vrátí seznam třech odznaků. Nakonec zavoláme testovanou servisu s metodou `getAllBadges(smokerId)`, která by nám podle nadefinovaného chování měla vrátit očekávaný objekt `SmokerBadgesAcquiredResponseDTO`, pokud je metoda implementována správně. Toto nakonec ověříme pomocí metody `assertEquals()`, která porovnává očekávaný a skutečný výstup. Pokud jsou oba výstupy shodné, test se provede úspěšně.

V souvislosti s jednotkovým testováním se ještě podíváme na metodu `testGetFirstUnviewedSmokerAnswer_NoUnviewedAnswerExists()`. V této metodě je zajímavostí, že očekávaným výstupem není datový objekt, ale vyhození výjimky. Testujeme zde totiž případ, kdy dochází k dotazu na vrácení nepřechtené odpovědi kuřáka. Žádná nová odpověď ale v databázi není, a tak `SmokerAnswerService` vyhodí výjimku, kterou controller odchytlí a umístí její zprávu do návratové zprávy.

```

@Test()
public void
    testGetFirstUnviewedSmokerAnswer_NoUnviewedAnswerExists() {

    Integer buddyId = 1, smokerId = 1;
    Buddy buddy = new Buddy();
    ...
    BuddySmokerPairing pairing = new BuddySmokerPairing();
    ...
    NextAnswerTimer nextAnswerTimer = new NextAnswerTimer();
    ...
    when(buddySmokerPairingRepository.findByBuddy_IdAndSmoker_Id
        (buddyId, smokerId)).thenReturn(Optional.of(pairing));
    when(nextAnswerTimerRepository.findByPairing_Id(pairing.
        getId())).thenReturn(Optional.of(nextAnswerTimer));
    when(smokerAnswerRepository.
        findFirstByPairing_IdAndViewedFalseOrderByDateTimeAsc(
            pairing.getId())).thenReturn(Optional.empty());

    assertThrows(SmokerAnswerNoContentException.class, () ->
        smokerAnswerService.getFirstUnviewedSmokerAnswer(buddyId,
            smokerId));
}

```

K otestování správnosti vyhozené výjimky slouží metoda **assertThrows**. Ta porovnává očekávaný literál třídy definující výjimku a výjimku vyhozenou z testované metody, v našem případě metody **getFristUnviewedSmokerAnswer**. Je nutné zmínit, že implementace využívá lambda výrazu, protože kdybychom metodu volali přímo jako vstupní parametr pro **assertThrows**, byl by výsledek Javou vyhodnocen dříve, než by došlo k samotnému zavolání metody **assertThrows**, která uvnitř své definice ve svých rodičovských třídách má implementovaný konstrukt **try..catch**. Tímto způsobem testujeme správně vyhozené výjimky ze servisní vrstvy i ve zbytku jednotkových testů.

Kapitola 9

Budoucí práce

Tato práce si kladla za cíl implementovat řadu funkcionalit pro mobilní aplikaci Buddy a nové serverové řešení, které podpoří svou flexibilitou budoucí vývoj této mobilní aplikace. Výstup této práce je však stále prototypem, který v současné chvíli není možné nasadit na produkční prostředí a uvést do provozu. To je dáno absencí potřebného rozhraní pro integraci na serverové řešení Mobilní aplikace pro odvykání kouření. Z toho důvodu se v této kapitole stručně zmíníme o funkcionalitách, které bude nutné v budoucnu implementovat, aby bylo dosaženo vytvoření komplexního systému, který bude schopen plně funkčnosti.

Integrace na serverovou část Mobilní aplikace pro odvykání kouření. V budoucí práci je nutné navrhnout a implementovat rozhraní pro integraci se serverovou částí Mobilní aplikace pro odvykání kouření. Toto rozhraní musí zajistit poskytnutí všech potřebných parametrů pro zobrazení aktuálních informací o kuřákovi. Můžeme zmínit například nutnost poskytnout odpovědi z kuřákových sezení. Klíčové bude také implementovat na straně Mobilní aplikace pro odvykání kouření funkci potvrzení o poskytnutí odpovědi ostatním osobám. Odpovědi mohou být často osobní a citlivé a nelze je tedy poskytovat bez vědomí a souhlasu kuřáka dalším osobám. S tím je spojené také šifrování odpovědí pro případ nechtěného úniku dat.

Autentizace a autorizace. V mobilní aplikaci Buddy je implementována základní integrace se serverovou částí Mobilní aplikace pro odvykání kouření pro registraci a přihlašování uživatelů. Je však nutné analyzovat, zda nebude výhodnější tuto funkcionalitu delegovat na serverové řešení aplikace Buddy. Nutností je totiž implementace autentizace a autorizace uživatele mobilní aplikace Buddy jednak pro zabezpečení vystavených endpointů serverovou částí, ale také pro autorizaci budoucích požadavků.

Nasazení serverové části do infrastruktury Mobilní aplikace pro odvykání kouření. Pro účely této diplomové práce bylo k serverovému řešení Mobilní aplikace pro odvykání kouření stejně tak k celé infrastruktuře přistupováno jako k „black boxu“. V budoucnu však bude potřeba tuto aplikaci nasadit na produkční prostředí. S tím souvisí případné úpravy a konfigurace aplikace podle parametrů prostředí.

Údržba mobilní aplikace Buddy. Aby mobilní aplikace Buddy setrvala v budoucnu aktualizovaná, bude nutné ji pravidelně udržovat. Prostředí vývoje frontendových nástrojů se v dnešní době vyvíjí velmi rychle a aplikace se bez pravidelné údržby může snadno stát zastaralou.

Přepis aplikace do Typescriptu. V současnosti mezi vývojáři je preferovanou volbou Typescript, jehož největší výhodou je statická typová kontrola. To umožňuje odhalovat chyby spojené s typy proměnných již při kompilaci. Typescript je též zpětně kompatibilní s Javascriptem, což umožňuje paralelně vyvíjet nové funkcionality v Typescriptu a provádět postupný přepis aplikace z Javascriptu. V počáteční fázi této diplomové práce byla vize původní aplikaci přepsat do Typescriptu, ale bohužel se ukázalo, že se jedná o velmi časově náročnou úlohu z důvodu struktury jednotlivých souborů. Proto bylo v rané fázi od této vize upuštěno.

Kapitola 10

Závěr

Cílem této diplomové práce bylo implementovat nadefinované požadavky pro mobilní aplikaci Buddy a poskytnout zcela nové řešení serverové části pro její efektivnější správu a flexibilitu pro vývoj nových požadavků.

Na začátku jsme se seznámili s problematikou kouření a odvykání. Podívali jsme se na historii tabáku a rozebrali jsme si statistiky úmrtnosti na kouření a počet prodaných cigaret v Česku i ve světě. Popsali jsme také dostupná opatření pro redukci kouření a podívali se na výrobky určené ke kouření. V poslední části kapitoly jsme si rozebrali metody odvykání kouření.

V další kapitole jsme si rozebrali trendy vývoje mobilních aplikací a vysvětlili si pojmy jako nativní a multiplatformní vývoj. Popsali jsme si nejpobulárnější platformy, pro které dnes probíhá vývoj mobilních aplikací a seznámili jsme se se dvěma nejrozšířenějšími frameworky pro řešení multiplatformního vývoje - Flutter a React Native.

V realizační části diplomové práce jsme nejprve ve čtvrté kapitole provedli analýzu nad existující aplikací Buddy, popsali si její funkcionality a rozebrali použité technologie. V páté kapitole jsme navázali návrhovou částí, v níž jsme si nejprve nadefinovali požadavky pro implementaci funkcionalit mobilní aplikace Buddy. Tyto požadavky jsme zasadili do kontextu existující analýzy z předchozího vývoje aplikace a sestavili nové funkční požadavky pro její doplnění. Dále jsme v souvislosti s návrhem vytvořili definici zpráv pro komunikaci frontendové části se serverovou částí. Navazovala návrhová část pro serverový systém. Zde jsme odůvodnili volbu technologií a navrhli jsme si pomocí UML diagramů důležité části systému.

V šesté kapitole věnující se implementaci jsme se podrobně podívali na rozbor zajímavých implementačních pasáží a to jak ze strany mobilní aplikace Buddy, tak serverové části. V neposlední řadě jsme výsledek implementace podrobili testování. Nad mobilní aplikací Buddy jsme provedli moderované testování s uživateli, které doplňuje předchozí testování aplikace v minulých pracích. Pro serverovou část jsme vytvořili sadu jednotkových testů. V poslední, deváté kapitole jsme si stručně popsali návrh budoucích implementačních kroků, které je nutné implementovat k dosažení úspěšného nasazení aplikace na produkční prostředí.

Cíle této diplomové práce se povedlo naplnit. Implementované serverové řešení i přes to, že v současnosti nenabízí integraci se serverovou částí Mobilní

aplikace pro odvykání kouření, tvoří dobrý základ pro budoucí rozvoj mobilní aplikace Buddy. Nutností však zůstává vytvoření vhodného rozhraní pro komunikaci mezi oběma backendy. Jen v takovém případě se skutečně bude moci využít plný potenciál, který se v aplikaci Buddy skrývá.



Literatura

- [1] Shanu Mishra, M.B. Mishra. *Tobacco: Its historical, cultural, oral, and periodontal health association*, [online]. Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3894096/>
- [2] Steven I. Hajdu, Manjunath S. Vadmal. *A note from History: The use of Tobacco*, [online] Dostupné z: <http://www.annclinlabsci.org/content/40/2/178.full.pdf>
- [3] World Health Organization. *Chapter 32, History of Tobacco*, [online] Dostupné z: <https://www.afro.who.int/sites/default/files/2017-09/Chapter%2032.%20The%20history%20of%20tobacco.pdf>
- [4] Office of the Commissioners of patents for inventions (Velká Británie, 1870). *Patents for inventions. Abridgments of specifications*, [online] Dostupné z: Google Books - https://www.google.cz/books/edition/Patents_for_inventions_Abridgments_of_sp/xoThFs0ZfskC?hl=cs&gbpv=1
- [5] David M. Burns, Lora Lee, Larry Z. Shen, Elizabeth Gilpin, H. Dennis Tolley, Jerry Vaughn, and Thomas G. Shanks. *Cigarette Smoking Behavior in the United States*, [online] Dostupné z: https://www.researchgate.net/profile/David-Burns-11/publication/313199597_Cigarette_smoking_behavior_in_the_United_States/links/595bc038458515117741b4e2/Cigarette-smoking-behavior-in-the-United-States.pdf
- [6] Dai X, Gakidou E, Lopez AD. *Tob Control 2022;31:129–137*, [online] Dostupné z: <https://tobaccocontrol.bmj.com/content/tobaccocontrol/31/2/129.full.pdf>
- [7] Hannah Ritchie, Max Roser. *Smoking*, [online] Dostupné z: <https://ourworldindata.org/smoking>
- [8] World Health Organization. *WHO Report on the Global Tobacco Epidemic, 2008, The MPOWER package*, [online] Dostupné z: https://apps.who.int/iris/bitstream/handle/10665/43818/9789241596282_eng.pdf?sequence=1

- [9] World Health Organization. *WHO Report on the Global Tobacco Epidemic, 2021, Addressing new and emerging products*, [online] Dostupné z: <https://www.who.int/publications/i/item/9789240032095>
- [10] Mravčík, V., Chomynová, P., Grohmannová, K., Janíková, B., Černíková, T., Rous, Z., Cibulka, J., Franková, E., Nechanská, B., Fidesová, H., 2021. *Zpráva o tabákových, nikotinových a souvisejících výrobcích v České republice 2021*, [online] Úřad vlády České republiky. Dostupné z: https://www.drogy-info.cz/data/obj_files/33573/1121/Zprava_o_tabakovych_nikotinovych_vyrobcich_v_CR_2021_fin.pdf, ISBN: 978-80-7440-276-0
- [11] Csémy L., Dvořáková Z., Fialová A., Kodl M., Malý M., Skývová M. pro Státní zdravotní ústav, 2022. *Národní výzkum užívání tabáku v České republice 2021 [NAUTA]*, [online], Dostupné z: <https://szu.cz/wp-content/uploads/2023/03/Narodni-vyzkum-uzivani-tabaku-a-alkoholu-v-Ceske-republice-2021.pdf>, ISBN: 978-80-7071-419-5
- [12] ČESKO. Vyhláška č. 261/2016 Sb., Vyhláška o tabákových výrobcích. *Sbírka zákonů*, [online] Dostupné z <https://www.zakonyprolidi.cz/cs/2016-261>
- [13] Intersog. *How Many Mobile App Developers Are There in the World?*, [online] Dostupné z: <https://intersog.com/blog/how-many-developers-are-there-in-the-world/>
- [14] Lionel Sujay Vailshery. *Developers population worldwide 2018-2024*, [online] Dostupné z: <https://www.statista.com/statistics/627312/worldwide-developer-population/>
- [15] Solomon Amar - Allstarsit blog. *How Many Software Engineers are There in the World?*, [online] Dostupné z: <https://www.allstarsit.com/blog/how-many-software-engineers-are-there-in-the-world>
- [16] Statista Research Department. *Mobile app revenue worldwide 2019-2027, by segment*, [online] Dostupné z: <https://www.statista.com/forecasts/1262892/mobile-app-revenue-worldwide-by-segment>
- [17] Jacob Schmitt. *Native vs cross-platform mobile app development, 2023*, [online] Dostupné z: <https://circleci.com/blog/native-vs-cross-platform-mobile-dev/>
- [18] Ajay Chebbi. *Choosing the best programming language for mobile app development, 2021*, [online] Dostupné z: <https://developer.ibm.com/articles/choosing-the-best-programming-language-for-mobile-app-development/>

- [19] Lionel Sujay Vailshery, Statista. *Cross-platform mobile frameworks used by software developers worldwide from 2019 to 2022, 2023*, [online] Dostupné z: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>
- [20] Catherine A., Haighton, Dorothy Newbury-Birch, Eileen F.S. Kaner. *Interventions for Addiction, Academic Press, 1st edition, 2013*, ISBN: 978-0123983381
- [21] Farkáš T. *Buddy aplikace pro odvykání kouření, Bakalářská práce, České vysoké učení technické, Fakulta elektrotechnická, 2021*, [online] Dostupné z: <https://dspace.cvut.cz/bitstream/handle/10467/94411/F3-BP-2021-Farkas-Tomas-bachelor-thesis.pdf?sequence=-1&isAllowed=y>
- [22] Statistics and Data. *Most popular backend frameworks 2012-2023, 2023*, [online] Dostupné z: <https://statisticsanddata.org/data/most-popular-backend-frameworks-2012-2023/>
- [23] Expo. *Expo Docs SplashScreen*, [online] Dostupné z: <https://docs.expo.dev/versions/latest/sdk/splash-screen/>
- [24] mdn web docs. *Using the Fetch API*, [online] Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch
- [25] mdn web docs. *await*, [online] Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/await>
- [26] mdn web docs. *Promise*, [online] Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Promise
- [27] mdn web docs. *try...catch*, [online] Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Statements/try...catch>
- [28] React Dev. *useState*, [online] Dostupné z: <https://react.dev/reference/react/useState#usestate>
- [29] React Dev. *useEffect*, [online] Dostupné z: <https://react.dev/reference/react/useEffect>
- [30] Steve Krug. *Don't make me think, Revisited, A Common Sense Approach to Web Usability, 2014*, ISBN: 978-0-321-96551-6
- [31] Jakob Nielsen, Nielsen Norman Group. *Why You Only Need to Test with 5 Users, 2000*, [online] Dostupné z: <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>

- [32] Java T Point. *Java Cron Expression*, [online] Dostupné z: <https://www.javatpoint.com/java-cron-expression>
- [33] Spring Docs. *Annotation Interface Scheduled*, [online] Dostupné z: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/scheduling/annotation/Scheduled.html>
- [34] Spring Docs. *Annotation Interface Component*, [online] Dostupné z: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/stereotype/Component.html>
- [35] Spring Docs. *Annotation Interface RequestMapping*, [online] Dostupné z: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/bind/annotation/RequestMapping.html>
- [36] Spring Docs. *Annotation Intergace RestController*, [online] Dostupné z: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/bind/annotation/RestController.html>
- [37] Java Oracle Docs. *Class Throwable*, [online] Dostupné z: <https://docs.oracle.com/javase/8/docs/api/java/lang/Throwable.html>
- [38] Spring Docs. *Annotation Interface ExceptionHandler*, [online] Dostupné z: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/bind/annotation/ExceptionHandler.html>
- [39] Spring Docs. *Annotation Interface ResponseStatus*, [online] Dostupné z: <https://docs.spring.io/spring-framework/docs/current/javadoc-api/org/springframework/web/bind/annotation/ResponseStatus.html>
- [40] Spring Docs. *Introduction to the Spring IoC Container and Beans*, [online] Dostupné z: <https://docs.spring.io/spring-framework/reference/core/beans/introduction.html>
- [41] JUnit 5 Docs. *JUnit 5 User Guide*, [online] Dostupné z: <https://junit.org/junit5/docs/current/user-guide/>
- [42] Java Docs. *Java Docs - Mockito*, [online] Dostupné z: <https://javadoc.io/doc/org.mockito/mockito-core/latest/org/mockito/Mockito.html>
- [43] Flutter. *Flutter architectural overview*, [online] Dostupné z: <https://docs.flutter.dev/resources/architectural-overview>
- [44] React Native. *React Native Dev Documentation*, [online] Dostupné z: <https://reactnative.dev/docs/getting-started>

- [45] React Native. *React Native Dev Documentation Built-in React Hooks*, [online] Dostupné z: <https://react.dev/reference/react/hooks>
- [46] Expo. *Expo Docs*, [online] Dostupné z: <https://docs.expo.dev/>
- [47] React Native. *React Native AsyncStorage*, [online] <https://reactnative.dev/docs/asyncstorage>
- [48] i18next. *Documentation Introduction*, [online] Dostupné z: <https://www.i18next.com/>
- [49] Erik Cupal. *Czech inflection JavaScript library*, [online] Dostupné z: <https://github.com/ErikCupal/czech-inflection/tree/master>
- [50] Ramda. *Ramda Documentation v0.19.1* [online] Dostupné z: <https://ramdajs.com/0.19.1/docs/>
- [51] Redux Js. *Getting Started With Redux* [online] Dostupné z: <https://redux.js.org/introduction/getting-started>
- [52] Národní stránky pro podporu odvykání kouření. *Proč přestat*, [online] Dostupné z <https://www.koureni-zabiji.cz/proc-prestat/>
- [53] MUDr. Jarmila Pradáčová, CSc. *Obecné informace pro pacienty - Kouření a zdraví, Liga proti rakovině Praha*, [online] Dostupné z: https://www.cdpr.cz/wp-content/uploads/2023/01/Koureni-a-zdravi_web.pdf
- [54] Loono. *Kouření a jeho vliv na zdraví*, [online] Dostupné z: <https://www.loono.cz/koureni-a-jeho-vliv-na-zdravi>
- [55] 1. lékařská fakulta Univerzity Karlovy. *Málo známé klinické souvislosti kouření*, [online] Dostupné z: <https://www.lf1.cuni.cz/malo-zname-klinicke-souvislosti-koureni>
- [56] European Monitoring Centre for Drugs and Drug Addiction. *ESPAD report 2019 – Results from the European school survey project on alcohol and other drugs*, Publications Office, 2020, [online] Dostupné z: <https://data.europa.eu/doi/10.2810/877033>
- [57] Australian Government, Department of Health and Aged Care. *Quitting methods, 2023*, [online] Dostupné z: <https://www.health.gov.au/topics/smoking-vaping-and-tobacco/how-to-quit/quitting-methods>
- [58] MedlinePlus. Bethesda (MD): National Library of Medicine (US). *Nicotine replacement therapy, 2023*, [online] Dostupné z: <https://medlineplus.gov/ency/article/007438.htm>
- [59] Wadgave U, Nagesh L. *Nicotine Replacement Therapy: An Overview*. *Int J Health Sci (Qassim)*, 2016, PMID: 27610066; PMCID: PMC5003586, [online] Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5003586/>

- [60] Fang YE, Zhang Z, Wang R, Yang B, Chen C, Nisa C, Tong X, Yan LL *Effectiveness of eHealth Smoking Cessation Interventions: Systematic Review and Meta-Analysis. J Med Internet Res, 2023, PMID: 37505802; PMCID: PMC10422176. [online] Dostupné z: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10422176/>*



Přílohy



Příloha A

Seznam zkratk

ANDS	Alternative Nicotine Delivery Systems
API	Application programming interface
DTO	Data Transfer Object
ENDS	Electronic Nicotine Delivery Systems
ENNDS	Electronic Non-Nicotine Delivery Systems
FEL	Fakulta elektrotechnická
HTP	Heated Tobacco Products
HTTP	Hypertext Transfer Protocol
IDE	Integrated development environment
JPA	Java persistence API
JVM	Java Virtual Machine
NRT	Noctine Replacement Therapy
ORM	Object-relational Mapping
REST	Representational State Transfer
SDK	Software development kit
SQL	Structured Query Language
UML	Unified Modeling Language
ČVUT	České vysoké učení technické

Příloha B

Odovědi respondentů z testovacího dotazníku

Scénář 1: Zobrazte si seznam odznaků, které kuřák získal během svého odvykání. Napište poznatky k tomuto průchodu aplikací. Co se vám líbilo? Co se vám nelíbilo? Byl průchod snadný a intuitivní? Co byste změnil/a?

Tabulka B.1: Odpovědi ke scénáři č. 1

Respondent	Odpověď
1	Ocenuji rozdělení na 3 části. Líbí se mi grafická stránka jednotlivých odznaků. Chybí mi možnost rozkliknutí jednotlivých odznaků, kde bych očekával jejich přesnou definici (za co je exkurak získal).
2	Martin má 14 odznaků. Odznaky byly poměrně skryté v okně ukázat cestu. Po kliknutí sem však již bylo zřejmé co byly odznaky. Odznaky byly umístěny na smysluplném místě ale pro člověka co vidí aplikaci úplně poprvé není zřejmé na první pohled kde se nachází.
3	Změnil bych názvosloví Ukázat cestu. Nevím na co. Myslel jsem si, že tam hledat odznaky, ale nejprve jsem se podíval jinam.
4	Nevím co znamená už se známe. Popisy odznaků jsou nejasné. Vizuálně hezké
5	Lepší by bylo, kdyby se to jmenovalo ocenění, jako třeba v aplikaci kondice na iphone. Intuitivně bych to zde nehledala, spíše náhodný nález.

Scénář 2: Přečtete si poslední odpověď z kuřákova sezení a potvrďte její přečtení. Napište poznatky k tomuto průchodu aplikací. Co se vám líbilo? Co se vám nelíbilo? Byl průchod snadný a intuitivní? Co byste změnil/a?

Tabulka B.2: Odpovědi ke scénáři č. 2

Respondent	Odpověď
1	Přístup jednoduchý a intuitivní, nic bych neměl. Tlačítko "potvrdit" považuji za vhodný krok. K uzavření bych dal, zda si "exkurak Martin" nezaslouží kromě pochvaly a podpory i radu protistrany.
2	První okno, které jsem otevřel byla historie odpovědí jelikož bezmyšlenkovitě odscrolloval moc daleko. Po konzultaci s moderátorem mi bylo řečeno, že se poslední odpověď ještě v historii nenachází. Toto mi připadá lehce neintuitivní. Historie by dle mého názoru měla obsahovat i nepřečtené odpovědi.
3	Nemohl jsem to najít. Poslední odpověď jsem hledal na konci aplikace a nemohl jsem najít konec.
4	Myslím si, že třetí osoba nemůže kuřákovi pomoci. Když budu pravidelně sledovat průběh odpovědí, tak asi když narazím na nějaký problém, pak kuřákovi pomůžu
5	Odpověď je jednoznačně přístupná. Dobře strukturovaná. Líbí se mi design i ikonka.

Scénář 3: Zobrazte si historii předchozích odpovědí kuřáka. Napište poznatky k tomuto průchodu aplikací. Co se vám líbilo? Co se vám nelíbilo? Byl průchod snadný a intuitivní? Co byste změnil/a?

Tabulka B.3: Odpovědi ke scénáři č. 3

Respondent	Odpověď
1	Kategorie "historie" se mi ze své podstaty líbí, připadá mi důležité mít možnost vrátit se k předchozím "seznamům".
2	Předchozí odpověď je validní i k tomuto scénáři. Tlačítko je umístěné na logické místě. Jediná výtko viz předchozí scénář. Lidé často mění názor, obzvláště když se jedná o nepříjemné záležitosti. Funkce zobrazení předchozích odpovědí smysl dává.
3	Když si vše pozorně přečtu, tak se k tomu nakonec dostanu.
4	Nevím co bych dodala.
5	Trvalo mi než jsem to našla, je to úplně dole, ale je to jasné.

Scénář 4: Zobrazte si statistiky za uplynulé týdny. Napište poznatky k tomuto průchodu aplikací. Co se vám líbilo? Co se vám nelíbilo? Byl průchod snadný a intuitivní? Co byste změnil/a?

Tabulka B.4: Odpovědi ke scénáři č. 4

Respondent	Odpověď
1	Kategorie statistiky se mi líbí, je jednoduchá a přehledná. Ke zvážení bych dal, zda tam není kolonka pro počet vykouřených cigaret za dané období.
2	Report bylo snadné nalézt. Nebylo ale zřejmé že možné horizontálně skrollovat. V aplikaci jsem neměl žádný náznak této funkcionality a tudíž jsem ji objevil až na radu moderátora. Ocenil bych vizuální hint že je v tomto okně možné horizontálně scrollovat.
3	Našel jsem hned, ale nepřičtu jednotlivé statistiky. Písmo je moc malé.
4	To jsou dobré a vypovídající statistiky. Líbí se mi to.
5	Platí totéž co při historii odpovědí

Scénář 5: Podívejte se na kuřákův postup na roadmapě na hlavní stránce aplikace a poté si přečtete detail o údržovací fázi. Napište poznatky k tomuto průchodu aplikací. Co se vám líbilo? Co se vám nelíbilo? Byl průchod snadný a intuitivní? Co byste změnil/a?

Tabulka B.5: Odpovědi ke scénáři č. 5

Respondent	Odpověď
1	Roadmapa se mi líbí, je přehledná a jednotlivé fáze jsou po rozkliknutí jednoznačně definovány.
2	Zde nemám žádnou výtku. Pochválil bych grafiku roadmapy. Je na první pohled zřejmé co je klikatelné a co ne.
3	Vizuálně hezké
4	Mapa se mi moc líbí.
5	Též vše jednoznačné. Graficky přívětivé. Líbí se mi hravé obrázky.

Scénář 6: Pošlete kuřákovi pozitivní zprávu, že mu držíte palce a že ho plně podporujete v jeho odvykání. Použijte alespoň jeden emoticon. Napište poznatky k tomuto průchodu aplikací. Co se vám líbilo? Co se vám nelíbilo? Byl průchod snadný a intuitivní? Co byste změnil/a?

Tabulka B.6: Odpovědi ke scénáři č. 6

Respondent	Odpověď
1	Průchod byl snadný a intuitivní, mám rozporuplné pocity z výzvy k použití emotikonu.
Pokračuje na další stránce	

Tabulka B.6 pokračování z předchozí stránky

Otázka	Odpověď
2	Funkce posláni zprávy je intuitivní a je hned zřejmé kam je potřeba kliknout. Připadá mi ale zbytečné mít na odesílacím formuláři zprávy emotikony jelikož většina moderních zařízení podporuje emotikony přímo z nativní klávesnice.
3	Tlačítko poslat zprávu je hrozně malé, nemohl jsem ho najít, ale to je spíše moje chyba.
4	Posláni zprávy, moc hezká věc. Připadá mi to jako nejdůležitější funkce aplikace.
5	Líbí se mi, že je to na standardním místě, kde jsou dnes lidé zvyklí zprávy hledat jako na webu chat.

Zdá se Vám aplikace Buddy přehledně řešená?

Tabulka B.7: Dotaz na přehlednost aplikace

Respondent	Odpověď
1	Ano
2	Ano, ale mohlo by to být lepší
3	Ano, ale mohlo by to být lepší
4	Ano
5	Ano, ale mohlo by to být lepší

Popište vlastními slovy pozitivní poznatky k přehlednosti aplikace.

Tabulka B.8: Pozitivní poznatky k přehlednosti aplikace

Respondent	Odpověď
1	Velmi ocenuji Roadmapu, celkove i rozdeleni do kategori historie odpovedi, reporty, opakovani sezeni mi pripada prehledne.
2	Pozitivně bych určitě hodnotil design aplikaci. Působí moderně. Designové nedostatky jsou spíše maličkosti.
3	Líbí se mi mapa. Nevím co jiného.
4	Aplikace je jednoduchá, což je dobře.
5	Hezky barevná a veselá aplikace. Vyvolává motivační pocit. Roadmapa je krásná a zobrazuje dobře průběh léčby. Aplikace má širokou škálu funkcí, historie, zprávy. Je to super.

Popište vlastními slovy nedostatky v přehlednosti aplikace.

Tabulka B.9: Negativní poznatky k přehlednosti aplikace

Respondent	Odpověď
1	Do kategorie historie odpovědi bych přidal dataci jednotlivých odpovědí.
2	Největší nedostatek mi přišla absence visual aid u horizontálního scrollování. Bez moderátora by mě nenapadlo že je možné v daném okně horizontálně scrollovat.
3	Zvolil bych jiná názvosloví. Hlavně to první - ukázat cestu. Malej text, ale mám špatný zrak. Možná když se aplikací listuje, tak by se text mohl zvětšovat.
4	Přepnout kuřáka bych změnila na změnit kuřáka
5	Přijde mi, že je toho moc pod sebou. Než se někam dostanu, trvá to hodně dlouho, lepší by bylo, kdyby byly jednotlivé části spíše přístupné k rozkliknutí ve 3 čárkách vlevo nahoře než abych musela tolik scrollovat. Reporty bych pojmenovala jako statistiky. Je to více přirozené.

Jaké funkcionality se Vám na aplikaci líbily nejvíce a proč?

Tabulka B.10: Výčet funkcionalit, které se líbily

Respondent	Odpověď
1	Nejvic se mi líbí Roadmapa. Vysvětluje vse podstatne o procesu odvykani, jaky typ podpory "Martin" v ktere fazi nejvice potrebuje a o cem vlastne ta ktera faze je. Navic je hezky a jednoduse zpracovana po graficke strance.
2	Funkce pochválení a možnost sledovat roadmapu.
3	Líbí se mi posílání zpravy. Skvělé je zobrazení částek kolik se ušetří. To se mi moc líbí.
4	Poslání zprávy, statistiky.
5	Ukázat cestu a poslat zprávu. Dobré je zobrazení peněz.

Jaké funkcionality se Vám na aplikaci nelíbily a proč?

Tabulka B.11: Výčet funkcionalit, které se nelíbily

Respondent	Odpověď
1	Nemoznost rozkliknuti jednotlivych odznaku za ucelem ziskani jejich presne definice.
2	Z funkcionality mi přišlo vše v pořádku. Vytknul bych akorát nejednotnost v designu v "hamburger menu". To designově vypadá jakoby náleželo dojiné aplikace.
3	Malý text.
Pokračuje na další stránce	

Tabulka B.11 pokračování z předchozí stránky

Otázka	Odpověď
4	Nemohu říct, že je něco, co by se mi vyloženě nelíbilo.
5	Přijde mi trochu zbytečné "jak to bude probíhat", i když chápu, že je to důležité pro pochopení jak kuřáka podporovat. Možná by bylo lepší to posunout až někam dolu, ale to je subjektivní

Pokud by Vás kuřák oslovil s žádostí o pomoc s jeho odvykáním a poslal Vám pozvánku do aplikace Buddy. Přijal byste ji a aplikaci pravidelně používal/a?

Tabulka B.12: Dotaz na přijetí pozvánky do aplikace

Respondent	Odpověď
1	Ano
2	Ano
3	Ano
4	Ano
5	Ano