



**ČESKÉ VYSOKÉ  
UČENÍ TECHNICKÉ  
V PRAZE**

**F3**

**Fakulta elektrotechnická  
Katedra počítačů**

**Bakalářská práce**

# **Serverové řešení aplikace Buddy App pro odvykání kouření**

**Vojtěch Novotný**

**Květen 2021**

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



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Novotný** Jméno: **Vojtěch** Osobní číslo: **466339**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávající katedra/ústav: **Katedra počítačů**  
Studijní program: **Otevřená informatika**  
Studijní obor: **Software**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Servrové řešení aplikace Buddy App pro odvykání kouření**

Název bakalářské práce anglicky:

**Server solution for Smoking Cessation Buddy App**

Pokyny pro vypracování:

- 1) Seznamte se s problematikou odvykání kouření.
- 2) Designujte servoevrové řešení BuddyApp aplikace o nové funkcionality. Aplikace by měla zahrnovat následující funkcionality: plán odvykacího procesu, statistiky a zaslání 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) bakalářské práce:

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

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

Datum zadání bakalářské práce: **15.02.2021**

Termín odevzdání bakalářské práce: \_\_\_\_\_

Platnost zadání bakalářské práce: **19.02.2023**

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

\_\_\_\_\_  
podpis vedoucí(ho) ústavu/katedry

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

## III. PŘEVZETÍ ZADÁNÍ

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

\_\_\_\_\_  
Datum převzetí zadání

\_\_\_\_\_  
Podpis studenta



## Poděkování / Prohlášení

Chtěl bych poděkovat své rodině a přátelům za podporu během celého studia. Také děkuji svému vedoucímu doc. Ing. Danielu Novákovi, Ph.D. za podporu při vypracovávání této práce.

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

V Praze dne 21. 5. 2021

.....

## Abstrakt / Abstract

Tato bakalářská práce se zabývá vývojem multiplatformních mobilních aplikací a návrhem serverové části aplikace Buddy App pro odvykání kouření. Výstupem práce je implementace 6 miniher ve frameworku React Native a návrh a implementace serverové části aplikace Buddy App ve frameworku Django.

V závěru práce jsou popsány výsledky uživatelského testování aplikace Buddy App.

**Klíčová slova:** mobilní aplikace; Android; iOS; React Native; Django; odvykání kouření.

This bachelor thesis is focused on the development of multiplatform mobile applications and the design of the server part of the Buddy App for smoking cessation. The output of the work is the implementation of 6 minigames in the React Native framework and the design and implementation of the server part of the Buddy App application in the Django framework.

At the end of the work, the results of user testing of the Buddy App application are presented.

**Keywords:** mobile application; Android; iOS; React Native; Django; smoking cessation.

**Title translation:** Server solution for Smoking Cessation Buddy App

# Obsah /

<b>1 Úvod</b> .....	1	4.7 Srovnání s nativní aplikací ....	20
1.1 Motivace.....	1	<b>5 Serverové řešení aplikace Buddy App</b> .....	22
1.2 Cíle práce .....	1	5.1 Současný stav aplikace Buddy App .....	22
1.3 Struktura práce .....	1	5.2 Django.....	23
<b>2 Odvykání kouření</b> .....	3	5.2.1 Struktura aplikace ve frameworku Django .....	24
2.1 Kouření v ČR .....	3	5.3 Registrace uživatele v Buddy App.....	25
2.2 Závislost na kouření .....	3	5.3.1 Proces pozvání a registrace uživatele.....	25
2.3 Léčba .....	4	5.3.2 Doménový model .....	25
2.3.1 Farmakologická léčba .....	4	5.3.3 RESTful API a sekvencní diagram.....	26
2.3.2 Psychosociální a behaviorální intervence .....	5	5.3.4 Implementace .....	27
2.3.3 Využití technologií v léčbě (eHealth).....	5	5.4 Zobrazení pokroku v odvykání.....	29
<b>3 Vývoj multiplatformních mobilních aplikací</b> .....	6	5.5 Expo push notifikace .....	29
3.1 Motivace pro multiplatformní vývoj.....	6	<b>6 Uživatelské testování Buddy App</b> .....	31
3.2 Současná multiplatformní řešení .....	6	6.1 Průběh testování.....	31
3.3 React Native .....	8	6.2 Participanti .....	31
3.4 Vývojové prostředí Expo .....	9	6.3 Testovací dotazník .....	32
<b>4 Minihry v React Native</b> .....	10	6.3.1 Zadání.....	32
4.1 Navigace v aplikaci .....	10	6.3.2 Otázky .....	33
4.2 Hra 2048 .....	11	6.4 Výsledky.....	33
4.2.1 Funkční požadavky a uživatelské rozhraní.....	11	<b>7 Závěr</b> .....	36
4.2.2 Stav hry .....	12	7.1 Rozšíření aplikace.....	36
4.2.3 Animace .....	13	<b>Literatura</b> .....	37
4.3 Hra Reversi.....	13	<b>A Seznam zkratk</b> .....	39
4.3.1 Funkční požadavky a uživatelské rozhraní.....	14		
4.3.2 Stav hry .....	14		
4.3.3 Herní logika .....	15		
4.4 Hra Piškvorky.....	16		
4.4.1 Funkční požadavky a uživatelské rozhraní.....	16		
4.4.2 Stav hry a herní logika ..	16		
4.5 Hry Kulička a Labyrint .....	17		
4.5.1 Funkční požadavky a uživatelské rozhraní.....	17		
4.5.2 Game Engine.....	18		
4.6 Hra Bublíny .....	19		
4.6.1 Funkční požadavky a uživatelské rozhraní.....	19		
4.6.2 Implementace herní logiky .....	19		

## Tabulky / Obrázky

<b>5.1.</b> Struktura HTTP požadavků... 26	<b>2.1.</b> Fagerströmův test závislosti na cigaretách .....4
	<b>3.1.</b> Ukázka nativní aplikace a její PWA verze .....7
	<b>3.2.</b> Deklarace UI v JSX .....8
	<b>4.1.</b> Ukázka využití React Navigation ..... 10
	<b>4.2.</b> Hlavní menu Miniher ..... 11
	<b>4.3.</b> Hra 2048 ..... 12
	<b>4.4.</b> Knihovna Animated..... 13
	<b>4.5.</b> Hra Reversi ..... 14
	<b>4.6.</b> Hra Piškvorky ..... 16
	<b>4.7.</b> Hry Kulička a Labyrint ..... 17
	<b>4.8.</b> Pozice kuličky ..... 18
	<b>4.9.</b> Srážka objektů v Matter.js .... 18
	<b>4.10.</b> Srážka objektů v Matter.js .... 19
	<b>4.11.</b> Prohledávání grafu do hloubky ..... 20
	<b>5.1.</b> Sezení v Buddy App ..... 23
	<b>5.2.</b> Struktura frameworku Django . 24
	<b>5.3.</b> Doménový model..... 26
	<b>5.4.</b> Sekvenční diagram ..... 27
	<b>5.5.</b> Třída ApiInviteBuddy ..... 28
	<b>5.6.</b> Registrační formulář ..... 28
	<b>5.7.</b> Sezení v Buddy App ..... 29
	<b>5.8.</b> Expo push notifikace..... 30
	<b>6.1.</b> Participantí - pohlaví a věk.... 32
	<b>6.2.</b> Participantí - mobilní aplikace . 32
	<b>6.3.</b> Obtížnost registrace..... 33
	<b>6.4.</b> Výsledky dotazníku ..... 34
	<b>6.5.</b> Užitečnost aplikace..... 35



# Kapitola 1

## Úvod

### 1.1 Motivace

Užívání tabáku je jednou z největších zdravotních hrozeb, kterému jako lidstvo čelíme. V současné době tato tabáková epidemie zabíjí po celém světě více než 8 miliónů lidí ročně, což zhruba odpovídá polovině všech lidí, kteří tabák užívají. Zatímco více než 7 miliónů z těchto úmrtí je zapříčeno přímým užíváním tabáku, okolo 1.2 miliónu nekuřáků umírá na následky pouhého vystavování se kouření druhých lidí [1]. Na základě zmíněných statistik je zřejmé, že prevence a odvykání od užívání tabáku představuje v dnešním světě určitou výzvu. Zároveň se ukazuje, že úspěšnost odvykání bez řízené léčby je velice nízká [2].

Mezi některé druhy léčby patří i psychosociální a behaviorální intervence. V současné době existují mobilní aplikace simulující tyto intervence pomocí sezení s virtuálním terapeutem [3]. Na trhu však chybí aplikace, která by do průběhu odvykání začleňovala i přátele a rodinu kuřáka. Aplikace, která by informovala blízké kuřáka o tom, co zažívá a jak mu mohou pomoci, by mohla značně přispět k úspěšnému ukončení odvykacího procesu.

### 1.2 Cíle práce

Práce je rozdělena na dvě části. První část se zabývá implementací šesti miniher ve frameworku React Native jako rozšíření stávající aplikace pro odvykání kouření. Mini-hry mají poskytovat okamžité rozrušení a odvedení pozornosti od chuti na cigaretu v rámci kritické fáze odvykání. Cílem této části bude popsat výhody a nevýhody vývoje multiplatformní aplikace v tomto frameworku v porovnání s vývojem aplikací nativních.

Druhou část této práce tvoří návrh a implementace serverové části aplikace Buddy App ve frameworku Django. Jedná se o aplikaci určenou pro přátele kuřáka, která je informuje o procesu odvykání. Buddy App má být propojena se serverem stávající aplikace pro odvykání kouření, což umožní přátelům kuřáka sledovat jeho pokrok v procesu odvykání a účinněji ho podporovat. Server má poskytovat rozhraní pro pozvání přátele kuřáka do aplikace Buddy App, registraci nového uživatele, zobrazení pokroku kuřáka v odvykání a zasílání zpráv do Buddy App pomocí notifikací.

### 1.3 Struktura práce

Kapitola 2 popisuje základní metody léčby závislosti na kouření. Kapitola 3 pojednává o různých přístupech při vývoji multiplatformních aplikací. Poté se věnuje konkrétně frameworku React Native a sadě nástrojů, které poskytuje vývojové prostředí Expo. Kapitola 4 popisuje implementaci šesti her ve frameworku React Native. Je popsána aplikační logika jednotlivých her a použití základních knihoven. V kapitole 5 je popsán návrh a implementace serverového rozhraní pro propojení aplikace Buddy App s již

existující aplikací pro odvykání kouření. V kapitole 6 je popsán výsledek uživatelského testování aplikace Buddy App. Kapitola 7 obsahuje závěrečné zhodnocení práce.

## Kapitola 2

### Odvykání kouření

Tato kapitola se věnuje problematice odvykání kouření. V podkapitole 2.1 jsou popsány výzvy spojené s užíváním tabáku v České republice. Kapitola 2.2 rozebírá podstatu závislosti na kouření. Poslední podkapitola 2.3 se poté věnuje různým druhům léčby na psychosociální a behaviorální i farmakologické úrovni.

#### 2.1 Kouření v ČR

Užívání tabáku je jednou z hlavních příčin nemocnosti a úmrtnosti v ČR. Téměř 20% celkové úmrtnosti v ČR je způsobeno kouřením [4]. Kromě škody na zdraví způsobuje konzumace tabáku celospolečenské náklady související se snížením produktivity práce kuřáků (jako je zvýšená pracovní neschopnost, kuřácké pauzy nebo invalidní důchody).

Nedávná studie Státního zdravotního úřadu ukazuje, že v roce 2019 bylo pouhých 25% kuřáků starších 15 let [5]. Tento trend je v porovnání s dalšími evropskými státy nadprůměrný, kouření dětí a adolescentů v ČR se tak stává závažným problémem. Kouření nezletilých navíc předpovídá vývoj užívání tabáku v budoucnu. Ze studií totiž vyplývá, že kuřáci, kteří začali s kouřením v dřívějším věku, mají menší pravděpodobnost, že odvyknou [6].

V České republice se zdá být ochrana obyvatel před tabákovým kouřem stále nedostatečná. Česká legislativa se například doposud zcela nevypořádala se zákazem kouření v restauracích a barech, čímž zaostává za mnoha vyspělými státy. Mezi další problémy patří i nedostatečné omezení reklamy či nezvýšení daní na tabákové výrobky.

#### 2.2 Závislost na kouření

Závislost na užívání tabáku se skládá ze dvou hlavních složek: psychosociální a behaviorální závislosti a fyzické/drogové závislosti na nikotinu.

Psychosociální a behaviorální závislost je založená na tzv. kuřáckých stereotypech. Tato závislost vychází z určitých opakujících se činností spojených s kouřením. Může se jednat o kuřáckou pauzu v práci, cigaretu po jídle nebo relaxaci. Tyto činnosti spojené s užíváním tabáku se stávají pro kuřáka zvykem.

Fyzická závislost je způsobena návykovou složkou cigarety - nikotinem. Drogová závislost na nikotinu vzniká stimulací nikotinových acetylcholinových receptorů v mozku [7]. V reakci na vysazení nikotinu se pak dostavuje řada abstinčních příznaků, mezi které patří například touha po cigaretě, podrážděnost, strach, kašel a další [8].

Obecně lze říci, že na nikotinu je závislý ten kuřák, který kouří nejméně 10–15 cigaret denně a první cigaretu si potřebuje zapálit do hodiny po probuzení. Pro zjištění míry závislosti na cigaretách se používá Fagerströmův test, který závislost hodnotí v rozmezí 0-10 bodů (viz 2.1).

U každé otázky prosím vyberte jednu z možností	
1. Jak brzy po probuzení si obvykle zapálíte svou první cigaretu?	do 5 min (3 body)
	do půl hod (2 body)
	do 1 hod (1 bod)
	za více než hod (0 bodů)
2. Je pro vás obtížné nekouřit tam, kde je kouření zakázáno, např. v kostele, knihovně nebo kině?	ano (1 bod)
	ne (0 bodů)
3. Kterou cigaretu byste nejvíce postrádal/a?	první po probuzení (1 bod)
	kteroukoliv jinou (0 bodů)
4. Kolik cigaret denně průměrně kouříte?	do 10 cigaret (0 bodů)
	11–20 cigaret (1 bod)
	21–30 cigaret (2 body)
	více než 30 cigaret (3 body)
5. Kouříte častěji v prvních hodinách po probuzení než ve zbytku dne?	ano (1 bod)
	ne (0 bodů)
6. Kouříte i během nemoci, když musíte ležet většinu dne v posteli?	ano (1 bod)
	ne (0 bodů)

**Obrázek 2.1.** Fagerströmův test závislosti na cigaretách (již jeden bod znamená určitou závislost), převzato z [7].

## 2.3 Léčba

Léčba závislosti na tabákových produktech zahrnuje psychosociální a behaviorální intervenci a farmakologickou léčbu. Obě části léčby jsou popsány v následujících podkapitolách 2.3.1 a 2.3.2. Poslední podkapitola se zabývá využitím technologií v léčbě 2.3.3.

### 2.3.1 Farmakologická léčba

V případě farmakologické léčby se jedná o užívání vareniklinu, bupropionu a náhradní terapie nikotinem.

Vareniklin (v ČR známý pod názvem Champix) je prvním lékem pro odvykání kouření, který neobsahuje nikotin. Vareniklin se řadí mezi parciální agonisty nikotin-acetylcholinových receptorů. Jeho vlivem dochází k blokaci příjemných pocitů po vykouření cigarety čímž snižuje abstinenci příznaky nedostatku nikotinu v těle. Důsledkem je pak to, že cigareta kuřákovi méně chutná a riziko relapsu se snižuje.

Bupropion je antidepresivum, které vedle léčby depresivních poruch lze využít i pro odvykání kouření. Bupropion díky zvýšení hladiny dopaminu a noradrenalinu potlačuje touhu po cigaretě a abstinenci příznaky po odnětí nikotinu.

Náhradní terapie nikotinem je založena na potlačení abstinenci příznaků alternativním přísunem nikotinu. Nikotin se do těla dostává pomocí žvýkaček, náplastí nebo nikotinových inhalátorů, které jsou volně dostupné v lékárnách (bez receptu). Kuřák si dává množství přípravku, které je menší než jeho denní potřeba nikotinu, a tuto dávku postupně snižuje.

### ■ 2.3.2 Psychosociální a behaviorální intervence

Psychosociální a behaviorální intervence spočívají v krátkých diskuzích mezi pacientem a lékařem. Studie ukazují, že motivační diskuze s pacientem může značně zvýšit úspěšnost v odvykacím procesu i bez další farmakologické léčby [9]. Prvním krokem těchto intervencí je motivovat kuřáka v touze přestat kouřit. Mezi důvody, proč kuřák nechce přestat, může patřit např. nedostatek informací o škodlivých účincích užívání tabáku a výhod odvykání, obavy z abstinčních příznaků nebo demotivace z předchozích neúspěšných pokusů. V případě, že kuřák přestat chce, se lékař snaží pacientovi pomoci rozpoznat nejčastější kuřácké situace a vytvořit plán, jak je prožít bez cigarety.

### ■ 2.3.3 Využití technologií v léčbě (eHealth)

Díky rostoucí dostupnosti mobilních zařízení a internetového připojení vznikají nové možnosti v oblasti terapie závislosti na kouření. Pomocí mobilních či webových aplikací lze poskytnout virtuální intervence v podobě textových prezentací, které dokáží nahradit diskuze s reálným terapeutem.

Nedávná studie ukazuje, že až tři čtvrtiny kuřáků, kteří více než měsíc odvykali pomocí mobilní aplikace, preferují tuto metodu před tradičními intervencemi s terapeutem [3]. Mezi výhody tohoto přístupu, které účastníci studie zmínili, patří dlouhodobá každodenní podpora, flexibilita v plánování intervencí, rychlá dostupnost a autentická simulace dialogu.

Tato práce se bude věnovat rozšíření mobilní aplikace poskytující zmíněné intervence s virtuálním terapeutem. Aplikace dále umožňuje zaznamenávat cíle a úspěchy uživatele a v reálném čase poskytovat podporu. Tato aplikace bude rozšířena o modul Miniher popsaný v kapitole 4 a propojena s aplikací Buddy App (kapitola 5).

# Kapitola 3

## Vývoj multiplatformních mobilních aplikací

### 3.1 Motivace pro multiplatformní vývoj

V dnešní době je vývoj mobilních aplikací rozdělen na dvě základní platformy - Android a iOS [10]. Z tohoto důvodu v poslední době roste poptávka po vývoji multiplatformních mobilních aplikací. Klasický vývoj mobilních aplikací pro obě platformy vyžaduje použití jejich oficiálních frameworků a implementaci stejných funkčních požadavků ve dvou odlišných programovacích jazycích.

Vývoj aplikací pro obě platformy tak sebou přináší řadu problémů.

Prvním zřejmým problémem je požadovaná znalost obou frameworků. Protože je běžné, že se programátor mobilních aplikací specializuje pouze na jeden z frameworků, vývojářské týmy tak musí mít Android i iOS programátory, kteří odděleně udržují vývoj na obou platformách. To je z finančního i časového hlediska neefektivní, jedná-li se o vývoj aplikací s prakticky identickou funkcionalitou na obou platformách.

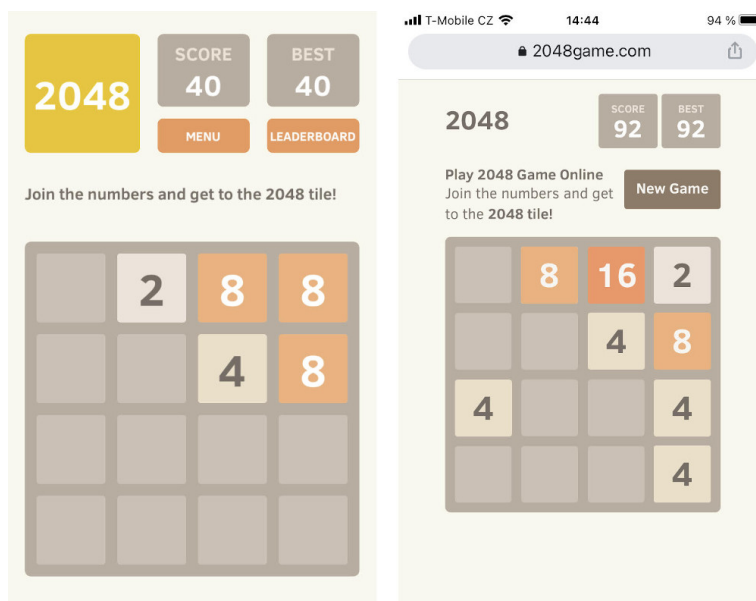
Dalším problémem je duplicita aplikační logiky. Stejné funkční požadavky je třeba souběžně přepisovat do dvou různých programovacích jazyků (Java/Kotlin pro Android a Swift/Objective-C pro iOS). Změny v aplikaci a její údržba tak přináší vždy dvojnásobnou práci pro vývojářský tým.

Z těchto důvodů je zřejmé, že v některých oblastech vývoje mobilních aplikací vzrůstá zájem po multiplatformních řešeních.

### 3.2 Současná multiplatformní řešení

Vývoj multiplatformních aplikací se dá rozdělit na tři základní kategorie: Progressive Web Apps, aplikace využívající WebView a hybridní aplikace poskytující nativní UI.

Progressive Web Apps (dále jen PWA) jsou webové aplikace běžící v mobilním prohlížeči, které vypadají a chovají se stejně jako mobilní aplikace dostupné z App Store nebo Google Play (viz obrázek 3.1). Protože se jedná o čistě webové stránky, vývoj mobilní i webové aplikace tak může probíhat současně ze stejného zdrojového kódu s využitím responzivního designu [11]. Tím se ušetří čas i finance na vývoj. Toto řešení má však několik nevýhod. Progressive Web App je zcela závislá na mobilním prohlížeči a není dostupná ze standardních Android a iOS obchodů aplikací. Ačkoli může v mnoha případech nahradit nativní mobilní aplikaci, hlavní nevýhodou je omezení přístupu k některým funkcím mobilních zařízení jako je GPS, Bluetooth, kamera a další.



**Obrázek 3.1.** Ukázka UI nativní aplikace oblíbené hry 2048 a její PWA verze.

Druhou kategorii tvoří aplikace vyvíjené pomocí frameworků, které využívají WebView dané platformy. Jedná se například o framework Apache Cordova<sup>1</sup>, který umožňuje zapouzdřit webovou aplikaci v HTML/Javascript do nativního kontejneru. Výhodou těchto aplikací využívajících WebView oproti PWA je, že framework může navíc poskytovat přístup k některým systémovým funkcím pomocí Javascriptového API. Aplikační logika a většina zdrojového kódu tak může být sdílena mezi mobilní i webovou aplikací. Nicméně ani toto řešení neumožňuje vždy plnohodnotně nahradit možnosti aplikací nativních. Hlavní nevýhodou je nízká výkonnost WebView enginů u starších verzí operačních systémů. Toto řešení tak nemusí poskytovat stejné UX/UI možnosti jako nativní frameworky.

Třetí kategorií multiplatformních (hybridních) aplikací jsou aplikace vyvíjené ve frameworku, který kompiluje UI popsany v jednom zdrojovém kódu do čistě nativních UI komponent na obou platformách. Mezi tyto frameworky v současnosti patří React Native<sup>2</sup>, Flutter<sup>3</sup> nebo Ionic<sup>4</sup>. Velikou výhodou těchto frameworků je, že výsledné aplikace téměř nelze odlišit od aplikací vyvíjených tradičním způsobem. Frameworky navíc poskytují širokou škálu API pro přístup k různým systémovým funkcím mobilního zařízení. Nicméně závislost na těchto API je i určitou nevýhodou, zvláště v situaci, kdy vychází nová verze SDK pro nativní platformu a je třeba vyčkat na vydání aktualizované verze daného API. Nicméně ze všech tří zmíněných kategorií multiplatformních řešení se tato kategorie nejvíce blíží aplikacím nativním, a z toho důvodu poskytuje lepší UX/UI a funkční možnosti než kategorie předešlé.

V současné době se multiplatformní frameworky generující nativní UI komponenty stávají čím dál více populárními. React Native patří mezi nejpoužívanější frameworky v této kategorii [12], a právě proto byl zvolen pro vývoj aplikace.

<sup>1</sup> <https://cordova.apache.org/>

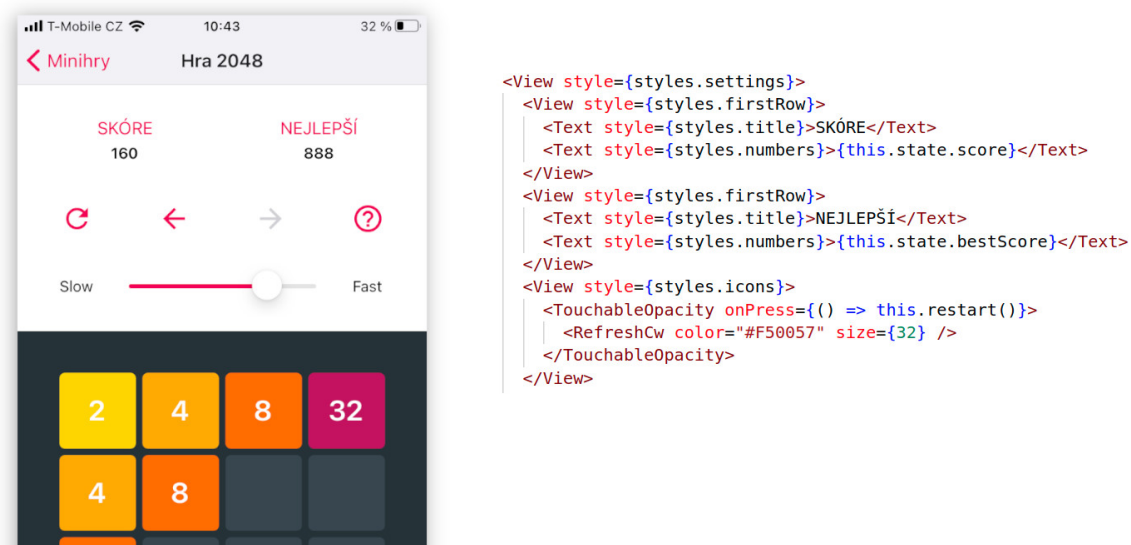
<sup>2</sup> <https://reactnative.dev/>

<sup>3</sup> <https://flutter.dev/>

<sup>4</sup> <https://ionicframework.com/>

### 3.3 React Native

React Native je JavaScriptový open-source framework pro tvorbu mobilních aplikací vyvíjený společností Facebook. React Native využívá knihovny React<sup>1</sup> od stejné společnosti pro tvorbu uživatelských rozhraní. Klíčovým konceptem této knihovny jsou React komponenty. Komponenta je určitá část uživatelského rozhraní reprezentovaná funkcí jazyka JavaScript, která vrací kompozici dalších komponent. Uživatelské rozhraní je tak definováno deklarativně podobně jako u statických webových stránek. K tomu se využívá jazyka JSX, který rozšiřuje syntaxi jazyka JavaScript o struktury připomínající XML. Na obrázku 3.2 je ukázka z uživatelského rozhraní hry 2048 a její definice v jazyce JSX.



**Obrázek 3.2.** Ukázka deklarace UI v jazyce JSX.

Jednotlivé komponenty vykreslují UI na základě předaných parametrů a vnitřního stavu. React nese své jméno kvůli způsobu, jakým vykresluje jednotlivé komponenty. Namísto překreslování celého abstraktního stromu komponent, React překresluje pouze ty komponenty, kterým se změnil stav (tedy *reaguje* na změnu stavu). Díky tomu React poskytuje dynamické UI, které rychle reaguje na interakci uživatele. Zároveň je zdrojový kód pro vývojáře snadno čitelný a udržitelný.

Zatímco byl React původně vyvíjen za účelem tvorby webových uživatelských rozhraní, React Native tuto knihovnu využívá k vykreslování nativních UI komponent pro iOS i Android. React Native tak nabízí vlastní komponenty jako jsou View, Text nebo Button, které na obou platformách mohou vypadat odlišně.

Kromě UI komponent React Native nabízí řadu užitečných API pro obě platformy jako je např. Appearance<sup>2</sup> pro zjištění tmavého/světlého režimu zařízení nebo Dimensions<sup>3</sup> pro zjištění rozměrů obrazovky. Kromě oficiálně poskytovaných API stále přibývá další řada open-source knihoven pro React Native rozšiřujících možnosti vývojářů, kteří investují do této platformy.

<sup>1</sup> <https://reactjs.org/>

<sup>2</sup> <https://reactnative.dev/docs/appearance>

<sup>3</sup> <https://reactnative.dev/docs/dimensions>



## 3.4 Vývojové prostředí Expo

Expo<sup>1</sup> je sada nástrojů usnadňujících vývoj React Native aplikací. Tradiční vývoj čistě React Native aplikací vyžaduje využití a znalost vývojových prostředí Android Studio a Xcode. Expo však poskytuje vlastní SDK, které obě prostředí obchází. Pomocí Expo CLI lze snadno vytvořit strukturu nového projektu, která naprosto odstiňuje vývojáře od nativních modulů. Vývojář tak nemusí mít znalost jazyků Java/Kotlin nebo Objective-C/Swift a k vývoji nepotřebuje prostředí macOS.

Expo navíc poskytuje vlastní mobilní aplikaci Expo Go<sup>2</sup> dostupnou na obou platformách, pomocí které lze vyvíjenou aplikaci snadno otestovat na reálných zařízeních. Aplikaci lze také jednoduše sestavit do binárních souborů *.apk* pro Android a *.ipa* pro iOS a publikovat na obchodech aplikací.

I když se vývoj v Expo prostředí zdá být bezproblémový, nemusí být vždy tou nejlepší volbou. V některých případech je potřeba do React Native aplikace přidat moduly v nativních jazycích, což Expo projekt neumožňuje. Expo CLI ale kdykoliv umožňuje Expo projekt převést na čistě React Native projekt. Dalším problémem může být nadměrná velikost sestavené aplikace (velikost jednoduché “Hello World” aplikace je mezi 15 až 20mb). Důvodem toho je, že sestavená aplikace obsahuje všechna API, která Expo poskytuje, tedy i ta, které samotná aplikace nevyužívá. Expo ale slibuje, že tento problém bude v budoucnu odstraněn [13].

---

<sup>1</sup> <https://expo.io/>

<sup>2</sup> <https://expo.io/client>

# Kapitola 4

## Minihry v React Native

Po seznámení se se základními principy frameworku React Native bylo implementováno rozšíření aplikace pro odvykání kouření v podobě šesti miniher. Tyto minihry mají poskytovat okamžité odvedení pozornosti od chuti na cigaretu v rámci kritické fáze odvykání. Návrh tohoto rozšíření vychází již z realizované aplikace pro Android vyvíjené v rámci jiné bakalářské práce [14]. Cílem této části práce bylo převedení stávající Android aplikace na aplikaci multiplatformní pomocí frameworku React Native. Z tohoto důvodu se budou následující podkapitoly věnovat především implementaci jednotlivých her, protože jejich návrh vychází ze zmíněné bakalářské práce. V poslední podkapitole 4.7 jsou shrnuty výhody a nevýhody implementace této aplikace v React Native v porovnání s původní nativní aplikací.

### 4.1 Navigace v aplikaci

Uživatelské rozhraní aplikace se sestává z úvodní obrazovky, ze které se spouští jednotlivé hry. Pro navigaci v aplikaci byla zvolena knihovna React Navigation<sup>1</sup>, která umožňuje strukturovat aplikaci do jednotlivých obrazovek pomocí různých druhů navigace.

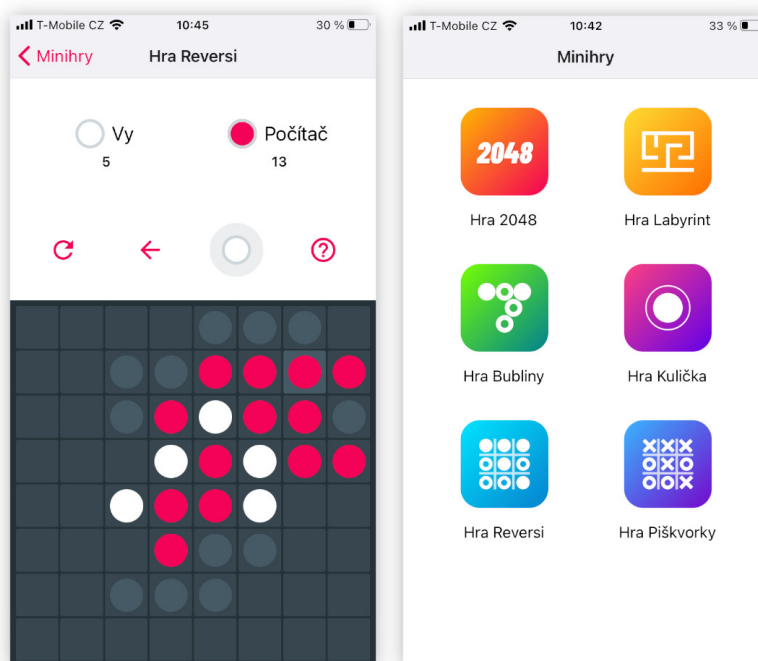
Pro tuto aplikaci byla zvolena *zásobníková navigace (Stack Navigation)*, která ukládá každou novou obrazovku na vrchol zásobníku reprezentující navigační strukturu. Ve výchozím nastavení Stack Navigation vykresluje v horní části obrazovky navigační panel s názvem obrazovky a možností vrátit se zpět k předešlé obrazovce v zásobníku, viz 4.2. Vzhled navigačního panelu se pak přizpůsobuje operačnímu systému.

Vzhled a další nastavení navigace se definuje pomocí komponent *NavigationContainer* a *Stack.Navigator*. Jednotlivé obrazovky jsou definovány komponentami *Stack.Screen*. Každá komponenta *Stack.Screen* má dva povinné atributy: název obrazovky, kterým se spouští, a odkaz na React komponentu, která ji reprezentuje, viz 4.1.

```
function App() {
  return (
    <NavigationContainer theme={MyTheme}>
      <Stack.Navigator initialRouteName="HomeScreen">
        <Stack.Screen name="Minihry" component={HomeScreen} />
        <Stack.Screen name="Hra 2048" component={Game2048} />
        <Stack.Screen name="Hra Kulička" component={GameBall} />
        <Stack.Screen name="Kulička Level" component={BallLevel} initialParams={{ level: 1 }}/>
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

**Obrázek 4.1.** Ukázka navigace definované pomocí knihovny React Navigation.

<sup>1</sup> <https://reactnavigation.org/>



**Obrázek 4.2.** Hlavní menu aplikace a Hra Reversi.

Samotná navigace pak probíhá pomocí funkce *navigate*<sup>1</sup>, která se automaticky přechází do všech obrazovek v parametru komponent *navigation*. Funkce *navigate* navíc jako druhý parametr přijímá objekt, ve kterém lze definovat další informace, které chceme přenést do nové obrazovky. Tyto informace jsou pak v nově otevřené obrazovce dostupné z parametru *route*.

Kromě funkce *navigate* Stack Navigation poskytuje také funkci *goBack*, která zobrazí předchozí obrazovku nebo funkci *replace*, která nahradí aktuální obrazovku jinou a tak ji vymaže ze zásobníku. S těmito nástroji se stává implementace navigace v aplikaci velice snadnou.

## 4.2 Hra 2048

Hra 2048 je populární hra, ve které se přesouvají a skládají očíslované kameny na hrací desce o 16 polích. Hra má jednoduchá pravidla. Hráč v každém tahu zvolí jeden ze čtyř směrů, ve kterém se budou kameny přesouvat. Po zvolení směru se kameny přesunou tak, že pokud dojde ke srážce dvou stejně očíslovaných kamenů, sloučí se v jeden s dvojnásobnou hodnotou. Pokud kámen narazí do jiného kamene s odlišnou hodnotou, zůstávají vedle sebe. Navíc po každém tahu na náhodném místě hrací desky přibudou další dva kameny s hodnotou 2 nebo 4. Cílem hry je volit takové tahy, aby nedošlo k úplnému zaplnění hrací desky a postupným skládáním kamenů složit kámen s hodnotou 2048 (odtud jméno hry).

### 4.2.1 Funkční požadavky a uživatelské rozhraní

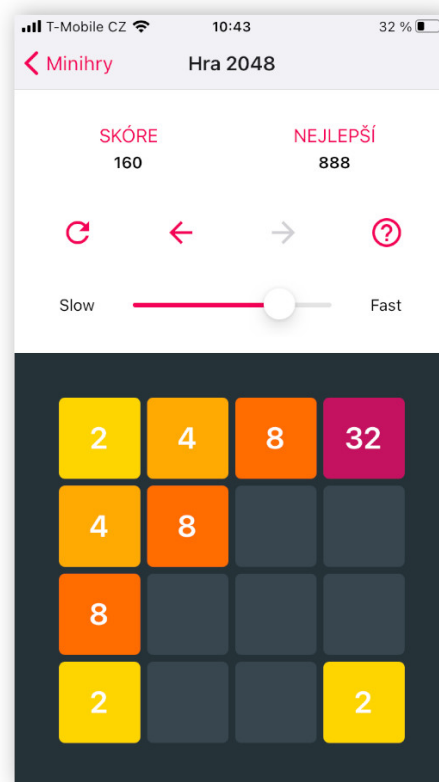
Mezi funkční požadavky této hry patří:

- Zobrazení aktuálního skóre

<sup>1</sup> <https://reactnavigation.org/docs/navigation-actions/>

- Ukládání nejlepšího skóre
- Resetování hry
- Možnost vrátit se zpět nebo jít vpřed o jeden tah
- Nápověda
- Animace přesouvání kamenů
- Nastavení rychlosti animace

Rozložení interakčních prvků uživatelského rozhraní vychází z původní Androidové aplikace. Grafický návrh byl převeden do více minimalistického stylu za využití knihovny Expo Vector Icons<sup>1</sup> pro vykreslování jednotlivých ikon, viz 4.3.



**Obrázek 4.3.** Uživatelské rozhraní hry 2048.

#### ■ 4.2.2 Stav hry

Hrací deska o 16 polích je reprezentována 2D polem s čísly, které odpovídají hodnotám umístěných kamenů. Prázdná pole jsou reprezentována nulou. Toto pole je pak v *render* metodě přímo mapováno na jednotlivé View komponenty představující kameny.

Stav React komponenty se ukládá do atributu *state* v podobě JavaScript objektu. Pro Hru 2048 byly zvoleny následující atributy tohoto objektu:

- **firstState** - 2D pole představující aktuální stav hry.
- **secondState** - Pomocný atribut, ukládající vypočtený nový stav hry. Tato proměnná slouží k vytvoření animace přesouvajících se kamenů z *firstState* do *secondState*.
- **prevState** - 2D pole udržující předchozí stav hry. Pokud chce hráč vrátit poslední tah, zkopíruje se její hodnota do atributu *firstState*.

<sup>1</sup> <https://docs.expo.io/guides/icons/>

- **nextState** - Atribut, do kterého se zkopíruje hodnota *firstState* v případě, že se hráč vrátil o tah zpět. Tento atribut se použije, když hráč chce jít znovu vpřed.
- **animations** - Seznam kamenů, které se mají přesunout. Každý prvek tohoto seznamu má počáteční a cílovou souřadnici a odkaz na animovanou pozici na vertikální a horizontální ose.
- **animationRunning** - Booleanovská hodnota nastavená na *false*, pokud právě probíhá animace kamenů. Její použití je popsáno v sekci 4.2.3.
- **animationSpeed** - Rychlost animace kamenů zvolená hráčem.
- **score** - Aktuální skóre.
- **bestScore** - Nejlepší skóre načtené z úložiště zařízení.
- **gameFinished** - Booleanovská hodnota ukazující, zda hra stále probíhá nebo již došlo k sestavení kamene s hodnotou 2048.

### 4.2.3 Animace

Animace kamenů je implementovaná pomocí React Native knihovny *Animated*<sup>1</sup>. Tato knihovna umožňuje animovat změnu číselné hodnoty v daném časovém intervalu. Poté, co hráč gestem ukáže zvolený směr, aplikace vyhodnotí, jaké kameny se mají přesunout. Tyto kameny jsou následně reprezentovány objekty uloženými do pole *animations* ve stavu komponenty. Jednotlivé objekty obsahují počáteční a koncové souřadnice kamene na herní desce společně se dvěma animovanými hodnotami (*Animated.Value()*) představujícími aktuální pozici při animaci ve vertikálním a horizontálním směru.

Po naplnění pole *animations* objekty reprezentující animované kameny je pro každý z nich spuštěna funkce *Animated.timing*. Tato funkce pro daný objekt spustí animaci její aktuální hodnoty do koncové hodnoty vypočtené z počátečních a koncových souřadnic objektu, viz 4.4. Rychlost animace je pak určena atributem *animationSpeed* ve stavu komponenty (viz sekce 4.2.2).

```
let animations = this.state.animations;
for (let i = 0; i < animations.length; i++) {
  let direction = animations[i][0][0] == animations[i][1][0] ? 1 : 0;
  Animated.timing(animations[i][3 - direction], {
    toValue:
      (animations[i][1][direction] - animations[i][0][direction]) * 76,
    duration: this.state.animationSpeed,
    useNativeDriver: true,
  }).start(({ finished }) => {
    //pokracovani po skonceni animace...
  });
}
```

**Obrázek 4.4.** Ukázka využití knihovny *Animated* v kódu.

## 4.3 Hra Reversi

Hra Reversi, také nazývaná Othello, je desková hra pro dva hráče hraná na desce 8x8 polí. Hráči na hrací desku postupně pokládají kameny, které jsou z jedné strany bílé a z druhé černé. Hráč, který je na tahu, se snaží položit kámen tak, aby mezi právě položeným kamenem a jiným kamenem své barvy uzavřel souvislou řadu (ve vertikálním, horizontálním nebo diagonálním směru) soupeřových kamenů. Uzavřené kameny soupeře se pak obrátí na barvu hráče, který provedl tah.

<sup>1</sup> <https://reactnative.dev/docs/animated>

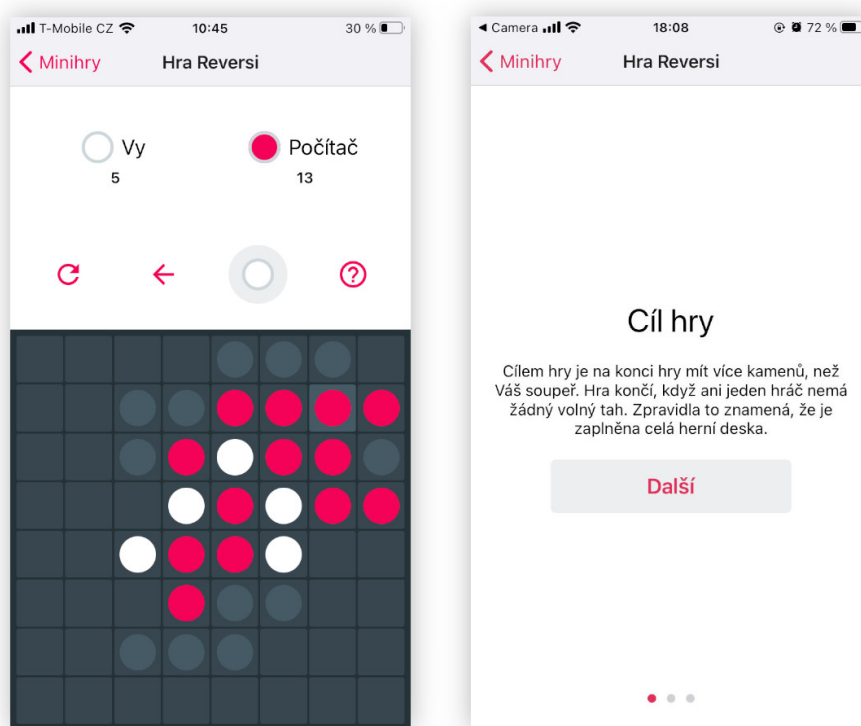
Hra končí v případě, že hrací deska je již zcela zaplněna kameny nebo v případě, že jednomu z hráčů se podaří otočit všechny kameny na svou barvu. Vyhrává ten hráč, který má na konci hry více otočených kamenů.

### 4.3.1 Funkční požadavky a uživatelské rozhraní

Protože se jedná o hru pro dva hráče, oproti funkčním požadavkům předchozí hry zde přibývá i požadavek na umělého protihráče. Funkční požadavky jsou:

- Zobrazení aktuálního skóre obou hráčů
- Resetování hry
- Možnost vrátit poslední tah
- Náповěda
- Animace otáčení kamenů
- Změna barvy, za kterou uživatel hraje

Na obrázku 4.5 je náhled na hlavní obrazovku hry a obrazovku nápovědy. Ovládací prvky jsou opět v horní části obrazovky jak tomu bylo u hry 2048.



Obrázek 4.5. Uživatelské rozhraní hry Reversi.

### 4.3.2 Stav hry

Podobně jako u hry 2048 je stav hrací desky reprezentovaná 2D polem s čísly 0 až 3, které po pořadí odpovídají prázdnému poli, poli s bílým kamenem, poli s červeným kamenem a poli, kam lze umístit další kámen (viz obrázek 4.5).

Mezi hlavní atributy stavu React komponenty představující tuto hru patří:

- `currentState` - 2D pole s aktuálním stavem hrací desky.

- **previousState** - Pomocný atribut, do kterého se před dalšími dvěma tahy hráčů kopíruje *currentState*. Tento atribut se využije v případě, že se hráč rozhodne vrátit svůj poslední tah.
- **player** - Booleanovská hodnota reprezentující zvolenou barvu hráče.
- **turn** - Booleanovská hodnota reprezentující, jaký hráč je na tahu.
- **score** - Pole o dvo prvcích, reprezentující aktuální skóre obou hráčů.
- **lastMove** - Pole o dvou prvcích reprezentující souřadnici posleního tahu hráče na hrací desce. Tento tah je pak pro přehlednost na hrací desce zvýrazněn.
- **turnedStones** - Pole souřadnic právě uzavřených kamenů, které se přetočí na barvu hráče, který provedl tah.
- **animations** - 2D pole animovaných hodnot (*Animated.Value*, viz 4.2.3) pro každé pole hrací desky. Tyto animované hodnoty se používají při animaci otáčení kamenů.

### 4.3.3 Herní logika

Implementace herní logiky byla dekomponována na tyto hlavní funkce:

- **getTurned(x, y)** - Funkce, která za argumenty bere tah hráče reprezentovaný souřadnicí nově položeného kamene na hrací desce. Tato funkce vrací pole souřadnic nově uzavřených kamenů protihráče.
- **showOptions()** - Funkce, která do hrací desky vykreslí všechny možné tahy pro hráče, který je na tahu. Tato funkce nic nevrací, ale přímo upravuje atribut *currentState* ve stavu komponenty.
- **move(x, y)** - Tato funkce provede nový tah na pozici uvedené v argumentu. Výsledkem funkce je změna stavu komponenty a spuštění animace otočení uzavřených kamenů. Jednotlivé kroky této funkce jsou:
  1. Vytvoření lokální kopie aktuálního stavu hry
  2. Výpočet otočených kamenů protihráče pomocí funkce *getTurned*
  3. Uložení stavu hrací desky do atributu *prevState* pro vrácení tahu
  4. Vymazání polí napovídajících možné tahy
  5. Výpočet aktuálního skóre z otočených kamenů
  6. Uložení aktuálního skóre, otočených kamenů a posledního tahu do stavu komponenty a spuštění animace
- **compMove()** - tato funkce představuje tah umělého protihráče. Jeho naivní strategie spočívá ve snaze otočit v každém tahu co nejvíce kamenů. Nejprve tedy každý možný tah ohodnotí počtem otočených kamenů pomocí funkce *getTurned*. Z takto ohodnocených tahů vybere ten s maximální hodnotou a zavolá na něj funkci *move*.

Výsledný průběh hry lze poslat následovně:

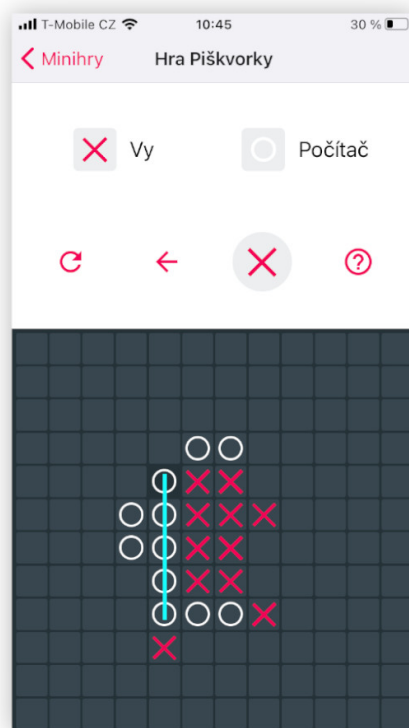
1. IF je na tahu umělý hráč
2. THEN zavolá se funkce *compMove*, která následně zavolá funkci *move*
3. ELSE hráč zvolením tahu zavolá funkci *move*
4. funkce *move* vypočte nový stav hry a spustí animaci otáčení kamenů
5. po skončení animace se změní hráč, který je na tahu
6. IF je konec hry
7. THEN vyhodnotí se, kdo vyhrál a zobrazí se výsledné skóre
8. ELSE vykreslí se nápověda možných tahů pomocí funkce *showOptions*
9. GO TO krok 1.

## 4.4 Hra Piškvorky

Hra piškvorky je desková hra, ve které se hráči střídají v kreslení křížků a koleček do čtvercové mřížky na hrací desce. Cílem hry je vytvořit nepřerušenu řadu pěti svých symbolů.

### 4.4.1 Funkční požadavky a uživatelské rozhraní

Funkční požadavky byly téměř stejné jako pro hru Reversi 4.3.1. I v této hře bylo zapotřebí vytvořit umělého protihráče. Rozložení ovládacích prvků uživatelského rozhraní je totožné se hrou Reversi. Hrací deska byla rozdělena pouze na 12x12 polí, aby uživatel neměl problém se správným zvolením svého tahu. Vzhled hry je vidět na obrázku 4.6.



Obrázek 4.6. Uživatelské rozhraní hry Piškvorky.

### 4.4.2 Stav hry a herní logika

Atributy stavu komponenty jsou téměř totožné se hrou Reversi (viz 4.3.3). Celkově nebylo potřeba ukládat tolik informací jako u hry Reversi, protože tato hra nevyžadovala žádné animace. Aplikační logika byla rozdělena do podobných funkcí jako v předchozí hře. V této hře bylo ale vyhodnocování konce hry značně složitější. K tomu bylo potřeba procházet všechny možné pětičky pozic na hrací desce ve všech čtyřech směrech.

Dále bylo potřeba vytvořit umělého hráče. Byla zvolena vcelku jednoduchá, ale pro uživatele stále zajímavá strategie popsaná takto:



```

IF mám již 4 symboly v řadě a mám místo na pátý
  THEN napíši pátý symbol do řady a vyhrávám
ELSE IF protihráč má 4 symboly v řadě a místo na pátý
  THEN zaplním jedno z míst v řadě protihráče
ELSE IF mám již 3 symboly v řadě a mám místo na zbývající
  THEN napíši čtvrtý symbol do řady
ELSE IF protihráč má 3 symboly v řadě a místo na zbývající
  THEN zaplním jedno z míst v řadě protihráče
ELSE IF mám již 2 symboly v řadě a mám místo na zbývající
  THEN napíši třetí symbol do řady
ELSE napíši symbol na náhodné pole, které sousedí s mým
  symbolem

```

## 4.5 Hry Kulička a Labyrint

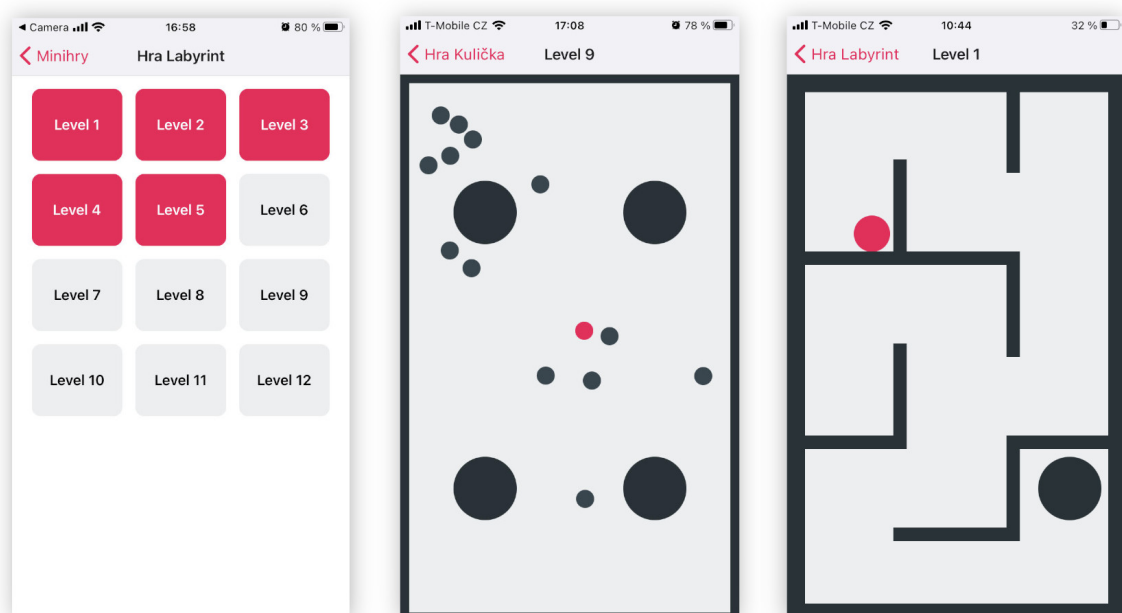
Hry Kulička a Labyrint jsou založeny na simulaci gravitace nakláněním telefonu. Ve hře Kulička se hráč snaží nakláněním telefonu navigovat všechny kuličky do děr na hrací ploše přičemž červená kulička musí spadnout do díry jako poslední. Ve hře Labyrint se hráč snaží v co nejkratším čase provést kuličku labyrintem do cílové díry.

### 4.5.1 Funkční požadavky a uživatelské rozhraní

Mezi funkční požadavky patří:

- simulace gravitace pomocí naklánění telefonu
- simulace srážek kuličky s překážkou
- měření času
- 12 úrovní obtížnosti pro obě hry
- ukládání splněných úrovní

Uživatelské rozhraní je rozděleno na dvě obrazovky, viz 4.7. První obrazovka slouží k výběru herní úrovně. Druhá pak vykresluje zvolenou hru.



**Obrázek 4.7.** Uživatelské rozhraní her Kulička a Labyrint.

## 4.5.2 Game Engine

K implementaci obou her bylo potřeba pracovat s akcelerometrem mobilního zařízení. K tomu byla využita knihovna Expo Sensors<sup>1</sup>, která poskytuje rozhraní k různým sensorům jako jsou magnetometr, barometr, gyroskop nebo akcelerometr. V případě akcelerometru rozhraní poskytuje přístup k aktuálnímu zrychlení vertikálního a horizontálního naklonění telefonu.

Dále byly použity knihovny React Native Game Engine<sup>2</sup> a Matter.js<sup>3</sup>, s jejíž pomocí bylo možné reprezentovat objekty ve hře (kulička, překážka, díra), měnit jejich rychlost a vyhodnocovat srážky dvou objektů.

Knihovna React Native Game Engine poskytuje komponentu *GameEngine*, která v 16ms intervalech vykresluje aktuální stav hry. Této komponentě jsou v attributech předány objekty hry a funkce, které jsou volány v každém cyklu vykreslování hry. Objekty hry jsou definovány svou velikostí (výškou a šířkou), pozicí na x-ové a y-ové ose a React Native komponentou, která je vykresluje.

Ve funkcích, které jsou volány v hrací smyčce, pak měníme pozici či rychlost hracích objektů pomocí knihovny Matter.js. Na obrázku 4.8 je ukázka změny rychlosti kuličky v závislosti na aktuálních hodnotách z akcelerometru (naklonění telefonu). Protože se tyto hodnoty liší v závislosti na operačním systému, bylo potřeba tuto skutečnost ošetřit.

```
let os = Device.osName == "iOS" ? 1 : -1;
Matter.Body.setVelocity(entities.Ball.body, {
  x: entities.Ball.body.velocity.x + os * acc.x,
  y: entities.Ball.body.velocity.y - os * acc.y,
});
```

**Obrázek 4.8.** Změna pozice kuličky v závislosti na naklonění zařízení.

Knihovna Matter.js také umožňuje reagovat na srážky dvou objektů. Na obrázku 4.9 je ukázka “srážky” kuličky s dírou ve hře Labyrint. V této ukázce se má kulička po spadnutí do díry zastavit pomocí funkce *setStatic*.

```
Matter.Events.on(engine, "collisionStart", (event) => {
  var pairs = event.pairs;

  var objA = pairs[0].bodyA.label;
  var objB = pairs[0].bodyB.label;

  if (objA === "Ball" && objB === "Goal") {
    Matter.Body.setStatic(entities.Ball.body, true);
    dispatch({ type: "gameOver" });
  }
});
```

**Obrázek 4.9.** Vyhodnocení srážky dvou objektů pomocí knihovny Matter.js.

Pomocí obou knihoven lze pak rozhodovat o chování herních prvků a vyhodnotit, kdy má hra skončit. Po skončení hry se stav komponenty *GameEngine* vyresetuje.

<sup>1</sup> <https://docs.expo.io/versions/latest/sdk/sensors/>

<sup>2</sup> <https://www.npmjs.com/package/react-native-game-engine>

<sup>3</sup> <https://brm.io/matter-js/>

## 4.6 Hra Bubliny

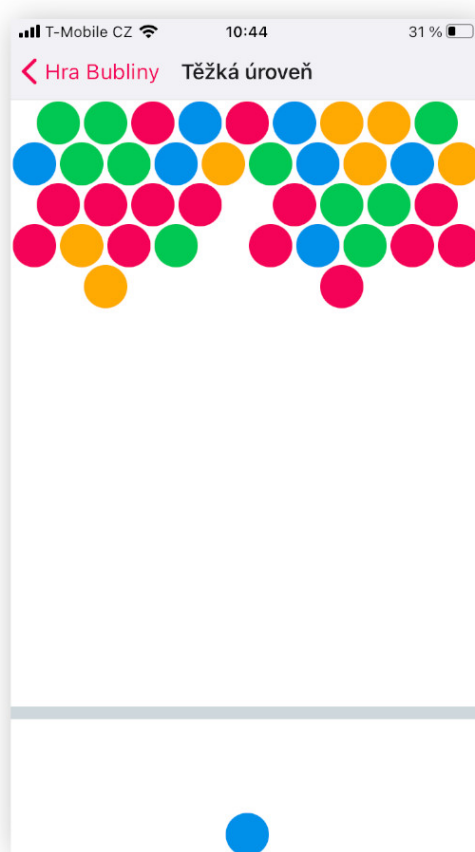
V této hře se hráč snaží střílet bubliny různé barvy do skupiny bublin se stejnou barvou. Pokud dojde ke srážce tří a více bublin stejné barvy, tyto bubliny popraskají. Pokud nedoje k prasknutí, vystřelená bublina zůstane na místě srážky a skupina bublin se tím zvětší. Cílem hry je co nejdříve popraskat všechny bubliny.

### 4.6.1 Funkční požadavky a uživatelské rozhraní

Funkční požadavky převážně vyplývají z pravidel hry:

- zvolení směru vystřelení bubliny
- odraz buliny od stěn
- praskání bublin stejné barvy
- tři úrovně obtížnosti

Na obrázku níže je vidět výsledná podoba uživatelského rozhraní. Šedá linie v dolní části obrazovky značí hranici, kterou rostoucí skupina bublin nesmí překročit.



**Obrázek 4.10.** Vyhodnocení srážky dvou objektů pomocí knihovny Matter.js.

### 4.6.2 Implementace herní logiky

Stejně jako u her Labyrint a Kulička 4.5, i zde byly použity knihovny React Native Game Engine a Matter.js pro animaci vystřelené bubliny a registraci srážek. Pomocí

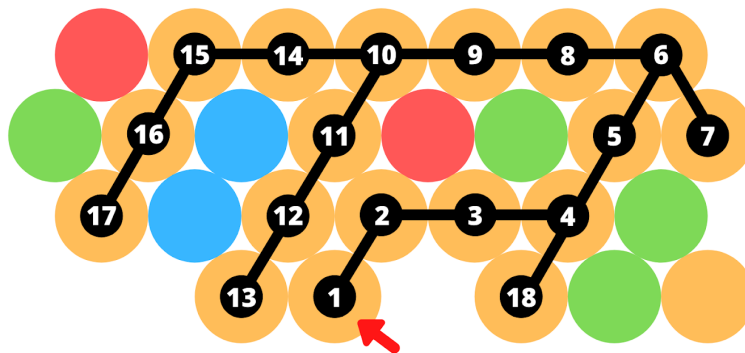
komponenty *TouchableWithoutFeedback*<sup>1</sup> bylo možné zjistit souřadnici dotyku hráče na obrazovce a pomocí té určit směr vystřelené bubliny. Při srážce bubliny se zdí se bublině mění vertikální rychlost na opačnou hodnotu. Složitější situace ale nastává při srážce vystřelené bubliny s ostatními.

Tím, jak se na sebe bubliny vrství, vzniká uspořádaná síť, ve které se každá bublina dotýká maximálně šesti sousedních bublin. Tato síť se dá reprezentovat neorientovaným grafem, ve kterém každá bublina představuje uzel a místo, kde se sousední bubliny dotýkají představuje hranu mezi nimi. K určení, jaké bubliny mají prasknout bylo potřeba vytvořit algoritmus, který by prohledával tento graf do hloubky počínaje uzlem, kde došlo ke střetu s vystřelenou bublinou. Algoritmus byl implementován rekurzivní funkcí popsanou níže.

Funkce bere za argumenty **uzel (n)** a **seznam již navštívených uzlů (array)**. Na začátku je tento seznam prázdný. Kroky funkce jsou popsány v následujícím pseudokódu:

1. přidej **n** do **array**
2. pro každou hranu  $e_i(n, m_i)$  vycházející z **n** do uzlu  $m_i$ :
3. IF  $m_i$  reprezentuje bublinu se stejnou barvou a není v **array**:
4. THEN rekurzivně volej funkci s parametry  $(m_i, \text{array})$ , její výsledek ulož do **array**
5. RETURN **array**

Průchod algoritmu grafem je znázorněn na obrázku 4.11. Funkce prohledává sousední uzly po směru hodinových ručiček počínaje pravou horní hranou. Na obrázku jsou vyznačeny černou barvou pouze hrany a uzly, po kterých algoritmus přechází. Uzly jsou očíslovány v pořadí, v jakém byly algoritmem navštíveny.



**Obrázek 4.11.** Prohledávání grafu do hloubky.

## 4.7 Srovnání s nativní aplikací

Během implementace této aplikace bylo zřejmé, jaké výhody a nevýhody toto multiplatformní řešení přináší.

Mezi některé výhody patří rychlost vývoje a testování. React Native umožňuje tzv. fast refresh<sup>2</sup>, což znamená, že změny ve zdrojovém kódu se okamžitě projevují

<sup>1</sup> <https://reactnative.dev/docs/touchablewithoutfeedback>

<sup>2</sup> <https://reactnative.dev/docs/fast-refresh>

ve vyvíjené aplikaci běžící na připojeném zařízení. Expo navíc umožňuje spouštět vyvíjenou aplikaci na Android i iOS zařízeních současně. Tím se značně usnadňuje rychlé testování aplikace na obou platformách.

Další zřejmou výhodou je použití stejného kódu pro vytvoření aplikací pro obě platformy. Až na pár výjimek se výsledná aplikace chová a vypadá stejně na Android i iOS, přičemž vykreslené uživatelské rozhraní využívá nativních grafických prvků.

Díky nástrojům, které poskytuje prostředí Expo, se vývoj obešel bez použití standardních vývojových prostředí jako je Android Studio či XCode. Vývoj se tak mohl plně soustředit na psaní kódu a nevěnovat tolik pozornosti různým konfiguracím pro obě platformy.

Obtíže nastávají v situaci, kdy aplikace vyžaduje některé náročnější funkce jako je rozpoznávání dotykových gest nebo práce s akcelerometrem. V této situaci je totiž vývojář zcela závislý na React Native knihovnách, které tyto funkce poskytují. V některých případech se ukázalo, že rozhraní poskytovaná těmito knihovnamy se chovají jinak na iOS než na Android zařízeních. Ve většině jednoduchých aplikací tyto problémy ale nenastávají.

# Kapitola 5

## Serverové řešení aplikace Buddy App

Po seznámení se s frameworkem React Native a implementaci miniher, bylo mým zadáním navrhnout a implementovat serverové řešení aplikace Buddy App, která je vyvíjená ve stejném frameworku. Aplikace Buddy App je určena pro přátele kuřáka, kteří se snaží podpořit jeho odvykací proces. Tato aplikace seznamuje uživatele s průběhem odvykání a možnými komplikacemi. Pomáhá také sledovat kuřákův postup a upozorňuje na jeho nové pokroky. Uživatel tak dokáže lépe pomoci kuřákovi v nejtěžších chvílích odvykání a může ho účinněji podporovat.

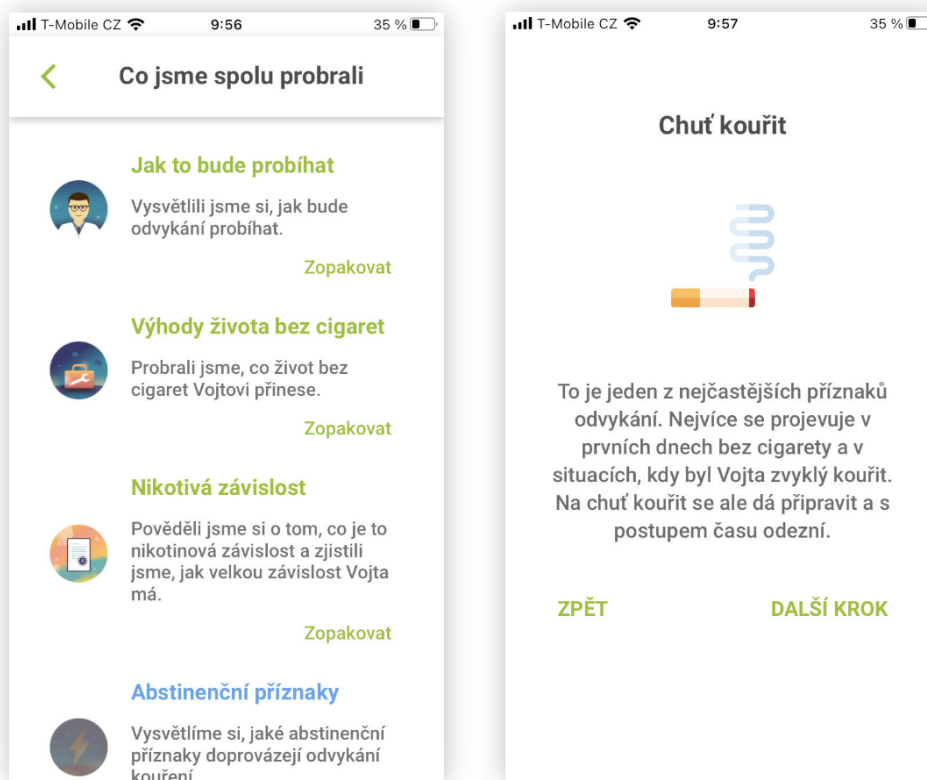
Cílem bylo napojit aplikaci Buddy App na server již existující aplikace pro odvykání kouření a umožnit tak uživateli sledovat a reagovat na pokrok v odvykacím procesu kuřáka. Současný stav Buddy App je popsán v podkapitole 5.1. Server odvykací aplikace je implementován pomocí webového frameworku Django, kterému se bude věnovat podkapitola 5.2. V následujících podkapitolách je popsán návrh a implementace jednotlivých rozhraní pro Buddy App. Mezi ně patří pozvání přítele kuřáka do aplikace Buddy App a registrace nového uživatele (podkapitola 5.3), zobrazení pokroku kuřáka v odvykání (podkapitola 5.4) a zaslání zpráv do Buddy App pomocí notifikací (podkapitola 5.5).

### 5.1 Současný stav aplikace Buddy App

Aplikace ve výchozím stavu poskytovala osm informačních sezení pro přítele podporujícího kuřáka v odvykání (viz obrázek 5.1). Tato sezení se zabývala následujícími tématy:

- **Průběh odvykání** - popis jednotlivých fází procesu odvykání
- **Výhody života bez cigaret** - popis pozitivních změn v životě kuřáka po zdoání závislosti
- **Nikotinová závislost** - vysvětlení vzniku a určení míry závislosti na nikotinu
- **Abstinenční příznaky** - popis různých druhů abstinenčních příznaků v průběhu odvykání
- **Podpora** - poskytnutí několika rad, jak reagovat na úspěchy i neúspěchy kuřáka v odvykání a jak ho skutečně psychicky podpořit
- **Zvládnutí chuti na cigaretu** - vysvětlení psychosociální a behaviorální závislosti na cigaretách, viz 2.2
- **Náhradní nikotinové přípravky** - popis různých druhů přípravků v rámci náhradní terapie nikotinem, viz 2.3.1
- **Zmírnění abstinenčních příznaků** - poskytnutí několika rad jak podpořit kuřáka při výskytu konkrétních abstinenčních příznaků

Jednotlivá sezení bylo potřeba synchronizovat s procesem odvykání sledovaného kuřáka. K tomu bylo potřeba navrhnout příslušná rozhraní pro komunikaci Buddy App s hlavní odvykací aplikací. Návrh a implementace těchto rozhraní jsou popsány v následujících podkapitolách.



Obrázek 5.1. Ukázka sezení v Buddy App

## 5.2 Django

Django<sup>1</sup> je jedním z populárních webových frameworků pro rychlý vývoj udržitelných a škálovatelných webových aplikací [15]. Jedná se o open-source projekt s podrobnou dokumentací a aktivní komunitou vývojářů. Framework Django je napsán v jazyce Python, který je k dispozici pro všechny hlavní operační systémy. Díky tomu aplikace není vázaná pouze na určitý typ serveru a je snadno přenositelná. Některé vlastnosti tohoto frameworku jsou popsány níže.

### ■ Rychlý vývoj

Django se vyznačuje tím, že poskytuje velké množství podrobně zdokumentovaných funkcí<sup>2</sup>, které vývoj webových aplikací běžně vyžaduje. Mezi tyto funkce patří např. zpracování příchozích formulářů, editování dat v modelu, perzistence dat, vytváření HTML šablon a další. Kromě dokumentace Django projekt poskytuje i vlastní tutoriály<sup>3</sup>, které značně usnadňují orientaci v tomto frameworku.

### ■ Univerzálnost

Django umožňuje vytvářet širokou škálu webových stránek od blogů či wiki stránek po sociální sítě nebo zpravodajské portály. Django lze napojit na jakýkoliv klientský framework. Umožňuje exportovat obsah v téměř jakémkoliv formátu (HTML, RSS, JSON, XML a další<sup>4</sup>).

<sup>1</sup> <https://www.djangoproject.com/>

<sup>2</sup> <https://docs.djangoproject.com/en/3.2/topics/>

<sup>3</sup> <https://docs.djangoproject.com/en/3.2/intro/>

<sup>4</sup> <https://docs.djangoproject.com/en/3.2/topics/templates/>

## ■ Bezpečnost

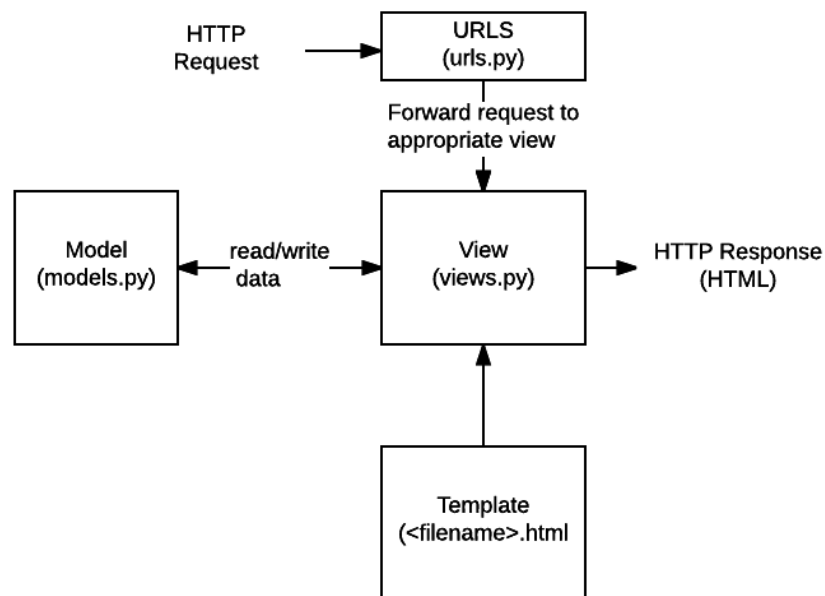
Django řeší velké množství bezpečnostních opatření za vývojáře a předchází tak častým implementačním chybám. Mezi tato opatření patří hashování hesel, správa sessions a další. Také poskytuje ochranu před bezpečnostními útoky typu SQL injection, cross-site scripting, cross-site request forgery a další [16].

## ■ Škálovatelnost

Django využívá architekturu “shared-nothing”, ve které je každá komponenta architektury nezávislá na ostatních a proto může být kdykoliv nahrazena jinou [17]. Tento přístup umožňuje škálovat aplikaci při vzrůstajícím provozu přidáním hardwaru na jakékoli úrovni (databázový server, aplikační server a jiné).

### 5.2.1 Struktura aplikace ve frameworku Django

Django dekomponuje aplikační logiku do několika zdrojových souborů, což umožňuje lepší čitelnost a údržbu. Běžný webový server zpracovává HTTP požadavky na základě URL adresy a informací obsažených v hlavičce a těle požadavku. Podle druhu požadavku pak čte nebo upravuje data v databázi nebo provádí jiné úkony. Nakonec server vrací klientovi požadovaná data zapouzdřená v HTML šabloně. Na obrázku 5.2 je zobrazena dekompozice zpracování HTTP požadavku do jednotlivých komponent (zdrojových souborů).



**Obrázek 5.2.** Struktura frameworku Django, převzato z [18]

Níže jsou popsány jednotlivé zdrojové soubory z obrázku 5.2.

- **urls.py** - V tomto souboru jsou definovány všechny endpointy, které aplikace poskytuje. Soubor obsahuje mapovací funkce, ve kterých lze pomocí regulárních výrazů definovat různé podoby URL adres a na ně mapovat příslušná *View*.
- **views.py** - *View* je funkce, která zpracovává HTTP požadavek a vrací příslušnou odpověď. *View* na základě požadavku přistupuje k datům pomocí tzv. *Modelů*, definovaných v souboru *models.py*. K formátování HTTP odpovědi se poté využije Django šablon.



- **models.py** - V tomto souboru je pomocí Python tříd popsán datový model aplikace s operacemi nad jednotlivými daty.
- **šablony** - Šablony jsou textové soubory, které slouží pro generování HTML stránek nebo jiných dokumentů. V šabloně je definována struktura dané stránky, do které *View* vkládá data získaná z *Modelů*.

## 5.3 Registrace uživatele v Buddy App

První funkcionalitou, kterou bylo potřeba navrhnout a implementovat, bylo pozvání a registrace nového uživatele v Buddy App. Cílem bylo propojit účet uživatele Buddy App s účtem kuřáka, kterému pomáhá, tak aby měl přístup k některým informacím o procesu jeho odvykání.

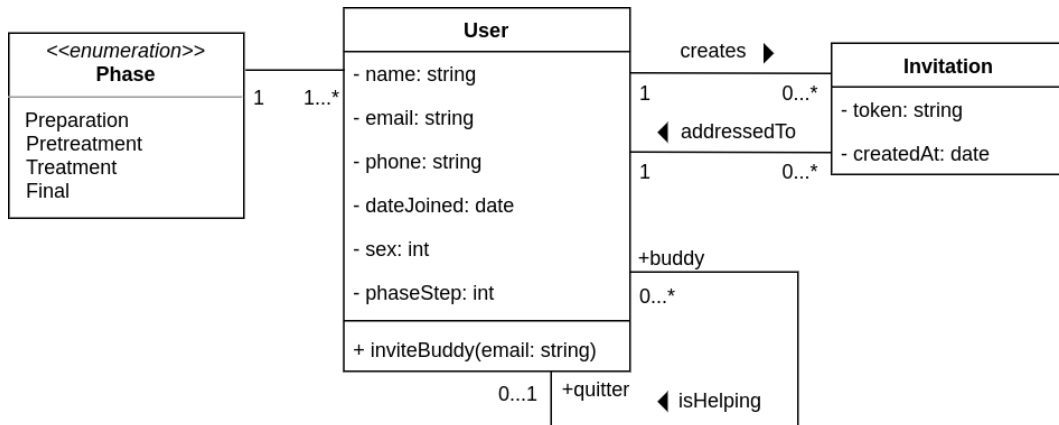
### 5.3.1 Proces pozvání a registrace uživatele

Registraci přítele kuřáka v aplikaci Buddy App musí předcházet samotná registrace kuřáka v hlavní odvykací aplikaci, se kterou bude Buddy App komunikovat. Dále bylo potřeba předejít nevyžádanému propojení účtu kuřáka s jiným účtem v Buddy App. Z toho důvodu bylo navrženo, aby registraci v Buddy App předcházelo pozvání od kuřáka. Scénář procesu pozvání a registrace je popsán níže.

1. Kuřák se registruje v hlavní odvykací aplikaci.
2. Kuřák v aplikaci zadá email nového uživatele, kterého chce pozvat do Buddy App a klikne na tlačítko "Invite buddy".
3. Systém uloží pozvánku a přiřadí k ní náhodně vygenerovaný kód.
4. Systém odešle kód pozvánky na emailovou adresu kuřáka a nového uživatele.
5. Uživatel se zaregistruje v Buddy App pomocí svého emailu a kódu pozvánky.
6. Systém ověří, zda existuje účet s emailem uživatele.
7. IF email uživatele je již přiřazen k jinému účtu
8. THEN systém uživateli zobrazí chybovou hlášku.
9. ELSE systém ověří, zda existuje pozvánka s kódem zadaným uživatelem.
10. IF pozvánka existuje
11. THEN systém vytvoří nový účet pro uživatele a propojí ho s autorem pozvánky.
12. ELSE systém uživateli zobrazí chybovou hlášku.

### 5.3.2 Doménový model

Doménový model je znázorněn na obrázku 5.3. Pro profil kuřáka a uživatele Buddy App byla použita stejná entita *User* s atributy popsány níže. Atribut *Phase* představuje fázi léčby, ve které se kuřák nachází v rámci hlavní odvykací aplikace. K propojení účtů kuřáka a uživatele Buddy App slouží reflexivní vazba *isHelping*. Tato vazba je vícenásobná, což umožňuje napojení více uživatelů Buddy App na stejného kuřáka. Entita *Invitation* představuje pozvánku s vygenerovaným kódem.



**Obrázek 5.3.** Doménový model Buddy App.

### 5.3.3 RESTful API a sekvenční diagram

Pro komunikaci Buddy App a odvykací aplikaci se serverem bylo navrženo RESTful API (viz [19]). Parametry jednotlivých HTTP požadavků pro pozvání a registraci uživatele v Buddy App jsou popsány v tabulce 5.1.

Pro vytvoření pozvánky do Buddy App se vyžaduje autentizace pomocí Bearer Token<sup>1</sup>. Tento token server vytváří při přihlášení uživatele do odvykací aplikace. Při registraci uživatele v Buddy App je podobný token vygenerován znovu.

Pokud je vytvoření pozvánky úspěšné, server vygeneruje sedmimístný registrační kód pro pozvaného uživatele, viz 5.3.1.

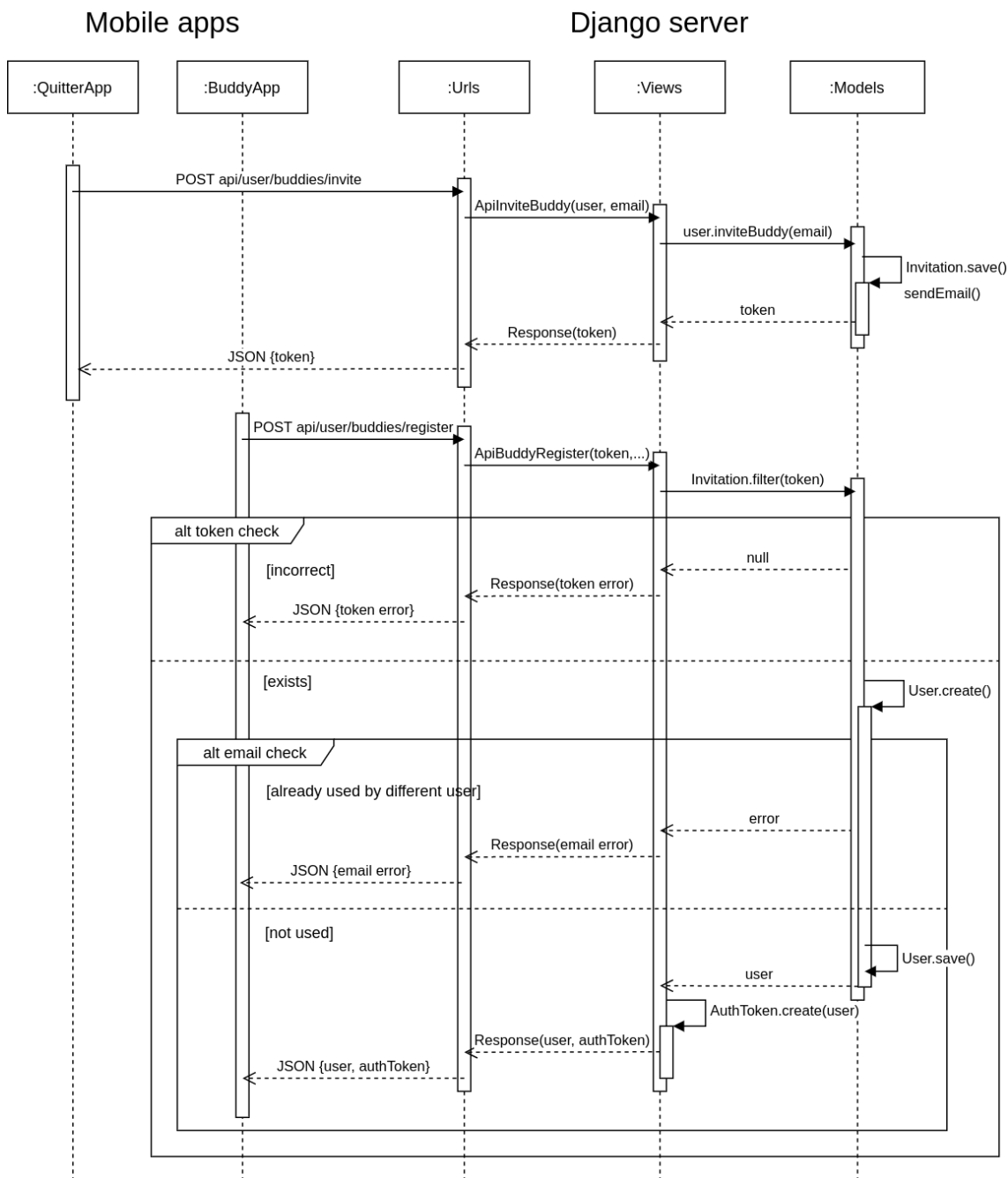
Parametry	Pozvání	Registrace
URL	/api/user/buddies/invite	/api/user/buddies/register
HTTP metoda	POST	POST
HTTP autentizace	Bearer Token	žádná
JSON objekt požadavku	email zvaného uživatele	email, jméno, heslo, pohlaví, jazyk, registrační token
JSON objekt odpovědi	úspěch (true/false), registrační token	objekt s informacemi o uživateli, autentizační token

**Tabulka 5.1.** Struktura HTTP požadavků pro pozvání a registraci uživatele v Buddy App.

Na sekvenčním diagramu 5.4 je znázorněna komunikace mobilních aplikací s vnitřními komponentami serveru. Jednotlivé komponenty serveru odpovídají Django komponentám popsaným v podkapitole 5.2.1. Komponenta *QuitterApp* představuje stávající aplikaci pro odvykání kouření.

V diagramu je také znázorněna validace registračního kódu (*alt token check*) a emailové adresy (*alt email check*).

<sup>1</sup> <https://oauth.net/2/bearer-tokens/>



**Obrázek 5.4.** Sekvenční diagram pozvání do Buddy App a registrace.

### 5.3.4 Implementace

Rozhraní bylo implementováno podle popsaného návrhu 5.4 ve všech třech komponentách serveru. V souboru `urls.py` byly přidány mapovací funkce na obě URL adresy. V komponentě `Views` byly vytvořeny dvě třídy `ApiInviteBuddy` a `ApiBuddyRegister`. V `models.py` byla přidána třída `BuddyInvite` představující pozvánku a ve třídě `User` přibyla metoda `invite_buddy` pro vytvoření nové pozvánky.

Jak již bylo zmíněno v podkapitole 5.2, Django řeší mnoho častých implementačních problémů za programátora. Na ukázce 5.5 je například využit vestavěný validátor emailových adres.

```

class ApiInviteBuddy(knox_views.LoginView):
    def post(self, request, format=None):
        validator = EmailValidator()

        try:
            validator(request.data.get('email'))
            invite = request.user.invite_buddy(request.data['email'])
        except ValidationError:
            return Response(
                {
                    'errors': {'email': _('Invalid buddy email provided.')},
                    'request': request.data,
                },
                status=status.HTTP_400_BAD_REQUEST
            )

        return Response({
            'success': True,
            'token': invite.token,
        })

```

**Obrázek 5.5.** Ukázka implementace třídy *ApiInviteBuddy*.

V klientské části aplikace Buddy App byla vytvořena obrazovka s registračním formulářem pro všechny údaje popsané v tabulce 5.1. Při potvrzení formuláře je pomocí JavaScript funkce *fetch*<sup>1</sup> provoláno implementované RESTful API. Na obrázku níže je náhled registračního formuláře.

The screenshot shows a mobile application interface for registration. At the top, the status bar shows 'T-Mobile CZ', signal strength, Wi-Fi, time '12:39', and battery '45%'. The app title is 'Registrace'. Below the title are four input fields: 'Křestní jméno', 'Email', 'Heslo', and 'Kód pozvánky'. Between the 'Email' and 'Heslo' fields are two radio buttons: 'Muž' (selected) and 'Žena'. Below the radio buttons is a key icon and the text: 'Zde prosím vlož nebo přepiš kód z pozvánky, která ti přišla emailem'. At the bottom of the form is a green button labeled 'ZAREGISTROVAT SE'.

**Obrázek 5.6.** Registrační formulář Buddy App.

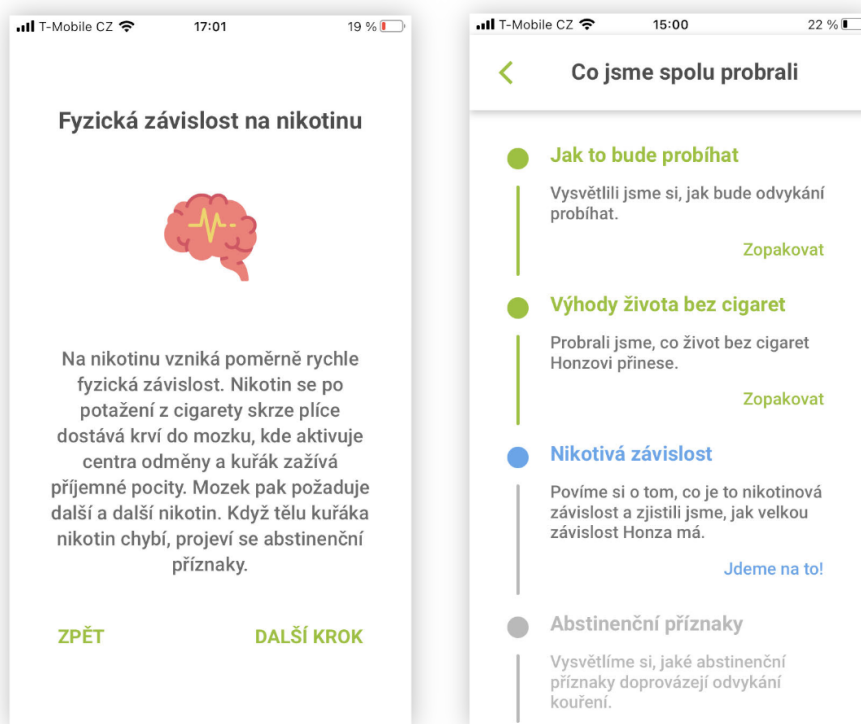
<sup>1</sup> [https://developer.mozilla.org/en-US/docs/Web/API/Fetch\\_API/Using\\_Fetch](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API/Using_Fetch)

## 5.4 Zobrazení pokroku v odvykání

Hlavní funkcí aplikace Buddy App je poskytování virtuálních informačních sezení o odvykání kouření. Tato sezení seznamují uživatele s průběhem odvykacího procesu a poskytují praktické rady, jak kuřáka podporovat. Cílem je, aby sezení v Buddy App byla synchronizována se sezeními určenými pro kuřáka v hlavní odvykací aplikaci.

Z toho důvodu bylo potřeba na serveru implementovat další rozhraní pro získání informací o kuřákovi, kterého uživatel podporuje. Podobně jako v podkapitole 5.3 byla vytvořena nová URL adresa směřující na *View* třídu, která pomocí vazby *isHelping* (viz 5.3.2) vrací informace o kuřákovi v JSON objektu.

Při otevření Buddy App je toto rozhraní provoláno, čímž se aktualizuje zobrazení pokroku kuřáka. Na základě získané informace o sezeních kuřáka se uživateli odemknou jeho dostupná sezení (nedojde tak k tomu, že by uživatel Buddy App byl v sezeních napřed). Na obrázku 5.7 je ukázka obrazovek ze sezení v Buddy App.



Obrázek 5.7. Ukázka sezení v Buddy App.

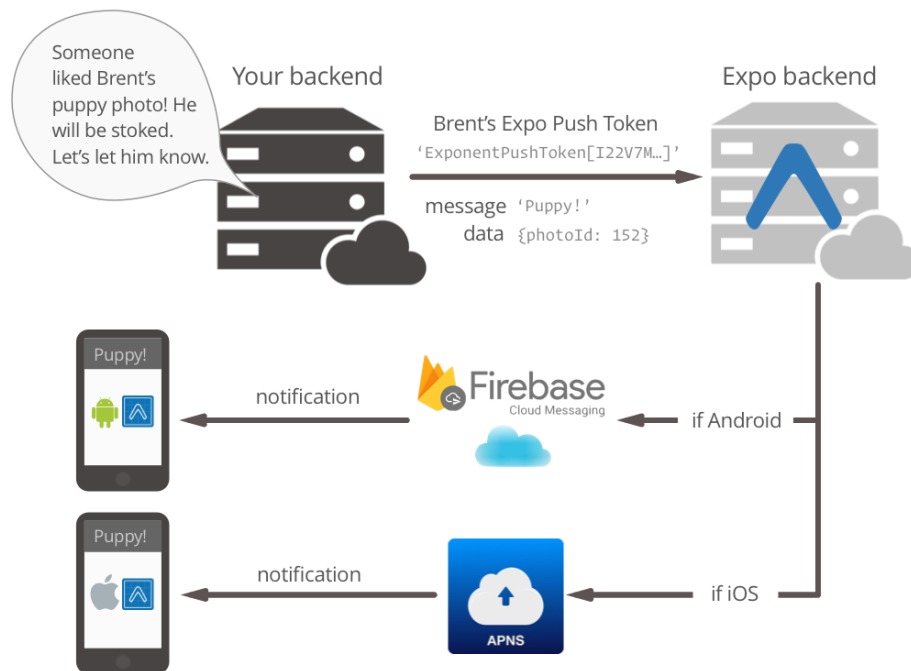
## 5.5 Expo push notifikace

Poslední funkcionalitou implementovanou na serveru bylo zasílání zpráv pomocí push notifikací přihlášeným uživatelům Buddy App. Tato funkcionalita je velice užitečná v případě, kdy chceme uživatele aplikace informovat v reálném čase o určitých událostech v procesu odvykání kuřáka. Například chceme uživatele upozornit, že kuřák právě dokončil určité sezení nebo že strávil první den bez cigarety. Notifikace se tak mohou stát důležitým prvkem v interakci uživatele s aplikací.

Protože je Buddy App vyvíjena ve vývojovém prostředí Expo (viz 3.4), k integrování push notifikací byla využita knihovna Expo Notifications<sup>1</sup>.

Knihovna Expo Notifications generuje pro každé zařízení unikátní *Expo push token*, který je pak využit při identifikaci příjemce notifikace. Prvním krokem v implementaci notifikací bylo generování a získání těchto *Expo push tokenů* od každého přihlášeného uživatele v Buddy App. Token byl vygenerován při prvním spuštění aplikace pomocí funkce *getExpoPushTokenAsync*, kterou knihovna poskytuje. Následně byl token přiložen k předchozím registračním údajům (viz 5.1) a odeslán na server.

V modelu *User* (viz 5.3.2) byl přidán nový atribut *expoPushToken* a metoda *send\_expo\_notification*. Notifikace jsou uživatelům zasílány pomocí Expo API na adrese `https://exp.host/--/api/v2/push/send`. Toto API přijímá HTTP POST požadavky s daty notifikace v JSON objektu v těle požadavku. Expo server na základě požadavku rozesílá příjemcům notifikace pomocí služeb *Firebase Cloud Messaging*<sup>2</sup> (Android) a *Apple Push Notification service*<sup>3</sup> (iOS). Schéma procesu je znázorněno na obrázku 5.8.



**Obrázek 5.8.** Schéma procesu zasílání notifikací pomocí Expo serveru, převzato z [20].

<sup>1</sup> <https://docs.expo.io/versions/latest/sdk/notifications/>

<sup>2</sup> <https://firebase.google.com/docs/cloud-messaging>

<sup>3</sup> <https://developer.apple.com/notifications/>

# Kapitola 6

## Uživatelské testování Buddy App

Tato kapitola popisuje uživatelské testování aplikace Buddy App. Podkapitola 6.1 popisuje jednotlivé fáze testování. V následující podkapitole 6.2 jsou popsány charakteristiky participantů. Podkapitola 6.3 obsahuje dotazník, který byl vyplněn jednotlivými participanty v rámci testování. Poslední podkapitola 6.3 shrnuje výsledky testování.

### 6.1 Průběh testování

Prvním krokem v testování byl výběr vhodných participantů. Dle Nielsen se nejlepší výsledky testu použitelnosti dostávají již při 5 participantech [21]. Testování se zúčastnilo celkem 7 participantů různých věkových kategorií (obrázek 6.1) a s různou pokročilostí v používání mobilních aplikací (obrázek 6.2).

V ideálním případě by testování aplikace mělo probíhat mezi participantem a kuřákem používající stávající odvykací aplikaci. Vzhledem ke koronavirové epidemii při psaní této práce nebylo možné tuto dvojici kuřáka a přítele zajistit. Participant testování tedy byli propojeni s profilem kuřáka, jehož proces odvykání byl simulován v odvykací aplikaci.

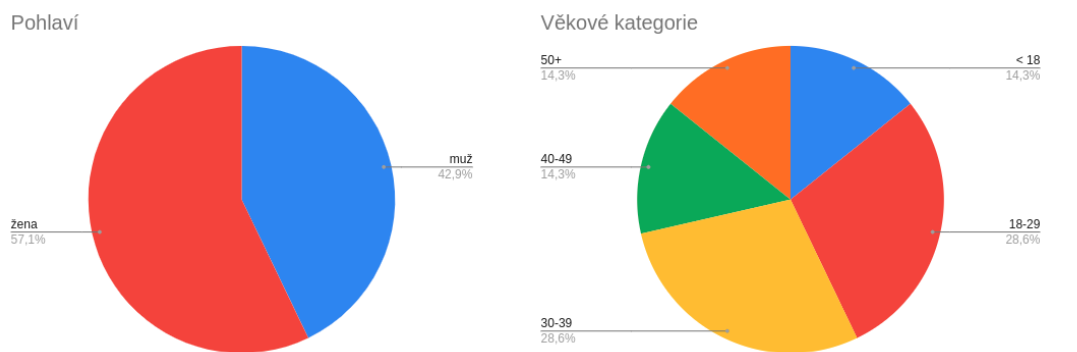
Participant byli nejprve seznámeni s účelem aplikace a průběhem testování. Byla jim vysvětlena simulace procesu odvykání a emailem zaslány registrační kódy (viz 5.3). Poté participant dostali zadání s úkoly popsány v sekci 6.3.1.

Na konci testování byl participantům předložen dotazník se sérií otázek zjišťujících spokojenost participantů s aplikací 6.3.2. Pomocí dotazníku je také zjištěno, jaký efekt měla aplikace na participanty z hlediska porozumění procesu odvykání a způsobů podpory kuřáka.

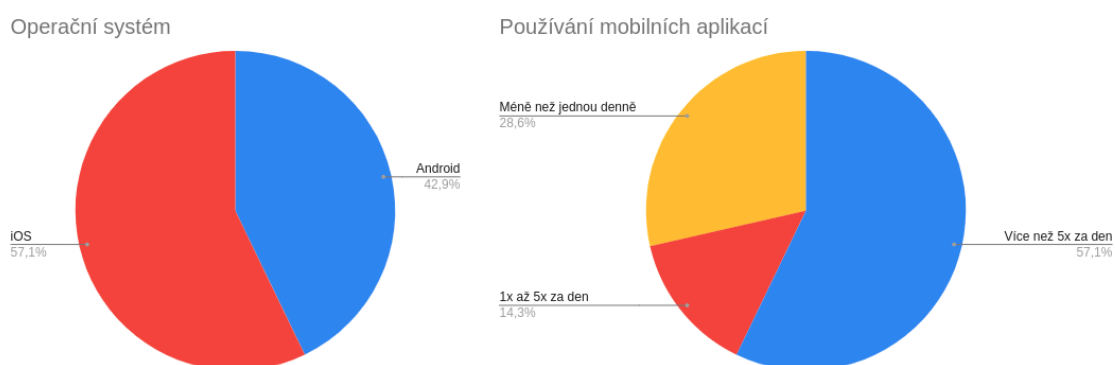
V závěru byly odpovědi a připomínky z dotazníku vyhodnoceny a bylo navrženo případné rozšíření aplikace o nové funkcionality 6.4.

### 6.2 Participant

Participantům testování byl nejdříve předložen krátký dotazník pro zjištění zastoupení pohlaví, věkových kategorií, testovaných platforem a pokročilosti v používání mobilních aplikací v rámci testování. Výsledky tohoto dotazníku jsou zobrazeny v grafech 6.1 a 6.2.



**Obrázek 6.1.** Grafy se zastoupením pohlaví a věkových kategorií participantů testování.



**Obrázek 6.2.** Graf se zastoupením iOS a Android uživatelů a graf s pokročilostí uživatelů s používáním mobilních aplikací.

## 6.3 Testovací dotazník

Po seznámení participantů s aplikací Buddy App jim byl předložen dotazník sestávající se ze dvou částí. První část obsahovala seznam úkolů, které participanti měli v rámci testování splnit 6.3.1. V druhé části participanti odpovídali na otázky ohledně jejich postřehů z testování 6.3.2.

### 6.3.1 Zadání

1. Zaregistrujte se v aplikaci pomocí registračního kódu, zaslaným na vaši emailovou adresu.
2. Zjistěte, jaké jsou fáze v procesu odvykání kouření.
3. Každý den projděte nově dostupné sezení.
4. Při procházení jednotlivých sezení, zjistěte tyto informace:
  - Jaké jsou výhody života bez cigaret?
  - Co je podstatou závislosti na cigaretách?
  - Jaké abstinční příznaky se mohou objevit při odvykání kouření?
  - Jak můžete kuřáka při odvykání podporovat?



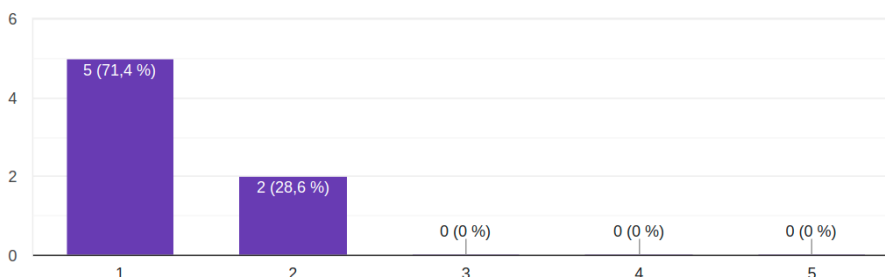
### 6.3.2 Otázky

1. Jak obtížná pro vás byla registrace v aplikaci na škále od 1 do 5? (1 - velice snadná)
2. Preferovali byste jiný druh registrace? Jaký?
3. Narazili jste na nějaký problém během procházení jednotlivých sezení?
4. Jsou nějaká témata, která vám v sezeních scházela a o kterých byste rád/a věděl víc?
5. Máte pocit, že lépe rozumíte procesu odvykání kouření?
  - Rozhodně ne
  - Spíše ne
  - Spíše ano
  - Rozhodně ano
6. Máte pocit, že víte, jak kuřáka podpořit v odvykání?
  - Rozhodně ne
  - Spíše ne
  - Spíše ano
  - Rozhodně ano
7. Co se vám na aplikaci líbilo nejvíce?
8. Co se vám na aplikaci líbilo nejméně?
9. Jak byste ohodnotili užitečnost aplikace na škále od 1 do 10?
10. Jakou další funkci byste v aplikaci ocenili?

## 6.4 Výsledky

Všech 7 participantů dokázalo bez větších problémů nainstalovat a spustit aplikaci. Registrace probíhala také téměř bez jakýchkoli problémů. Pouze v jediném případě byl uživatel zmaten a místo registrace se snažil přihlásit. Všichni participanté věděli jak použít registrační kód z emailové zprávy a úspěšně se zaregistrovali. Obtížnost registrace byla na škále 1 až 5 (1 - velice snadná) ohodnocena známkou 1.28, participantům tedy vyhovovala (viz obrázek 6.3).

Jak obtížná pro vás byla registrace v aplikaci na škále od 1 do 5?



**Obrázek 6.3.** Ohodnocení obtížnosti registrace v Buddy App (1 - velice snadná).

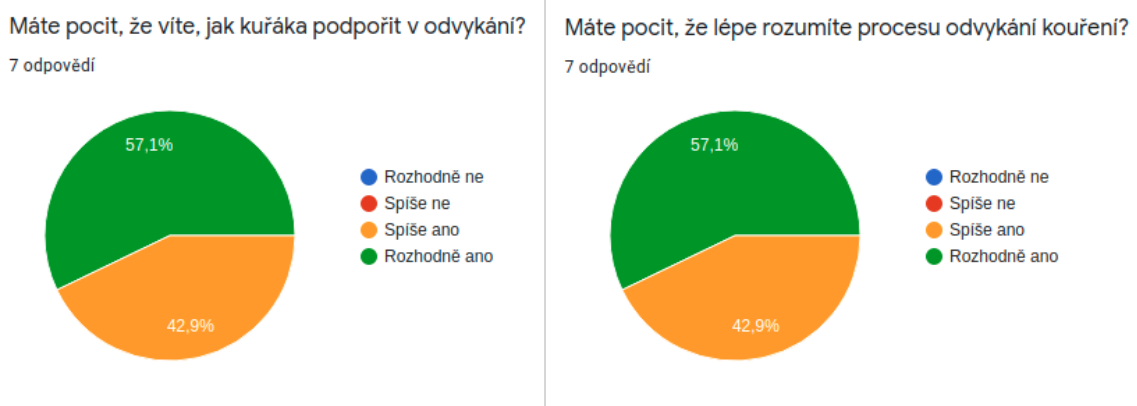
Pouze jeden z participantů navrhl alternativní způsob registrace, který by nevyžadoval registrační kód. Aplikace by se tak stala dostupnější a sloužila by pouze jako znalostní báze dokud by se kuřák v odvykací aplikaci nepropojil s uživatelem v

Buddy App pomocí emailové adresy, se kterou se uživatel zaregistroval. Propojení v systému by tak mohlo proběhnout automaticky.

Během procházení jednotlivých sezení nedošlo téměř k žádným technickým problémům. Synchronizace sezení v Buddy App a odvykací aplikaci probíhala podle očekávání.

Většina participantů uvedla, že jim seznam probraných témat připadal kompletní. Mezi návrhy na další témata patřily obecné informace o odvykání jako jsou statistiky ukazující jak dlouho odvykání průměrně trvá, jaká je průměrná úspěšnost nebo u jakých věkových kategorií hlavně k závislosti dochází. Nicméně tato témata se netýkala přímo podpory kuřáka.

Téměř všichni participanté uvedli, že lépe rozumí procesu odvykání a tomu, jak kuřáka podpořit (viz obrázek 6.4).



**Obrázek 6.4.** Odpovědi na otázky 5 a 6 z dotazníku .

Mezi funkce aplikace, které participanté nejvíce ocenili, patřily tyto:

- Srozumitelné a dobře vysvětlené procesy odvykání, kterými si kuřák prochází
- Modelové rozhovory s kuřákem a celkové zaměření na praktickou podporu a povzbuzování kuřáka v odvykání
- Pozitivní přístup virtuálního terapeuta se zachováním potřebné míry reality
- Simulace reálného rozhovoru (oslovení jménem)
- Jednoduchý a přehledný design

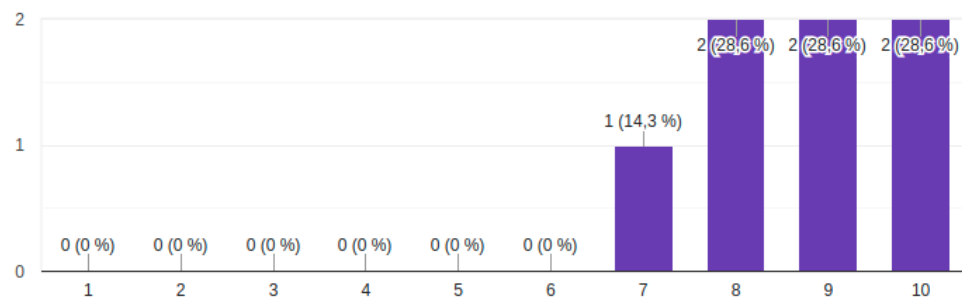
Tři participanté ze sedmi uvedli, že by uvítali více interakce s kuřákem. Mezi další připomínky patřily tyto:

- Chybí zobrazení v kolikátém kroku sezení se uživatel nachází a kolik mu zbývá do konce
- Chybí nastavení jazyka či nočního režimu

Celkovou užitečnost aplikace participanté ohodnotili známkou 8.7 z 10 (viz graf 6.5).

Jak byste ohodnotili užitečnost aplikace na škále od 1 do 10?

7 odpovědí



**Obrázek 6.5.** Výsledky hodnocení užitečnosti aplikace (10 - velice užitečná).

Participantů dále navrhli rozšíření aplikace o tyto funkcionality:

- Každodenní návrh jídelníčku, cvičení a jiných fyzických aktivit pro závislého
- Interakce mezi několika podporovateli závislého např. formou skupinového chatu
- Možnost kuřáka informovat svého přítele, jak se dnes cítí, jakou má náladu (zmíněno 2x)
- Propojení plánovaných sezení s kalendářem

Z výsledků dotazníku lze usoudit, že participantů byli s aplikací obecně spokojeni a ocenili především její informační stránku. Mezi hlavní nedostatky aplikace se zdá být nízká míra interakce s kuřákem. Mezi některé návrhy na zlepšení patří sdílení nálady kuřáka s uživatelem aplikace či skupinový chat všech podporovatelů. Rozšíření aplikace se diskutuje dále v kapitole 7.1.

# Kapitola 7

## Závěr

Cílem práce bylo rozšíření aplikace pro odvykání kouření o multiplatformní modul šesti miniher a propojení s aplikací Buddy App. Dále bylo navrženo serverové rozhraní pro registraci v Buddy App, získávání statistik o kuřákovi a zasílání zpráv pomocí notifikací. Na závěr byla aplikace Buddy App otestována na 7 uživatelích.

Během práce došlo k seznámení se s frameworkem React Native pro vývoj multiplatformních aplikací. Byly zjištěny jeho výhody i nedostatky v porovnání s vývojem aplikací nativních 4.7. Mezi zřejmé výhody patří rychlejší a pohodlnější způsob vývoje za využití jednotného programovacího jazyka pro obě platformy. Oproti vývoji nativních aplikací je ale vývoj v tomto frameworku závislý na React Native knihovnách, které nemusí být vždy dostatečně udržované.

Během vývoje modulu miniher došlo k řešení několika zajímavých problémů. Mezi ně patřila např. navigace v aplikaci 4.1, animace 2D objektů ve hře 2048 4.2.3 a implementace virtuálního protihráče ve hrách Reversi 4.3 a Piškvorky 4.4. Ve třech zbývajících hrách byl využit herní engine a akcelerometr k simulaci navigace kuliček pomocí naklánění zařízení a srážek s jinými objekty 4.5.2. Zajímavá byla i implementace herní logiky ve hře Bublíny, kde byl využit algoritmus prohledávání grafu do hloubky 4.6.2.

V druhé části práce došlo k návrhu a implementaci několika serverových rozhraní pro komunikaci aplikace Buddy App s hlavní odvykací aplikací. Nejprve došlo k seznámení se s webovým frameworkem Django a s jeho vlastnostmi 5.2. Dále byl navrhnut a implementován proces pozvání a registrace nového uživatele v Buddy App 5.3. Pro synchronizaci sezení v obou aplikacích bylo vytvořeno API s informacemi o procesu odvykání kuřáka 5.4. Nakonec bylo využito Expo push notifikací k zasílání zpráv ze serveru do Buddy App 5.5.

V závěru proběhlo uživatelské testování funkčnosti implementovaných řešení a použitelnosti aplikace Buddy App. Participantů testování nezaznamenali žádné problémy v propojení Buddy App s odvykací aplikací. Většina z nich na konci testování oznámila, že mnohem lépe rozumí procesu odvykání a vědí, jak kuřáka podpořit 6.4. Participantů také ocenili pozitivní a osobní přístup virtuálního terapeuta.

## 7.1 Rozšíření aplikace

Několik participantů nezávisle na sobě uvedlo, že by uvítali více interakce s kuřákem. Mezi navrhované rozšíření aplikace o nové funkcionality patřilo zaznamenávání a sdílení nálady kuřáka s jeho přítelem. Tato funkcionality by mohla značně přispět ke zvýšení interakčnosti aplikace a podpoře kuřáka v okamžiku, kdy to nejvíce potřebuje.

Kuřák by mohl v odvykací aplikaci zvolit jak se cítí na určité škále emocí označené emotikony. S vědomím, že jeho přítel v Buddy App označenou náladu uvidí, by se zaznamenaná data uložila na server do modelu *User* (viz doménový model 5.3.2) a následně zaslala pomocí notifikací do Buddy App. Uživatel Buddy App by tímto způsobem mohl okamžitě vědět, jaký druh podpory kuřák potřebuje.



## Literatura

- [1] World Health Organization. *Fact Sheets - Tobacco*. 2020.  
<https://www.who.int/news-room/fact-sheets/detail/tobacco>. [online] (navštíveno dne 2020-04-12).
- [2] John R. Hughes, Josue Keely a Shelly Naud. *Shape of the relapse curve and longterm abstinence among untreated smokers*. 2004.
- [3] A. Kulháněk, R. Gabrhelík, D. Novák, V. Burda a H Brendryen. *eHealth Intervention for Smoking Cessation for Czech Tobacco Smokers: Pilot Study of User Acceptance*. 2019.
- [4] Hana Sovinová, Ladislav Csémy, Bohumil Procházka a Stanislava Kottnaurová. *Smoking-attributable mortality in the Czech Republic, Journal of Public Health volume 16, 37–42*. 2008.  
<https://link.springer.com/article/10.1007/s10389-007-0116-2>.
- [5] Státní zdravotní ústav. *Užívání tabáku a alkoholu v České republice*. 2019.  
[http://www.szu.cz/uploads/documents/szu/aktual/zprava\\_tabak\\_alkohol\\_cr\\_2019.pdf](http://www.szu.cz/uploads/documents/szu/aktual/zprava_tabak_alkohol_cr_2019.pdf).
- [6] J. Greplová. *Prevalence kouření a postoje k užívání tabáku mezi nezletilými v České republice. Adiktologie, 13(3–4), 240–247*. 2013.  
[http://199757.w57.wedos.ws/wp-content/uploads/2018/09/2013\\_3-4\\_Greplova\\_Prevalence-koureni.pdf](http://199757.w57.wedos.ws/wp-content/uploads/2018/09/2013_3-4_Greplova_Prevalence-koureni.pdf).
- [7] Eva Králíková, Richard Česka, Alexandra Pánková, Lenka Štěpánková, Kamila Zvolská, Vladislava Felbrová, Stanislava Kulovaná a Miroslav Zvolský. *Doporučení pro léčbu závislosti na tabáku, Vnitř Lék 2015; 61(5, Suppl 1): 1S4–1S15*. 2015.  
[https://www.kardio-cz.cz/data/upload/Doporuceni\\_pro\\_lecibu\\_zavislosti\\_na\\_tabaku.pdf](https://www.kardio-cz.cz/data/upload/Doporuceni_pro_lecibu_zavislosti_na_tabaku.pdf).
- [8] John R. Hughes. *Effects of abstinence from tobacco: valid symptoms and time course*. 2007.
- [9] Lim G, Park I, Park S, Song S, Kim H a Kim S. *Effectiveness of smoking cessation using motivational interviewing in patients consulting a pulmonologist*. 2014.
- [10] *Mobile Operating System Market Share Worldwide*.  
<https://gs.statcounter.com/os-market-share/mobile/worldwide>. [online] (navštíveno dne 2020-04-12).
- [11] *Progressive web apps (PWAs)*.  
[https://developer.mozilla.org/en-US/docs/Web/Progressive\\_web\\_apps](https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps). [online] (navštíveno dne 2020-04-12).
- [12] *Cross-platform mobile frameworks used by developers worldwide 2019 and 2020*.

- <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>. [online] (navštíveno dne 2020-04-12).
- [13] *Expo documentation*.  
<https://docs.expo.io/introduction/why-not-expo/>. [online] (navštíveno dne 2020-04-12).
- [14] Václav Šídlo. *Rozšíření mobilní aplikace pro podporu odvykání kouření*. 2020.
- [15] *Most popular web frameworks among developers worldwide 2020*.  
<https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web/>. [online] (navštíveno dne 2020-04-12).
- [16] *Security in Django*.  
<https://docs.djangoproject.com/en/3.2/topics/security/>. [online] (navštíveno dne 2020-04-12).
- [17] Dave Wright. *The Advantages of a Shared Nothing Architecture for Truly Non-Disruptive Upgrades*. 2014.  
<https://blog.netapp.com/blogs/the-advantages-of-a-shared-nothing-architecture-for-truly-non-disruptive-upgrades/>.
- [18] *Django introduction, MDN Web Docs*.  
<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction>. [online] (navštíveno dne 2020-04-12).
- [19] *What is REST*.  
<https://restfulapi.net/>. [online] (navštíveno dne 2020-04-12).
- [20] *Sending Notifications with Expo's Push API*.  
<https://docs.expo.io/push-notifications/sending-notifications/>.
- [21] Jakob Nielsen. *Why You Only Need to Test with 5 Users*.  
<https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>.



# Příloha **A**

## Seznam zkratk

<b>API</b>	Application Programming Interface
<b>ČR</b>	Česká republika
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	Hypertext Transfer Protocol
<b>JSON</b>	JavaScript Object Notation
<b>JSX</b>	JavaScript XML
<b>PWA</b>	Progressive Web App
<b>SDK</b>	Software Development Kit
<b>SQL</b>	Structured Query Language
<b>REST</b>	Representational State Transfer
<b>RSS</b>	RDF Site Summary or Really Simple Syndication
<b>UI</b>	User Interface
<b>UX</b>	User Experience
<b>XML</b>	Extensible Markup Language



## Příloha **B**

### Obsah elektronické přílohy

minihry/ zdrojový kód modulu Miniher  
buddy\_app/ zdrojový kód serverového rozhraní aplikace Buddy App