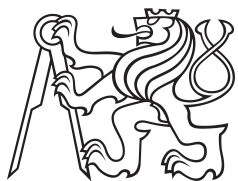


Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačů

## Návrh a tvorba frontend strany platformy pro podporu gastronomie

**Martin Nešněra**

Vedoucí: Ing. Pavel Náplava, Ph.D.  
Obor: Softwarové inženýrství a technologie  
Květen 2022



## I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Nešněra** Jméno: **Martin** Osobní číslo: **491916**  
Fakulta/ústav: **Fakulta elektrotechnická**  
Zadávací katedra/ústav: **Katedra počítačů**  
Studijní program: **Softwarové inženýrství a technologie**

## II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

**Návrh a tvorba frontend strany platformy pro podporu gastronomie**

Název bakalářské práce anglicky:

**Design and implementation of a gastronomy platform frontend**

Pokyny pro vypracování:

Navrhněte a implementujte frontend část platformy pro podporu gastronomie, založenou na soutěžení zákazníků. Frontend bude realizován formou webové aplikace pro správu soutěží, a mobilní zákaznické aplikace, umožňující účastnit se aktivních soutěží. Soutěže budou probíhat formou získávání kreditů na základě načtených účtenek (útraty), soutěžení mezi zákazníky a získání bonusů dle pravidel, definovaných při tvorbě soutěže.

Postupujte následovně:

- 1) Analyzujte požadavky frontend části platformy, včetně procesních diagramů, demonstrujících práci uživatelů v aplikacích.
- 2) Navrhněte design mobilní a webové aplikace.
- 3) Zvolte vhodnou architekturu a technologii pro vývoj aplikací, reflektujte napojení na backend, tvořený v rámci bakalářské práce Michala Šalagy.
- 4) Navržené aplikace realizujte.
- 5) Proveďte ověření funkčnosti aplikací pomocí uživatelských testů.
- 6) Vhodným způsobem ověřte napojení na backend část aplikace.
- 7) Zhodnoťte využitelnost aplikace a navrhněte vhodný obchodní model.

Seznam doporučené literatury:

1. React. Dostupné z: <https://reactjs.org/>
2. React Native. Dostupné z: <https://reactnative.dev/>
3. Redux Toolkit. Dostupné z: <https://redux-toolkit.js.org/>

Jméno a pracoviště vedoucí(ho) bakalářské práce:

**Ing. Pavel Náplava, Ph.D. Centrum znalostního managementu FEL**

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

Datum zadání bakalářské práce: **02.02.2022**

Termín odevzdání bakalářské práce: **20.05.2022**

Platnost zadání bakalářské práce: **30.09.2023**

Ing. Pavel Náplava, 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

Děkuji vedoucímu mé práce Ing. Pavlu Náplavovi, Ph.D. za jeho profesionální přístup, cenné rady a ochotu, která mi byla vždy poskytnuta v rámci průběžných konzultací i mimo ně. Dále bych chtěl poděkovat všem účastníkům proběhlého uživatelského testování, kteří si pro účely testování dobrovolně vyhradili svůj čas.

## Prohlášení

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

V Praze, 10. května 2022

## Abstrakt

Cílem této bakalářské práce je analýza, návrh, implementace a testování základní funkcionality klientské části soutěžní platformy pro podporu gastronomického odvětví. Platforma se skládá z multiplatformní mobilní aplikace dostupné pro širokou veřejnost a webové aplikace, která umožní správu jednotlivých událostí. Práce navazuje na analýzu a implementaci serverové části, realizované v rámci bakalářské práce kolegy Michala Šalagy. Součástí práce je i zhodnocení, zda má výsledná platforma potenciál pro využití v obchodním světě.

**Klíčová slova:** Klientská strana, soutěžní platforma, React, Redux, uživatelské testování, obchodní model

**Vedoucí:** Ing. Pavel Náplava, Ph.D.

## Abstract

The aim of this thesis was analysis, design, implementation and testing of basic functionality of gastronomy platform client side. Designed platform is composed from multiplatform mobile application, which is available for business customers, and web application, which is used to manage individual events. This thesis follows up to analysis and implementation of server side, realized by colleague Michal Šalaga in his bachelor thesis. Evaluation of platform's business potential is also part of this thesis.

**Keywords:** Client side, competition platform, React, Redux, usability testing, business model

**Title translation:** Design and implementation of a gastronomy platform frontend

# Obsah

<b>1 Úvod</b>	<b>1</b>	6.3.2 Socket.io Client API.....	40
1.1 Původní motivace.....	1	<b>7 Testování</b>	<b>41</b>
1.2 Struktura práce .....	2	7.1 Průběh testování .....	41
<b>2 Popis řešení</b>	<b>3</b>	7.2 Testovací scénáře .....	42
2.1 Myšlenka .....	3	7.2.1 Scénář 1 - Registrace nového uživatele mobilní aplikace.....	42
2.2 Rozdělení práce .....	4	7.2.2 Scénář 2 - Přihlášení uživatele mobilní aplikace .....	42
2.3 Mobilní a webová aplikace .....	4	7.2.3 Scénář 3 - Přihlášení uživatele webové aplikace .....	42
2.4 Aktuální situace .....	5	7.2.4 Scénář 4 - Změna hesla uživatele .....	43
<b>3 Existující řešení</b>	<b>7</b>	7.2.5 Scénář 5 - Zablokování uživatele .....	43
3.1 Beersport .....	7	7.2.6 Scénář 6 - Vytvoření účtu podniku .....	43
3.2 Trifft .....	8	7.2.7 Scénář 7 - Vytvoření události	44
3.3 Vyhodnocení .....	9	7.2.8 Scénář 8 - Schválení účtenky	44
<b>4 Analýza požadavků</b>	<b>11</b>	7.2.9 Scénář 9 - Ukončení události	45
4.1 Analýza funkčních a nefunkčních požadavků .....	11	7.2.10 Scénář 10 - Potvrzení transakce .....	45
4.1.1 Mobilní aplikace .....	11	7.3 Výstupy testování.....	46
4.1.2 Webová aplikace .....	13	<b>8 Využití platformy na trhu</b>	<b>47</b>
4.2 Diagramy případů užití .....	15	8.1 SWOT analýza .....	47
4.2.1 Mobilní aplikace .....	16	8.1.1 Silné stránky .....	47
4.2.2 Webová aplikace .....	20	8.1.2 Slabé stránky.....	48
<b>5 Návrh klientské strany platformy</b>	<b>25</b>	8.1.3 Příležitosti .....	48
5.1 Procesní diagramy .....	25	8.1.4 Hrozby .....	48
5.1.1 Proces realizace události ....	25	8.1.5 Vyhodnocení SWOT analýzy	48
5.1.2 Proces zpracování transakce .	26	8.2 Obchodní model platformy.....	48
5.2 Prototypování klientských aplikací	27	8.2.1 Klíčoví partneři .....	48
5.2.1 Použitý software pro tvorbu prototypů .....	28	8.2.2 Klíčové aktivity .....	49
5.2.2 Testování low-fidelity prototypů .....	30	8.2.3 Klíčové zdroje .....	49
5.3 Technologie .....	31	8.2.4 Poskytovaná hodnota.....	49
5.3.1 Framework .....	31	8.2.5 Vztahy se zákazníky .....	49
5.3.2 Globální správa stavů .....	32	8.2.6 Distribuční kanály .....	49
5.4 Architektura platformy .....	33	8.2.7 Zákaznické segmenty .....	50
5.4.1 MERN Stack .....	33	8.2.8 Struktura nákladů .....	50
5.4.2 REST .....	35	8.2.9 Zdroje příjmů .....	51
5.4.3 Socket.io .....	35	8.2.10 Náklady za cloudové technologie .....	52
5.4.4 OAuth 2.0 .....	35	8.2.11 Vyhodnocení obchodního modelu .....	52
<b>6 Implementace</b>	<b>37</b>	<b>9 Závěr</b>	<b>53</b>
6.1 Architektura klientských aplikací	37	9.1 Budoucí stav .....	53
6.2 Struktura projektů.....	38		
6.2.1 Mobilní aplikace .....	39		
6.2.2 Webová aplikace .....	39		
6.3 Komunikace se serverovou částí.	40		
6.3.1 Fetch Base Query .....	40		

<b>Literatura</b>	<b>55</b>
<b>Appendices</b>	<b>59</b>
<b>A Low-fidelity prototypy</b>	<b>59</b>
A.1 Low-fidelity prototypy mobilní aplikace .....	59
A.2 Low-fidelity prototypy webové aplikace .....	62
<b>B High-fidelity prototypy</b>	<b>65</b>
B.1 High-fidelity prototypy mobilní aplikace .....	65
B.2 High-fidelity prototypy webové aplikace .....	68
<b>C Seznam použitých zkratk</b>	<b>71</b>
<b>D Obsah přiloženého komprimovaného souboru</b>	<b>73</b>



## Obrázky

2.1 Grafický popis funkce platformy .	4	B.2 Prototyp stránky s detailem události .....	66
3.1 Ukázka uživatelského rozhraní mobilní aplikace Beersport .....	8	B.3 Prototyp profilové stránky .....	67
3.2 Ukázka uživatelského rozhraní mobilní aplikace Trifft .....	9	B.4 Prototyp stránky s odměnami ..	67
4.1 Diagram případů užití nepřihlášeného uživatele mobilní aplikace .....	16	B.5 Prototyp stránky s aktivovanou odměnou .....	68
4.2 Diagram případů užití přihlášeného uživatele mobilní aplikace .....	17	B.6 Prototyp úvodní stránky s transakcemi .....	68
4.3 Diagram případů užití nepřihlášeného uživatele webové aplikace .....	20	B.7 Prototyp stránky s podniky ....	69
4.4 Diagram případů užití přihlášeného uživatele webové aplikace .....	21	B.8 Prototyp stránky s detailem podniku .....	69
5.1 Diagram procesu realizace události .....	26	B.9 Prototyp stránky s událostmi ..	70
5.2 Diagram procesu zpracování transakce .....	27	B.10 Prototyp stránky s detailem události .....	70
5.3 Screenshot aplikace Adobe XD .	29		
5.4 Screenshot aplikace Figma .....	29		
5.5 Grafický popis souboru technologií MERN Stack [9] .....	34		
6.1 Grafický popis knihovny Redux Toolkit [8] .....	38		
A.1 Prototyp úvodní stránky s událostmi .....	59		
A.2 Prototyp stránky s detailem události .....	60		
A.3 Prototyp stránky s účtenkami ..	60		
A.4 Prototyp profilové stránky .....	61		
A.5 Prototyp stránky s odměnami ..	61		
A.6 Prototyp stránky s aktivovanou odměnou .....	62		
A.7 Prototyp úvodní stránky s transakcemi .....	62		
A.8 Prototyp stránky s podniky ...	63		
A.9 Prototyp stránky s detailem podniku .....	63		
A.10 Prototyp stránky s událostmi .	64		
B.1 Prototyp úvodní stránky s událostmi .....	65		

## Tabulky

7.1 Výstupy uživatelského testování 46

# Kapitola 1

## Úvod

V této části práce uvedu původní motivaci, která mě a kolegu Michala Šalagu vedla k volbě tohoto tématu bakalářské práce.

### 1.1 Původní motivace

Nacházíme se v pandemické době. Ať chceme či nikoli, pandemie každého z nás zásadně ovlivnila fyzicky či psychicky. Po celém světě byla a jsou zaváděna různá opatření, která mají zabránit šíření nákazy a ztrátám na lidských životech. Otázkou, zda všechna opatření v této době měla své opodstatnění, se ve své práci zabývat nebudu. Pokusím se ale snížit dopad jednoho z mnoha problémů, který jsme s kolegu určili jako možný následek pandemie.

Oním problémem, kterým se v rámci této práce zabývám jsou finanční ztráty gastronomických podniků. Mnoho z nich od počátku pandemie zkrachovalo a ty, které přežily tuto krizi, zažily obrovský propad tržeb. Provozovatelé podniků byli následně nuceni hledat řešení, která by potlačila dopady pandemie a umožnila jim alespoň provizorně fungovat. Jedním z provizorních řešení bylo otevírání výdejních okének, kam si mohli stálí zákazníci zajít a vyzvednout svou objednávku. Další možností bylo navázání spolupráce s kurýrními společnostmi, jako jsou Wolt, Bolt, či Dáme Jídlo. Čím dál více podniků začalo s nimi spolupracovat, aby si mohli kompenzovat své ztráty nehledě na to, zda s jejich obchodními modely souhlasily. Tyto společnosti sice pomáhají zvládnout situaci v podnicích, ale berou si za to nemalé provize z každé objednávky, která je pro podniky velkým problémem. Kromě zmíněných provizí přispělo zvýšené využívání kurýrních společností i ke snížení návštěvnosti v době, kdy mnohá z opatření nebyla platná.

Právě pokles motivace osobně navštěvovat tyto podniky je problém, kterým se v rámci této práce zabýváme. Lidé si zkrátka zvykli na servis, díky kterému se nemusí vzdálit ze svého domova a aktuálně neexistuje žádné účinné řešení tohoto problému. Je třeba nalézt takové řešení, které by zákazníkům přineslo kromě uspokojení jejich stravovacích potřeb i nějakou přidanou hodnotu, která by je současně motivovala do podniků přijít.

Cílem této bakalářské práce je návrh a implementace takového řešení, které by mohlo pomoci s řešením nízké návštěvnosti v gastronomických podnicích a nenutilo by majitele podniků k placení vysokých provizí z každého prodeje.

## ■ 1.2 Struktura práce

Na základě výše uvedené motivace 1.1 se ve své práci zabývám nalezením řešení, které bude řešit danou problematiku podniků z oblasti gastronomie. Navržené řešení porovnávám s existujícími řešeními a hodnotím, zda má smysl jej realizovat. Poté pro něj provádím analýzu požadavků a navrhuji vhodný design. Dále provádím rešerši potřebných technologií, potřebných pro implementaci řešení. V části, věnující se implementaci, popisuji strukturu a architekturu nalezeného řešení a uvádím, jak komunikuje se serverovou částí kolegy Michala Šalagy [10]. Práce je završena testováním nalezeného řešení a určením jeho možného využití na trhu.

## Kapitola 2

### Popis řešení

Následující kapitola obsahuje popis myšlenky námi navrženého řešení. Jelikož původní motivace 1.1 a myšlenka 2.1 vycházejí z doby, kdy byla v naší zemi platná mnohá opatření, omezující volný pohyb osob, rozhodl jsem se uvést, jak byla změněna původní motivace, když se situace uvolnila.

### 2.1 Myšlenka

S kolegou Michalem Šalagou jsme se zabývali nalezením možného řešení, které by potencionálně mohlo zvýšit návštěvnost v těchto podnicích. Napadlo nás vytvořit soutěžní platformu, skládající se z multiplatformní mobilní a webové aplikace, určené pro správu událostí či transakcí.

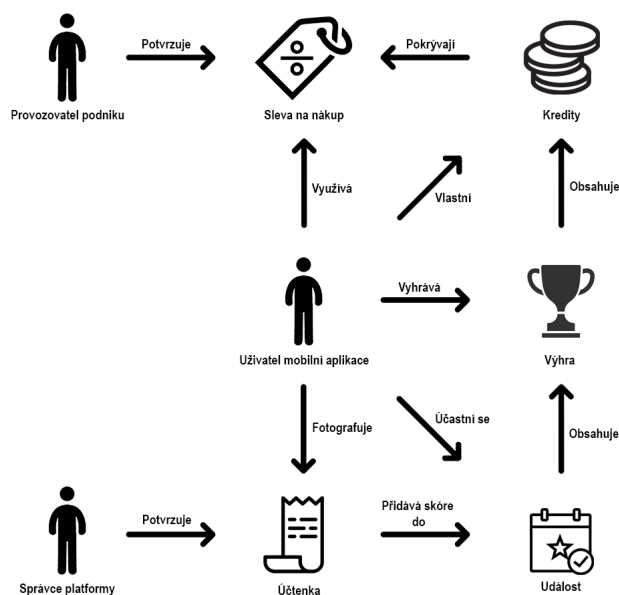
Hlavním cílem této platformy je organizace časově omezených událostí v s námi spolupracujících podnicích. Těchto událostí se mohou účastnit všichni uživatelé mobilní aplikace a následně mohou mezi sebou soutěžit v tom, kdo v průběhu události v podniku utratí nejvíce za libovolné položky v menu podniku. Účastníci mohou fotit účtenky pomocí mobilní aplikace a správce platformy je ve webové aplikaci manuálně potvrdí, či zamítne. V případě potvrzení se uživatelé přičte skóre, odpovídající jeho útratě. Po skončení události obdrží účastníci s nejvyšším počtem bodů virtuální měnu (dále zmíněnou jako *kredity*), kterou mohou kdykoli v uplatnit v podniku, ve kterém je vyhráli. Kredity uplatní prostřednictvím mobilní aplikace a poté, co bude aktivováno zvolené množství, musí jej potvrdit provozovatel podniku ve webové aplikaci. Zákazníkovi je následně poskytnuta sleva na zvolené zboží a kredity budou odečteny.

Výsledná soutěžní platforma nese název „DrinKing“, jelikož je krátký, chytlavý a snadno zapamatovatelný. Implementace a zavedení této platformy by mohlo pomoci probudit zájem zejména u studentů, ke kterým máme v rámci naší sociální bubliny nejbliže.

Následující obrázek představuje grafický popis vztahů mezi subjekty, které jsou výše zmíněny: <sup>1</sup>

---

<sup>1</sup>Obrázek je výsledkem spojení více originálních obrázků ze zdrojů [1] [2] [3] [4] [5] [6] [7]



Obrázek 2.1: Grafický popis funkce platformy

## 2.2 Rozdělení práce

S kolegou Michalem Šalagou [10] jsme se rozhodli rozdělit práci na platformě na dvě části, které budou realizované v oddělených bakalářských pracích. Rozhodnutí vychází ze zkušeností z předešlých semestrálních prací v rámci studia. Jelikož mne baví implementace a návrh clientských aplikací, rozhodl jsem se orientovat moji práci pouze na clientskou stranu bez jakéhokoli zásahu do serverové části. Kolega si naopak dle svých preferencí zvolil práci, orientovanou pouze na serverovou část.

## 2.3 Mobilní a webová aplikace

Z myšlenky 2.1 vyplývá, že dojde k návrhu a implementaci mobilní a webové clientské aplikace. K rozhodnutí došlo kvůli tomu, aby se předešlo zbytečné komplexitě mobilní aplikace, jelikož je třeba pracovat s více rolemi uživatelů, přičemž každá z nich požaduje rozdílnou funkcionalitu.

Mobilní aplikace bude zákazníkům podniků poskytovat přehled o událostech, kterých se budou moci v době konání zúčastnit. Umožní jim do těchto událostí fotografovat účtenky, či zobrazovat historii již nahraných. Dále budou moci zákazníci aktivovat vlastněné kredity a spravovat své uživatelské účty.

Oproti tomu je webová aplikace zamýšlena jako prostředí, které bude k dispozici pouze pro určité role uživatelů - správce platformy a provozovatele podniků. Správce platformy bude smět provádět jakékoli akce - potvrzování transakcí (aktivovaných odměn), vytváření účtů podniků, správu uživatelů,

či vytváření událostí. Narozdíl od správce budou moci provozovatelé podniků využít pouze části funkcionality - potvrzování transakcí a úpravu podnikového profilu.

## ■ 2.4 Aktuální situace

Ačkoli je zřejmé, že virus Covid-19 bude nadále přetrvávajícím problémem dalších pár let, společnost si postupně zvyká na jeho přítomnost a snaží se o návrat do stavu podobném předpandemickému. Jelikož se postupně k tomuto stavu přibližujeme, došlo i k úpravě naší původní myšlenky.

Původně jsme se zabývali nízkou návštěvností podniků kvůli nízké motivaci zákazníků. Náš předpoklad, že si společnost zvykne na servis, se v současné době prokázal jako mylný. Ihned po zrušení nařízení, regulujících a omezujících návštěvnost podniků došlo k velice rychlému návratu zákazníků do nich.

Přestože došlo ke změně situace, není nutné námi vytvořený koncept měnit, jelikož byl od začátku navržen tak, aby obecně motivoval zákazníky k osobní návštěvě podniků. Univerzálnost našeho řešení umožňuje vytvořit jednotný soutěžně věrnostní program napříč všemi spolupracujícími podniky, ve kterém se budou uživatelé snadno orientovat, pokud jej již využívají. Ze strany provozovatelů podniků umožní soutěžní charakter platformy dočasně zvýšit návštěvnost v období konání události.





## Kapitola 3

### Existující řešení

V následující kapitole uvádím existující řešení, která úzce souvisí s námi řešenou problematikou 1.1 a snaží se ji řešit.

Přestože je na trhu velké množství aplikací, které jsou orientované na podporu gastronomie a fungují na principu věrnostního programu či soutěže, neexistuje přímá alternativa k našemu možnému řešení. Pro nalezení řešení, podobných našemu, jsem využil žebříčky nejoblíbenějších aplikací na platformách App Store a Google Play. Dále jsem vycházel z informací, poskytnutých třetími osobami, či recenzí aplikací na internetu. Na základě průzkumu jsem určil dvě možné platformy, které mají k našemu řešení nejbližší - Trifft a Beersport. Tyto dvě platformy jsem uvedl, jelikož je jejich hlavním cílem motivace zákazníků k osobním návštěvám gastronomických podniků.<sup>1</sup>

#### 3.1 Beersport

Beersport je platforma dostupná na zařízeních s operačním systémem iOS a Android, která funguje na principu užití kupónů na konkrétní piva v podnicích, nabízených od spolupracujících pivovarů. Zmíněné kupóny je možné získat finančním převodem, získáním určitého počtu kreditů v rámci soutěží, či zcela zdarma.

Pokud uživatel vlastní kupón na pivo, může navštívit vybraný podnik, který nápoj nabízí. V podniku si následně nápoj objedná a ve chvíli, když má nápoj na stole, smí jej naskenovat prostřednictvím mobilní aplikace. Aplikace fotografii vyhodnotí a pokud vše proběhne bez problému, bude mu na jeho účet zaslána částka v hodnotě objednaného nápoje. Celý proces je dovršen zaplacením zákazníka u pokladny.

Celý koncept platformy je velmi zajímavý z hlediska minimálního zapojení personálu do celého procesu. Vzhled mobilní aplikace je uživatelsky přívětivý a po zkušební dobu jsem nenarazil na žádný problém, který by mi znesnadnil její užití. Jako jediný nedostatek tohoto konceptu jsem vyhodnotil právě omezení pouze na pivní sortiment.

Výhodou Beersport je fakt, že se jedná o dlouhodobě fungující projekt,

---

<sup>1</sup>Kolega Michal Šalaga ve své práci [10] zmiňuje také platformu Wolt, nicméně já na řešenou problematiku uvádím jiný úhel pohledu.

který má pouze na platformě Google Play přes 100 000 stažení. Dále má velmi propracované uživatelské rozhraní a obsahuje funkční gamifikační prvky.

Na druhou stranu je nevýhodou, že narozdíl od našeho řešení neposkytuje možnost získání slev na jiné produkty než piva.<sup>2</sup>



Obrázek 3.1: Ukázka uživatelského rozhraní mobilní aplikace Beersport

## 3.2 Trifft

Trifft je platformou, která využívá kupónového systému a je k dispozici na platformě iOS i Android. Uživatelé aplikace mohou ve spolupracujících podnicích po konzumaci skenovat QR kódy na účtenkách a získávat kredity po každé návštěvě. Pokud uživatel získá dostatek kreditů, smí je uplatnit za vybraný kupón a následně jej v podniku uplatnit. V aplikaci existuje žebříček zákazníků, kteří zde utratili nejvíce peněz. V tomto případě jsem objevil velké množství nedostatků jak po technickém zpracování mobilní aplikace, tak i po konceptuální stránce.

Mobilní aplikace již nebyla po dobu jednoho roku aktualizována a na zkušebním zařízení s operačním systémem iOS byla téměř nepoužitelná, jelikož dochází k častým pádům aplikace. V určitých situacích nebylo možné

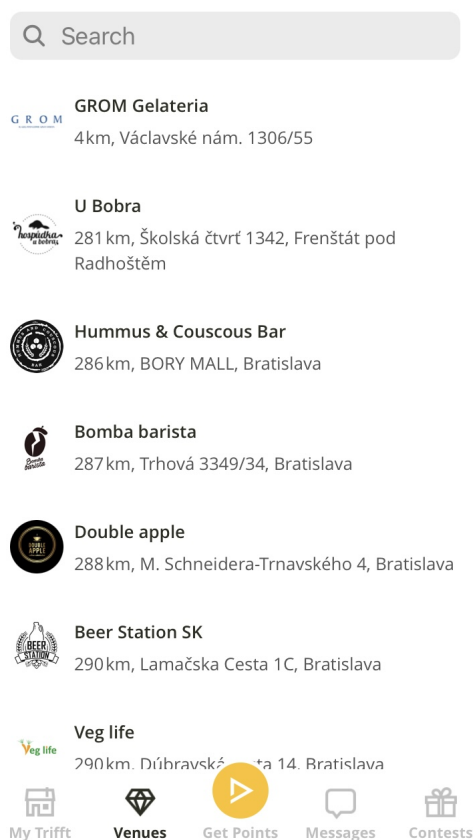
<sup>2</sup>Snímek obrazovky byl pořízen autorem práce.

vyplnit formulářová pole, jelikož se rozložení aplikace nepřizpůsobí situaci, kdy je zobrazena klávesnice. Dalším technickým nedostatkem je přítomnost volby registrace pomocí platformy Facebook, i když služba není již pro tuto aplikaci dostupná.

Po konceptuální stránce je velkým nedostatkem platformy vysoká cena jednotlivých kupónů a nízká variabilita nabízeného zboží. Existence zmíněného žebříčku zákazníků je také sporná, jelikož účastníci nejsou nijak odměněni.

U Trifft jsem nenalezl mnoho výhod. Hlavní výhodou je pouze množství aktivních spoluprací s podniky (především na území Slovenska) a fakt, že se orientuje na jakýkoli sortiment.

Oproti tomu má Trifft značné nevýhody. Mobilní aplikace Trifft není udržovaná a pravidelně aktualizovaná. To má za následek nemožnost přihlášení pomocí platform Google, či Facebook a také špatnou optimalizaci na novějších mobilních zařízeních.<sup>3</sup>



Obrázek 3.2: Ukázka uživatelského rozhraní mobilní aplikace Trifft

### 3.3 Vyhodnocení

Přestože má naše řešení společné znaky s výše uvedenými platformami, je dle mého názoru dostatečně odlišné, aby mělo smysl jej implementovat. Hlavním

<sup>3</sup>Snímek obrazovky byl pořízen autorem práce.

rozdílem je orientace na gamifikační prvek (soutěžení mezi zákazníky), přičemž výše uvedená řešení tento prvek přidávají pouze jako doplněk. Z tohoto důvodu lze chápat naši platformu jako originální řešení uvedené problematiky 1.1, které má smysl vyzkoušet.

Při realizaci našeho řešení bude nutné vyvarovat se obdobných technických nedostatků jako má platforma Triffit. Mezi tyto nedostatky bych zařadil především špatnou optimalizaci pro novější zařízení a absenci průběžných aktualizací.

Mezi pozitivní poznatky, které poskytla řešerše existujících řešení bych oproti tomu zařadil vzhled uživatelského rozhraní Beersport. V rámci své práce bych proto chtěl navrhnout obdobně přehledné rozhraní jako má právě Beersport.

## Kapitola 4

### Analýza požadavků

V této části bakalářské práce se zabývám detailnější analýzou požadavků soutěžní platformy. Nejprve analyzuji všechny funkční a nefunkční požadavky, které vycházejí z výše uvedené myšlenky projektu a následně uvedu možné případy užití, které navazují na ně. Uvedené požadavky a případy užití budou také zohledňovat pozitivní a negativní poznatky, neulezené při řešení existujících řešení 3.

#### 4.1 Analýza funkčních a nefunkčních požadavků

Analýza požadavků je důležitou součástí každého projektu, jelikož si v rámci ní definujeme všechny stěžejní požadavky, které mají být součástí výsledného produktu [11]. Požadavky dělíme na:

- **Funkční požadavky** - Takové požadavky, které definují jak se bude chovat systém.
- **Nefunkční požadavky** - Takové požadavky, které definují jaké bude mít systém vlastnosti.

V následující části definuji funkční a nefunkční požadavky jak pro mobilní, tak i webovou aplikaci.

##### 4.1.1 Mobilní aplikace

##### Funkční požadavky mobilní aplikace

- **FR01 – Registrace nového uživatele**  
Systém umožní uživateli registrovat se a vytvořit nový účet.
- **FR02 – Přihlášení uživatele**  
Systém umožní uživateli přihlásit se ke svému účtu.
- **FR03 – Odhlášení uživatele**  
Systém umožní uživateli odhlásit se.

- **FR04 – Změna hesla uživatele**  
Systém umožní přihlášenému uživateli změnit si své heslo.
- **FR05 – Smazání účtu uživatele**  
Systém umožní uživateli smazat svůj účet.
- **FR06 – Zobrazení profilu a statistik uživatele**  
Systém umožní zobrazení detailu a statistik spojených s účtem přihlášeného uživatele.
- **FR07 – Přidání profilové fotografie**  
Systém umožní uživateli přidání profilové fotografie.
- **FR08 – Změna profilové fotografie**  
Systém umožní uživateli změnu existující profilové fotografie.
- **FR09 – Odstranění profilové fotografie**  
Systém umožní uživateli odstranit existující profilovou fotografii.
- **FR10 – Zobrazení událostí**  
Systém umožní uživateli zobrazení seznamu všech nastávajících či probíhajících událostí.
- **FR11 – Filtrování událostí**  
Systém umožní uživateli filtrovat položky ze seznamu všech nastávajících či probíhajících událostí podle vybraného kritéria.
- **FR12 – Zobrazení všech vyfotografovaných účtenek**  
Systém umožní uživateli zobrazení seznamu všech vyfotografovaných účtenek, vázajících se na určitou událost.
- **FR13 – Přidání nové fotografie účtenky**  
Systém umožní uživateli přidání nové fotografie účtenky.
- **FR14 – Zobrazení detailu probíhající či nadcházející události**  
Systém umožní uživateli zobrazení detailu probíhající či nadcházející události.
- **FR15 – Zobrazení všech aktuálně dostupných odměn**  
Systém umožní uživateli zobrazení seznamu všech aktuálně dostupných odměn.
- **FR16 – Aktivace dostupné odměny**  
Systém umožní uživateli zvolit si a aktivovat určitý počet dostupných odměn/kreditů.
- **FR17 – Zobrazení detailu transakce**  
Systém umožní uživateli zobrazení detailu aktivní transakce.
- **FR18 – Možnost skrytí transakce**  
Systém umožní uživateli skrýt detail probíhající transakce.

- **FR19 – Možnost otevření detailu transakce**  
Systém umožní uživateli zobrazení detailu probíhající transakce.

- **FR20 – Ukončení transakce**  
Systém umožní uživateli ukončit aktivní transakci.

#### ■ **Nefunkční požadavky mobilní aplikace**

- **NFR01 – Responsivní design**  
Systém bude mít responzivní design, aby bylo docíleno přehledného zobrazení všech částí aplikace při odlišných rozměrech obrazovky.
- **NFR02 – Konzistentní UI**  
Systém bude obsahovat konzistentní UI, které bude v uživateli evokovat dojem, že se stále nachází ve stejné aplikaci.
- **NFR03 – Škálovatelnost**  
Zdrojový kód systému bude psán s ohledem na možné budoucí rozšíření funkcionality.
- **NFR04 – Bezpečnost**  
Systém bude bezpečně uchovávat nejcitlivější uživatelská data.
- **NFR05 – Kompatibilita s platformami iOS a Android**  
Systém bude plně kompatibilní s platformami iOS a Android.
- **NFR06 – Využití real-time komunikace**  
Systém bude částečně využívat technologie real-time komunikace, aby bylo dosaženo okamžité propagace odpovědi na klientské straně.
- **NFR07 – Vyhotovení aplikace v anglickém jazyce**  
Systém bude kompletně lokalizován do anglického jazyka.

#### ■ **4.1.2 Webová aplikace**

##### ■ **Funkční požadavky webové aplikace**

- **FR01 – Přihlášení uživatele**  
Systém umožní přihlášení uživatele k jeho účtu.
- **FR02 – Odhlášení uživatele**  
Systém umožní uživateli odhlásit se.
- **FR03 – Zobrazení dostupných podniků**  
Systém umožní uživateli zobrazení seznamu všech podniků, ke kterým má jako přihlášený uživatel oprávnění.
- **FR04 – Zobrazení detailu podniku**  
Systém umožní uživateli zobrazení detailu konkrétního podniku.

- **FR05 – Přidání fotografie podniku**  
Systém umožní uživateli přidání profilové fotografie podniku.
- **FR06 – Smazání fotografie podniku**  
Systém umožní uživateli odstranění profilové fotografie podniku.
- **FR07 – Zobrazení aktivních transakcí**  
Systém umožní uživateli zobrazení seznamu všech aktivních transakcí v podniku, ke kterým má oprávnění.
- **FR08 – Potvrzení aktivní transakce**  
Systém umožní uživateli potvrzení aktivní transakce.
- **FR09 – Odmítnutí aktivní transakce**  
Systém umožní uživateli odmítnutí aktivní transakce.
- **FR10 – Vytvoření nového účtu provozovatele a podniku**  
Systém umožní uživateli vytvořit nový účet provozovatele podniku společně s účtem podniku.
- **FR11 – Zobrazení seznamu událostí**  
Systém umožní uživateli zobrazit si seznam všech událostí.
- **FR12 – Filtrování událostí**  
Systém umožní uživateli filtrovat seznamu všech událostí podle zvoleného kritéria.
- **FR13 – Vytvoření nové události**  
Systém umožní uživateli vytvoření nové události v rámci zvoleného podniku.
- **FR14 – Odstranění události**  
Systém umožní uživateli odstranění již existující události.
- **FR15 – Ukončení události**  
Systém umožní uživateli ukončení události.
- **FR16 – Zobrazení účtenek**  
Systém umožní uživateli zobrazení seznamu všech účtenek.
- **FR17 – Filtrování účtenek**  
Systém umožní uživateli filtrovat seznamu všech účtenek podle zvoleného kritéria.
- **FR18 – Potvrzení účtenky**  
Systém umožní uživateli potvrzení účtenky.
- **FR19 – Odmítnutí účtenky**  
Systém umožní uživateli odmítnutí účtenky.
- **FR20 – Zobrazení seznamu všech uživatelů**  
Systém umožní uživateli zobrazení seznamu se všemi uživateli.



- **FR21 – Smazání účtu uživatele**  
Systém umožní uživateli smazání účtu zvoleného uživatele.
- **FR22 – Vyřazení účastníka z události**  
Systém umožní uživateli vyřazení zvoleného účastníka z probíhající události.
- **FR23 – Zablokování účtu účastníka**  
Systém umožní uživateli zablokovat účet jiného účastníka.
- **FR24 – Zrušení blokování účtu účastníka**  
Systém umožní uživateli zrušit blokování účastníka.

#### ■ Nefunkční požadavky webové aplikace

- **NFR01 – Responsivní design**  
Systém bude mít responzivní design, aby bylo docíleno přehledného zobrazení všech částí aplikace při odlišných rozměrech obrazovky.
- **NFR02 – Konzistentní UI**  
Systém bude obsahovat konzistentní UI, které bude v uživateli evokovat dojem, že se stále nachází ve stejné aplikaci.
- **NFR03 – Škálovatelnost**  
Zdrojový kód systému bude psán s ohledem na možné budoucí rozšíření funkcionality.
- **NFR04 – Bezpečnost**  
Systém bude bezpečně uchovávat nejcitlivější uživatelská data.
- **NFR05 – Kompatibilita s webovými prohlížeči s jádrem Chromium**  
Systém bude plně kompatibilní s webovými prohlížeči postavenými na jádře Chromium.
- **NFR06 – Využití real-time komunikace**  
Systém bude částečně využívat technologie real-time komunikace, aby bylo dosaženo okamžité propagace odpovědi na klientské straně.
- **NFR07 – Vyhotovení aplikace v anglickém jazyce**  
Systém bude kompletně lokalizován do anglického jazyka.

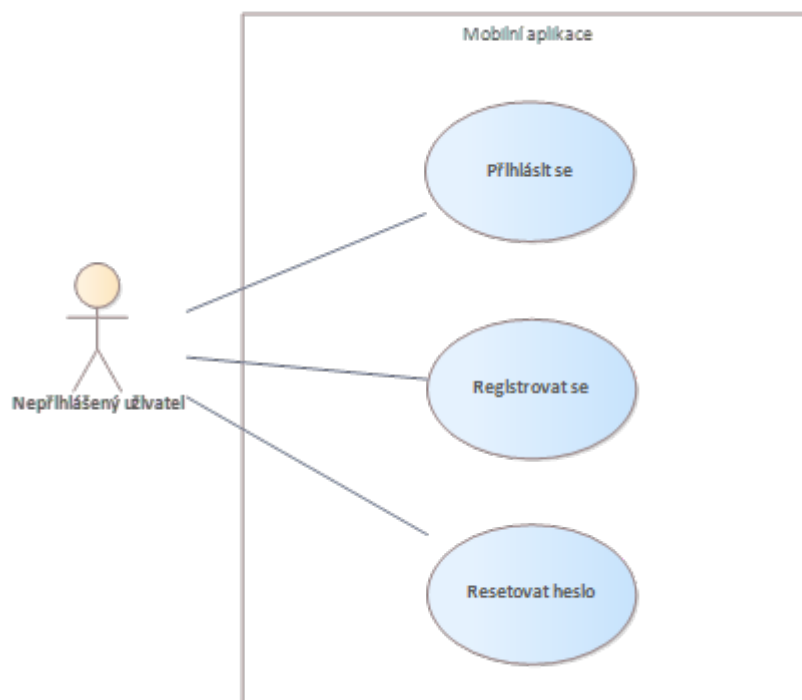
## ■ 4.2 Diagramy případů užití

Diagramy případů užití představují grafické zobrazení všech možných interakcí uživatele se systémem, přičemž vycházejí z již definovaných funkčních požadavků. Mezi základní části diagramu patří prvky aktér a případ užití [12].

### 4.2.1 Mobilní aplikace

#### ■ Nepřihlášený uživatel

Následující obrázek představuje grafický popis všech případů užití pro nepřihlášeného uživatele mobilní aplikace: 4.1



**Obrázek 4.1:** Diagram případů užití nepřihlášeného uživatele mobilní aplikace

#### ■ UC01 - Přihlásit se [Nepřihlášený uživatel]

Uživatel vyplní pole heslo a přihlašovací jméno ve formuláři a stiskne tlačítko „Přihlásit“. Systém jej poté přesměruje na úvodní stránku.

#### ■ UC02 - Registrovat se [Nepřihlášený uživatel]

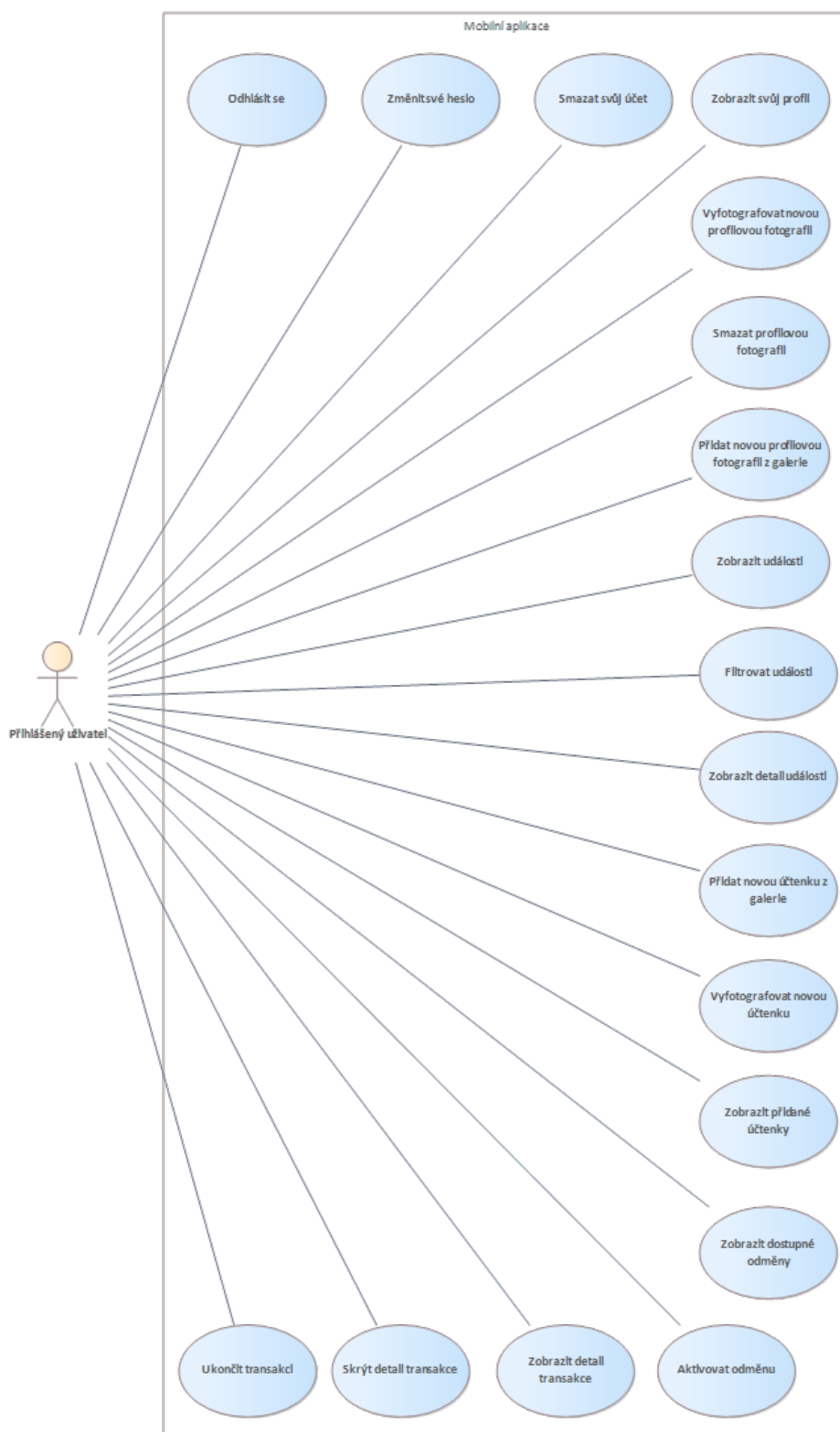
Uživatel vyplní pole jméno, email, heslo a potvrzení hesla ve formuláři. Následně stiskne tlačítko „Registrovat“ a systém jej poté přesměruje na úvodní stránku.

#### ■ UC03 - Resetovat heslo [Nepřihlášený uživatel]

Uživatel stiskne tlačítko „Zapomněl jsem heslo“. Systém zobrazí formulář pro vyplnění emailové adresy a uživatel pole vyplní a stiskne tlačítko „Odeslat“.

#### ■ Přihlášený uživatel

Následující obrázek představuje grafický popis všech případů užití pro přihlášeného uživatele mobilní aplikace: 4.2



Obrázek 4.2: Diagram případů užití přihlášeného uživatele mobilní aplikace

- **UC04 - Odhlásit se [Přihlášený uživatel]**  
Uživatel stiskne tlačítko „Odhlásit se“. Systém jej následně přesměruje na přihlašovací stránku.
- **UC05 - Změnit své heslo [Přihlášený uživatel]**  
Uživatel stiskne tlačítko „Změnit heslo“. Systém zobrazí formulář pro změnu hesla. Uživatel vyplní pole aktuální heslo, nové heslo, potvrzení nového hesla a poté stiskne tlačítko „Uložit“. Systém jej následně přesměruje na přihlašovací stránku.
- **UC06 - Smazat svůj účet [Přihlášený uživatel]**  
Uživatel stiskne tlačítko „Smazat účet“ a potvrdí akci stisknutím tlačítka „Smazat účet“. Systém jej následně přesměruje na přihlašovací stránku.
- **UC07 - Zobrazit svůj profil [Přihlášený uživatel]**  
Uživatel stiskne tlačítko „Profil“ na spodní navigační liště. Systém jej přesměruje na profilovou stránku, která obsahuje statistiky a informace o uživateli.
- **UC08 - Vyfotografovat novou profilovou fotografií [Přihlášený uživatel]**  
Uživatel stiskne tlačítko s ikonou ve tvaru tužky. Systém zobrazí modální okno s možnostmi „Přidat profilovou fotografii z galerie“, „Vyfotografovat novou profilovou fotografii“ a „Smazat profilovou fotografii“. Uživatel stiskne tlačítko „Vyfotografovat novou profilovou fotografii“ a systém otevře okno s rozhraním fotoaparátu. Uživatel stiskne tlačítko spouště a systém jej opět přesměruje na profilovou stránku.
- **UC09 - Vybrat novou profilovou fotografií z galerie [Přihlášený uživatel]**  
Uživatel stiskne tlačítko s ikonou ve tvaru tužky. Systém zobrazí modální okno s možnostmi „Přidat profilovou fotografii z galerie“, „Vyfotografovat novou profilovou fotografii“ a „Smazat profilovou fotografii“. Uživatel stiskne tlačítko „Přidat profilovou fotografii z galerie“ a systém otevře okno s obrázky obsaženými v úložišti zařízení. Uživatel stiskne vybraný obrázek a systém jej opět přesměruje na profilovou stránku.
- **UC10 - Smazat profilovou fotografií [Přihlášený uživatel]**  
Uživatel stiskne tlačítko s ikonou ve tvaru tužky. Systém zobrazí modální okno s možnostmi „Přidat profilovou fotografii z galerie“, „Vyfotografovat novou profilovou fotografii“ a „Smazat profilovou fotografii“. Uživatel stiskne tlačítko „Smazat profilovou fotografii“ a systém následně zavře modální okno s možnostmi.
- **UC11 - Zobrazit události [Přihlášený uživatel]**  
Uživatel stiskne tlačítko „Události“ umístěné ve spodní navigační liště. Systém jej přesměruje na stránku s událostmi, zobrazí seznam událostí a tlačítka pro filtrování událostí.

- **UC12 - Filtrovat události [Přihlášený uživatel]**

Uživatel stiskne tlačítko s názvem „Nadcházející“ nebo „Probíhající“ v horní filtrovací liště. Systém poté zobrazí seznam událostí, které se nachází ve zvoleném stavu.
- **UC13 - Zobrazit detail události [Přihlášený uživatel]**

Uživatel stiskne obrázek události na stránce s událostmi. Systém uživatele přesměruje na stránku s detaily události osahující tlačítka s možnostmi „Přidat účtenku“, „Historie“. Stránka také obsahuje tabulku s účastníky a tabulku s odměnami.
- **UC14 - Přidat novou účtenku z galerie [Přihlášený uživatel]**

Uživatel stiskne tlačítko „Přidat účtenku“. Systém otevře modální okno s možnostmi „Přidat fotografii z galerie“ a „Vyfotografovat novou fotografii“. Uživatel stiskne tlačítko „Přidat fotografii z galerie“ a systém otevře okno s obrázky obsaženými v úložišti zařízení. Uživatel stiskne vybraný obrázek a systém jej opět přesměruje na stránku s detailem události.
- **UC15 - Vyfotografovat novou účtenku [Přihlášený uživatel]**

Uživatel stiskne tlačítko „Přidat účtenku“. Systém otevře modální okno s možnostmi „Přidat fotografii z galerie“ a „Vyfotografovat novou fotografii“. Uživatel stiskne tlačítko „Vyfotografovat novou fotografii“ a systém otevře okno s rozhraním fotoaparátu. Uživatel stiskne tlačítko spouště a systém jej opět přesměruje na stránku s detailem události.
- **UC16 - Zobrazit přidané účtenky [Přihlášený uživatel]**

Uživatel stiskne tlačítko „Historie“. Systém otevře stránku obsahující seznam všech uživatelem přidanými fotografiemi účtenek.
- **UC17 - Zobrazit dostupné odměny [Přihlášený uživatel]**

Uživatel stiskne tlačítko „Odměny“. Systém otevře stránku obsahující seznam všech uživatelem získaných odměn.
- **UC18 - Aktivovat odměnu [Přihlášený uživatel]**

Uživatel stiskne obrázek odměny a systém následně zobrazí modální okno obsahující informace o odměně a formulář, obsahující pole s textem „Zadejte počet kreditů“. Uživatel vyplní pole, stiskne tlačítko „Aktivovat odměnu“ a systém uživatele přesměruje na stránku s probíhající transakcí.
- **UC19 - Skrýt detail transakce [Přihlášený uživatel]**

Uživatel stiskne tlačítko pro zavření obrazovky a systém jej přesměruje na předchozí stránku, na které se nacházel. Systém poté zobrazí notifikaci o aktivní transakci v liště nacházející se nad navigační lištou.
- **UC20 - Zobrazit detail transakce [Přihlášený uživatel]**

Uživatel stiskne tlačítko uvnitř notifikační lišty s aktivní transakcí. Systém uživatele poté přesměruje na stránku s probíhající transakcí.

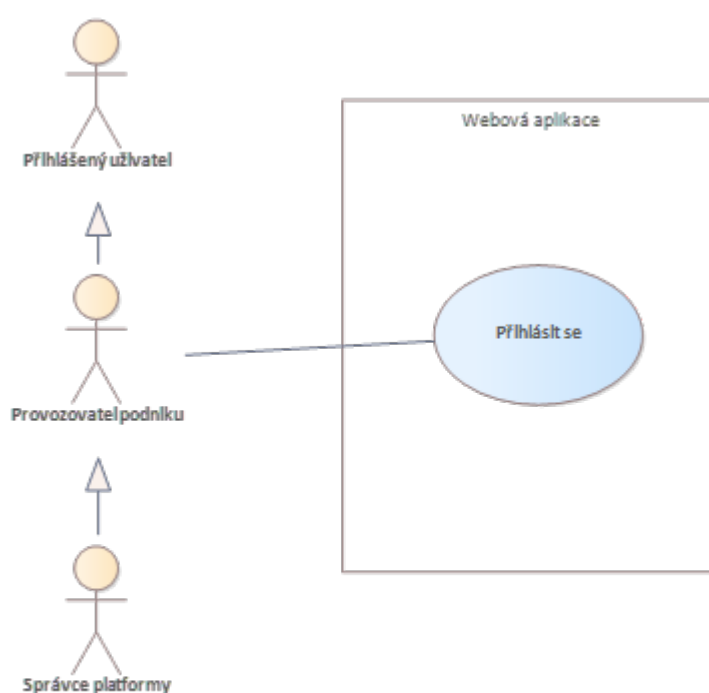
■ **UC21 - Ukončit transakci [Přihlášený uživatel]**

Uživatel stiskne tlačítko „Ukončit transakci“ a systém jej následně přesměruje na stránku, na které se předtím nacházel a skryje notifikační lištu s aktivní transakcí.

■ **4.2.2 Webová aplikace**

■ **Nepřihlášený uživatel**

Následující obrázek představuje grafický popis všech případů užití pro nepřihlášeného uživatele webové aplikace: 4.3



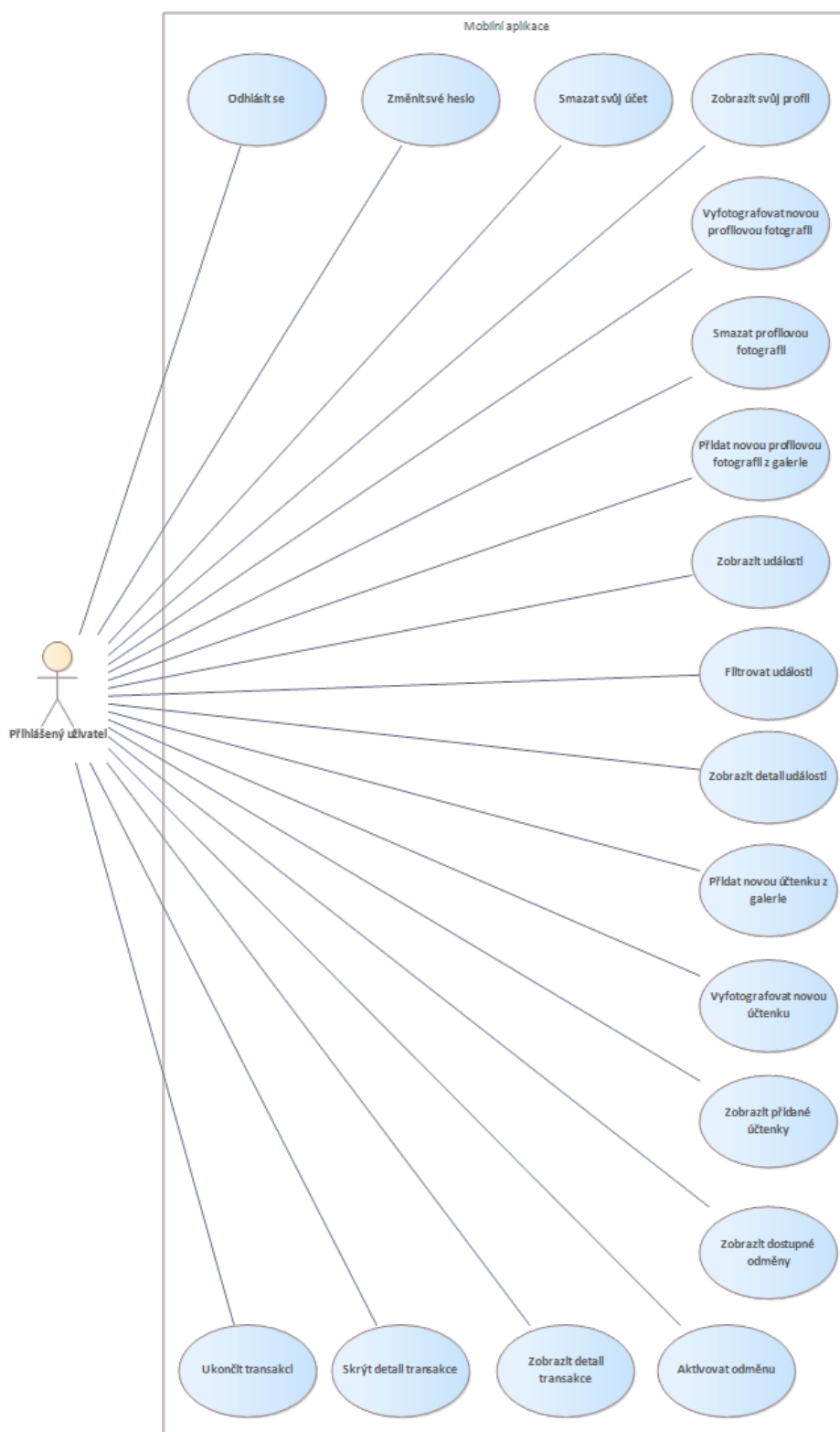
**Obrázek 4.3:** Diagram případů užití nepřihlášeného uživatele webové aplikace

■ **UC01 - Přihlásit se [Nepřihlášený uživatel]**

Uživatel vyplní pole heslo a přihlašovací jméno ve formuláři a stiskne tlačítko „Přihlásit“. Systém jej poté přesměruje na úvodní stránku.

■ **Přihlášený uživatel**

Následující obrázek představuje grafický popis všech případů užití pro přihlášeného uživatele webové aplikace: 4.4



Obrázek 4.4: Diagram případů užití přihlášeného uživatele webové aplikace

- **UC02 - Odhlásit se [Přihlášený uživatel]**  
Uživatel stiskne tlačítko „Odhlásit“ a systém jej poté přesměruje na přihlašovací stránku.
- **UC03 - Zobrazit dostupné podniky [Provozovatel podniku]**  
Uživatel stiskne tlačítko „Podniky“ v navigační liště a systém ho přesměruje na stránku se seznamem všech dostupných podniků.
- **UC04 - Zobrazit detail podniku [Provozovatel podniku]**  
Uživatel stiskne obrázek podniku a systém jej přesměruje na stránku s detaily vybraného podniku.
- **UC05 - Přidat fotografii podniku [Provozovatel podniku]**  
Uživatel stiskne tlačítko s ikonou ve tvaru tužky a systém poté zobrazí modální okno s formulářem pro výběr souboru z úložiště zařízení a možnostmi „Přidat“ a „Smazat aktuální“. Uživatel vybere soubor a stiskne tlačítko „Přidat“. Systém poté zavře modální okno.
- **UC06 - Smazat fotografii podniku [Provozovatel podniku]**  
Uživatel stiskne tlačítko s ikonou ve tvaru tužky a systém poté zobrazí modální okno s formulářem pro výběr souboru z úložiště zařízení a možnostmi „Přidat“ a „Smazat aktuální“. Uživatel stiskne tlačítko „Smazat aktuální“. Systém poté zavře modální okno.
- **UC07 - Zobrazit aktivní transakce [Provozovatel podniku]**  
Uživatel stiskne tlačítko „Transakce“ nacházející se v navigační liště a systém jej přesměruje na stránku s transakcemi.
- **UC08 - Potvrdit aktivní transakci [Provozovatel podniku]**  
Uživatel stiskne tlačítko „Potvrdit“ u vybrané transakce. Systém poté skryje potvrzenou transakci.
- **UC09 - Odmítnout aktivní transakci [Provozovatel podniku]**  
Uživatel stiskne tlačítko „Odmítnout“ u vybrané transakce. Systém poté skryje potvrzenou transakci.
- **UC10 - Vytvořit novou událost [Správce platformy]**  
Uživatel stiskne tlačítko „Vytvořit událost“ a systém poté otevře modální okno s formulářem obsahující pole název, začátek události, konec události a odměny. Uživatel pole vyplní a stiskne tlačítko „Vytvořit“. Systém poté zavře modální okno.
- **UC11 - Zobrazit události [Správce platformy]**  
Uživatel stiskne tlačítko „Události“ nacházející se v navigační liště. Systém jej přesměruje na stránku se seznamem všech událostí a možnostmi filtrování pro nadcházející, probíhající, revidované a proběhlé události.
- **UC12 - Filtrovat události [Správce platformy]**  
Uživatel stiskne vybrané filtrační tlačítko a systém poté zobrazí jen události, které jsou ve vybraném stavu.



- **UC13 - Zobrazit detail události [Správce platformy]**

Uživatel stiskne položku události v seznamu a systém jej přesměruje na stránku s detaily probíhající události. Na stránce se nachází seznam všech čekajících účtenek účastníků. Na stránce se nachází tlačítka pro filtrování účtenek pro čekající a vyřízené účtenky a možnosti „Ukončit událost“ a „Odstranit událost“.
- **UC14 - Filtrovat účtenky [Správce platformy]**

Uživatel stiskne vybrané tlačítko pro filtrování seznamu účtenek. Systém následně zobrazí seznam účtenek, které odpovídají zvolenému kritériu.
- **UC15 - Ukončit událost [Správce platformy]**

Uživatel stiskne tlačítko „Ukončit událost“ a systém poté změní stav položky z revidované na ukončenou.
- **UC16 - Smazat událost [Správce platformy]**

Uživatel stiskne tlačítko „Smazat událost“ a systém poté přesměruje uživatele na stránku s událostmi a událost smaže.
- **UC17 - Odmítnout účtenku [Správce platformy]**

Uživatel stiskne tlačítko „Odmítnout“ u vybrané položky ze seznamu účtenek. Systém poté skryje položku a změní její stav jako „odmítnutá“.
- **UC18 - Potvrdit účtenku [Správce platformy]**

Uživatel vyplní pole identifikátor účtenky, hodnota účtenky a poté stiskne tlačítko „Potvrdit“. Pokud systém již obsahuje účtenku s tímto identifikátorem, zobrazí chybovou hlášku, v opačném případě systém účtenku skryje a změní její stav na „potvrzená“.
- **UC19 - Vyřadit účastníka z události [Správce platformy]**

Uživatel vybere účastníka ze seznamu a stiskne tlačítko s ikonou ve tvaru kříže. Systém zobrazí modální okno pro potvrzení a uživatel stiskne tlačítko „Potvrdit“. Systém poté odstraní účastníka z události.
- **UC20 - Zobrazit všechny uživatele [Správce platformy]**

Uživatel stiskne tlačítko „Uživatelé“ nacházející se v navigační liště. Systém jej následně přesměruje na stránku se seznamem všech uživatelů s možností „Vytvořit účet provozovatele“.
- **UC21 - Vytvořit účet provozovatele a podniku [Správce platformy]**

Uživatel stiskne tlačítko „Vytvořit účet provozovatele“ a systém následně zobrazí modální okno s formulářem obsahujícím pole jméno, telefonní číslo, email, PSČ, číslo budovy, ulice a název podniku. Uživatel pole vyplní a stiskne tlačítko „Vytvořit“. Systém poté zavře modální okno a vytvoří nový účet provozovatele a jeho podniku.
- **UC22 - Smazat účet uživatele [Správce platformy]**

Uživatel vybere účet ze seznamu a stiskne tlačítko „Smazat účet“. Systém

zobrazí modální okno pro potvrzení a uživatel stiskne tlačítko „Potvrdit“. Systém poté odstraní účet.

- **UC23 - Filtrovat uživatele [Správce platformy]**  
Uživatel vyplní filtrovací pole ve vrchní části obrazovky a systém poté zobrazí pouze ty účty, které obsahují zadané znaky.
- **UC24 - Zablokovat uživatele [Správce platformy]**  
Uživatel vybere účet ze seznamu a stiskne tlačítko „Zablokovat účet“. Systém zobrazí modální okno pro potvrzení a uživatel stiskne tlačítko „Potvrdit“. Systém poté zablokuje účet.
- **UC25 - Zrušit blokování uživatele [Správce platformy]**  
Uživatel vybere účet ze seznamu a stiskne tlačítko „Zrušit blokování účtu“. Systém zobrazí modální okno pro potvrzení a uživatel stiskne tlačítko „Potvrdit“. Systém poté zruší blokování účtu.

## Kapitola 5

### Návrh klientské strany platformy

V následující kapitole se zabývám návrhem mobilní a webové aplikace. Nejprve vytvořím procesní diagramy popisující klíčovou funkcionalitu platformy, poté vytvořím prototypy obou aplikací a nakonec zhodnotím jednotlivé technologie, které budou klíčové při implementaci základní funkcionality obou aplikací. Při tvorbě návrhu klientské strany jsem vycházel z požadavků a případů užití, definovaných v předchozí kapitole 4.

#### 5.1 Procesní diagramy

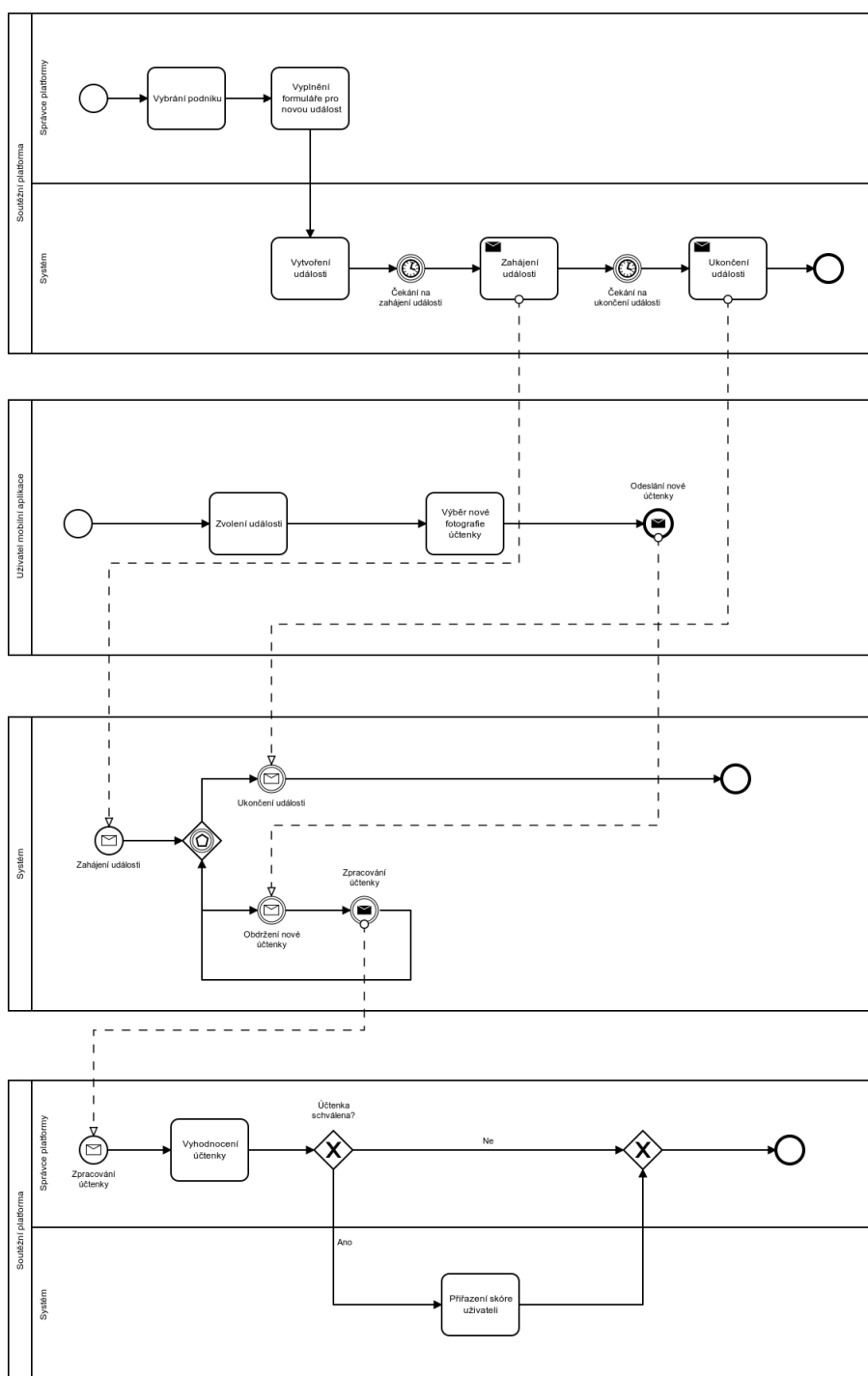
Procesní diagramy představují grafický popis jakéhokoli procesu. Znázorňují potřebné vstupy, výstupy a posloupnost jednotlivých kroků, které je třeba splnit, aby bylo zajištěno správné fungování procesu [13].

V této části se věnuji popisu klíčových procesů platformy. Navržené diagramy budou sloužit jako podklad při tvorbě prototypů či implementaci její základní funkcionality.

Níže uvedené diagramy představují nesložitější části funkcionality platformy. Rozhodl jsem se je zde uvést pro demonstraci komunikace mezi serverovou a klientskou částí a pro příklad jejich závislosti na definovaných případech užití 4.2.

##### 5.1.1 Proces realizace události

Následující proces 5.1 popisuje situaci, kdy je vytvořena nová nadcházející událost v existujícím podniku. Jakmile nastane čas jejího zahájení, je spuštěn proces, který zachytává zaslané účtenky z mobilní aplikace a ty následně musí potvrdit v rámci potvrzovacího procesu správce platformy. Ve chvíli, kdy dojde k ukončení události, není možné již účtenky nahrát, avšak správce je stále musí vyhodnotit.

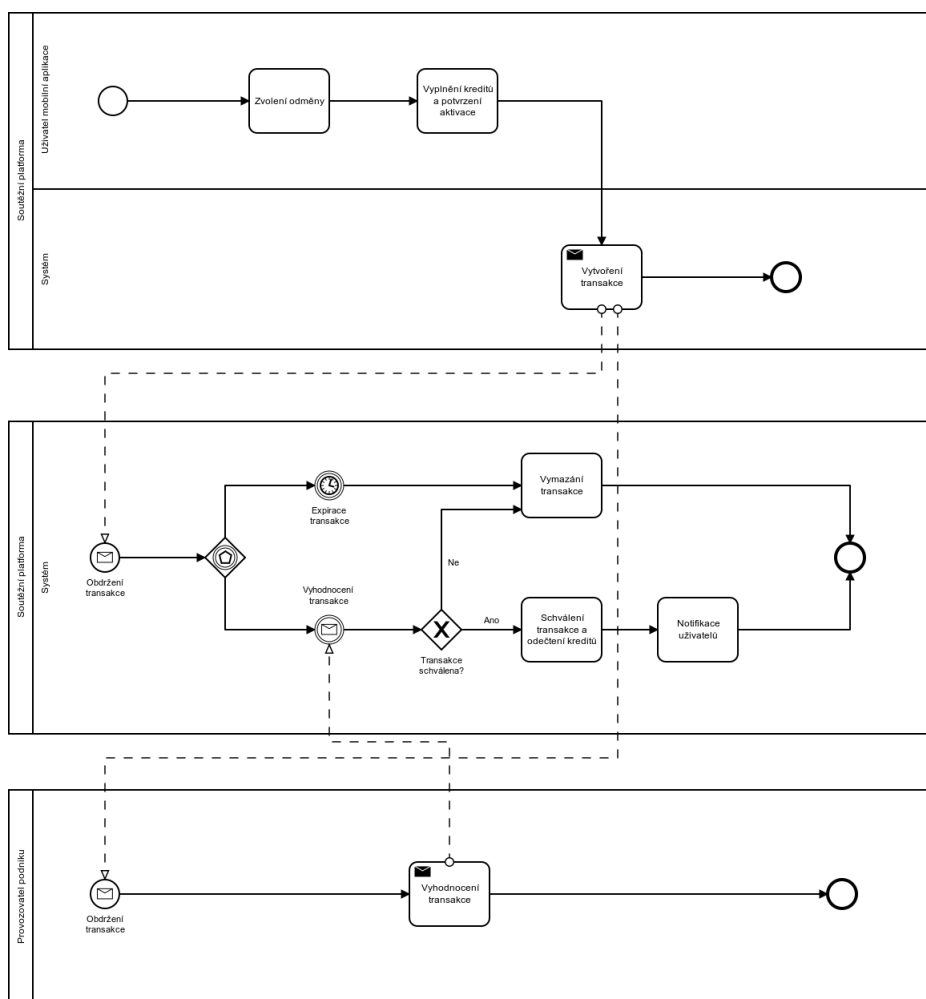


Obrázek 5.1: Diagram procesu realizace události

### 5.1.2 Proces zpracování transakce

Tento proces 5.2 zachycuje situaci, kdy uživatel mobilní aplikace vlastní odměnu v rámci konkrétního podniku. Jakmile ji uživatel aktivuje, spustí se proces zpracování nové transakce. Danou transakci je možné ze strany

provozovatele podniku potvrdit či odmítnout. V případě vypršení časového limitu je transakce automaticky odmítnuta.



Obrázek 5.2: Diagram procesu zpracování transakce

## 5.2 Prototypování klientských aplikací

Prototypování aplikací hraje ve vývoji klientské strany software zásadní roli. Díky prototypování aplikací máme možnost získat zpětnou vazbu od testovacích subjektů, zda jim aplikace vyhovuje z designového hlediska, zda se v ní snadno orientují nebo zda se jim zdá kompletní. Odpověď na každou z těchto otázek je možné získat bez psaní jakéhokoli kódu a díky znalostem odpovědí na tyto otázky lze také ušetřit mnoho času při implementaci její architektury či struktury [16]. Jednotlivé typy prototypů se od sebe liší mírou detailu a dělíme je na:

### ■ Low-fidelity prototypy

Low-fidelity prototypy zachycují souslednost jednotlivých stránek apli-

kace (dále zmíněno jako *flow aplikace*). Prototypy také definují základní rozložení klíčových prvků na obrazovce.

### ■ High-fidelity prototypy

High-fidelity prototypy zachycují detailní návrh aplikace s designovými prvky jako jsou například animace. Rovněž výrazně usnadňují stylování klientské strany aplikace, jelikož definují přesné odsazení, velikosti a tvary jednotlivých prvků na obrazovce.

V rámci své práce jsem nejprve vytvořil low-fidelity prototypy mobilní A.1 a webové aplikace A.2, které jsem testoval s pomocí malé skupiny o počtu 6ti subjektů. Low-fidelity prototypy jsem vytvořil za účelem navržení flow aplikací. Následně jsem vytvořil high-fidelity prototypy, abych si mohl lépe rozvrhnout, jaké komponenty (a jejich design) bude nutné v rámci aplikací implementovat a využil jsem je k demonstraci platformy provozovatelům podniků B.1 B.2.

## ■ 5.2.1 Použitý software pro tvorbu prototypů

Na trhu je dispozici mnoho nástrojů pro prototypování aplikací, mezi nejoblíbenější z nich patří Figma, Adobe XD, InVisionStudio či Webflow. V rámci své práce jsem se proto rozhodl použít a porovnat dva z těchto nástrojů - jeden pro low-fidelity a druhý pro high-fidelity prototypy klientských aplikací.

### ■ Adobe XD

Jak již název napovídá, Adobe XD je nástroj vytvořený společností Adobe a ve své práci jsem jej využil pro tvorbu low-fidelity prototypů. V základní bezplatné verzi je možné pomocí Adobe XD vytvářet i zobrazovat prototypy, avšak možnosti pro sdílení či skupinovou úpravu prototypů jsou značně omezené. Uživatel má k dispozici vytvořit pouze jeden skupinový dokument a možnost vytvoření sdíleného odkazu pro svůj prototyp [15].

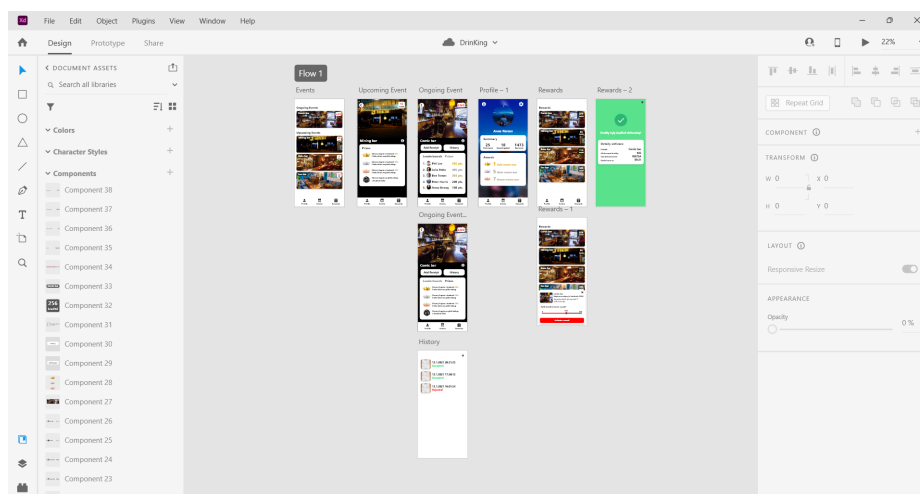
Adobe nabízí kromě desktopové verze i mobilní aplikaci, ve které je možné zobrazit celý prototyp a následně s ním interagovat. V mobilní aplikaci ale není možné upravovat jednotlivé prvky prototypu, jelikož pro editaci slouží pouze desktopová aplikace.

V desktopové aplikaci je možné shlukovat manuálně vytvořené prvky/tvary do komponent, které následně lze hromadně přemísťovat po obrazovce. Dále je možné vytvářet jednotlivé stránky aplikace, definovat interakce s prvky, či možnost přidání komunitně vytvořených pluginů, obsahujících například ikony.

Obrázek níže zachycuje uživatelské rozhraní aplikace Adobe XD: 5.3 <sup>1</sup>

---

<sup>1</sup>Snímek obrazovky byl pořízen autorem práce.

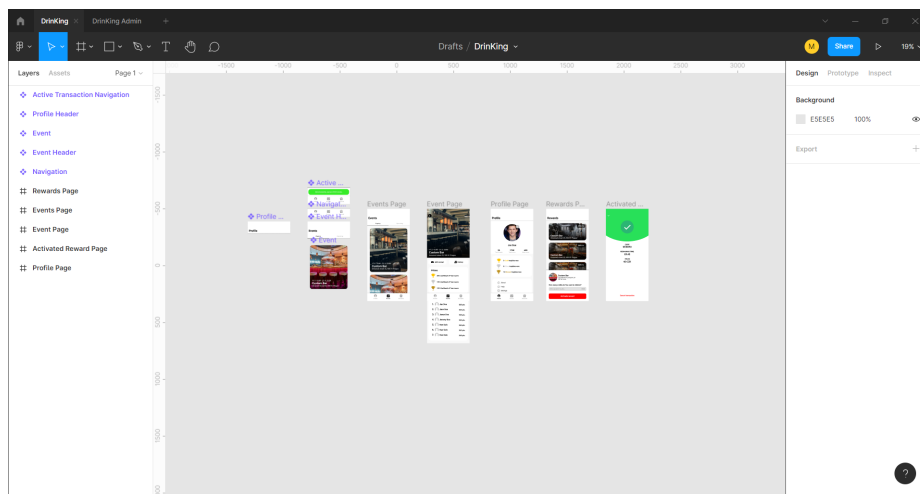


Obrázek 5.3: Screenshot aplikace Adobe XD

## ■ Figma

Figu jsem využil pro tvorbu high-fidelity prototypů. Stejně jako Adobe XD poskytuje mobilní i desktopovou aplikaci, ale lze editovat prototypy i v rámci webového prohlížeče, jinak se v rámci své základní funkcionality nijak zásadně neliší. Narozdíl od Adobe XD není nijak omezena skupinová kolaborace pro jednotlivé projekty v bezplatné verzi [14].

Obrázek níže zachycuje uživatelské rozhraní aplikace Figma: 5.4 <sup>2</sup>



Obrázek 5.4: Screenshot aplikace Figma

## ■ Závěr

Pokud není v praxi nutné kolaborovat s více lidmi na stejném projektu při tvorbě prototypů, jsou oba nástroje na velice podobné úrovni a v principu

<sup>2</sup>Snímek obrazovky byl pořízen autorem práce.

nezáleží, jaký z nich nakonec zvolíme. Z tohoto důvodu jsem zvolil obě uvedené technologie - Adobe XD pro low-fidelity prototypy a Figma pro high-fidelity prototypy.

## ■ 5.2.2 Testování low-fidelity prototypů

V následující části se budu věnovat popisu a výstupu testování prototypů. Pro ověření navržených flow obou aplikací jsem využil low-fidelity prototypy a představil jsem je postupně skupině o počtu 6ti subjektů z řad studentů, které považuji za hlavní cílovou skupinu soutěžní platformy. Pro demonstraci jsem využil mobilní verzi aplikace Adobe XD.

### ■ Průběh testování

Nejprve jsem každému subjektu popsal celkový smysl platformy a poté jsem postupně každému ze subjektů dal úkol. V rámci zadaného úkolu musel každý subjekt bez cizí pomoci provést určitou aktivitu. Příklad této aktivity bylo například aktivování dostupné odměny nebo zobrazení historie přidanych účtenek. Během testování jsem si psal poznámky a poté co subjekt dokončil aktivitu, ptal jsem se na otázky, související s flow aplikace, a na subjektem odhalené nedostatky.

### ■ Výstup testování

Na základě testování byly zjištěny v low-fidelity prototypch následující nedostatky:

- **Nedostatek:** Princip aktivace odměny nebyl dostatečně intuitivní. Bylo nutné upravit flow mobilní aplikace, aby uživatel nebyl nucen mít stále otevřenou stránku s transakcí.

**Řešení:** Do prototypu byla přidána funkcionalita, spočívající v možnosti skrytí stránky s transakcí bez nutnosti ukončení transakce. Ikona aktivní transakce byla poté přidána do transakční lišty umístěné nad navigační lištou.

- **Nedostatek:** Prototyp obsahoval neintuitivní design hlavní stránky s událostmi. Bylo zjištěno, že ve chvíli, kdy by se na hlavní stránce nacházely desítky událostí, bylo by nutné posouvat prstem po obrazovce až do doby, než se kurzor dostane na úroveň k první nadcházející události.

**Řešení:** Tento nedostatek byl vyřešen přidáním přepínače v horní části obrazovky, který umožní v jakékoli chvíli přepnout výběr nadcházejících a probíhajících událostí.

Uvedené nedostatky a jejich opravy byly také zahrnuty při tvorbě high-fidelity prototypů. Ty ale sloužily pouze jako podklad při designování aplikací a nebyly testovány stejným způsobem jako low-fidelity prototypy.



## ■ 5.3 Technologie

V této části se budu zabývat výběrem a zhodnocením vhodných technologií potřebných při implementaci obou klientských aplikací. Jelikož je nutné implementovat dvě klientské aplikace, je třeba najít takové technologie, které umožní lepší znovupoužitelnost kódu.

### ■ 5.3.1 Framework

Přestože je nutné implementovat dvě odlišné aplikace s rozdílnou funkcionalitou, je stále možné výrazně zkrátit jejich vývoj vhodnou volbou frameworku, který umožní v obou případech aplikovat obdobné implementační postupy. Jako vhodné řešení problému bylo identifikováno použití jednoho či kombinace frameworků jako jsou např.: Flutter, React a React Native, nebo Ionic. Jelikož mám zkušenosti s jazykem Javascript, rozhodl jsem se uvažovat pouze o řešeních postavených právě na jazyku Javascript. Dále jsem se rozhodl vyloučit řešení spočívající ve tvorbě progresivní webové aplikace, protože v současné době mají na rozdílných platformách různé vlastnosti. Příkladem může být například nemožnost posílat push notifikace u zařízení s iOS.

### ■ ReactJs a React Native

ReactJs i React Native jsou Javascriptové frameworky vytvořené a udržované společností Meta (dříve Facebook). Oba frameworky umožňují zapisování struktury aplikace pomocí JSX, které rozšiřuje syntax jazyka Javascript a umožňuje ho zapisovat ve formě podobné XML. Pro jednotlivé části aplikace je vhodné vytvářet komponenty, které umožní lepší přehlednost psaného kódu a bude je možné přepoužít [17].

React Native je oproti ReactJs zamýšlen jako multiplatformní framework a ne pouze jako nástroj pro tvorbu webových aplikací, který dodržuje syntax ReactJs. Hlavním cílem React Native je tvorba mostu mezi nativním kódem dané platformy a umožnění kompatibility napříč mobilními platformami iOS a Android. Bohužel je ale na mnoha místech značně omezen, jelikož neposkytuje podporu pro HTML a poskytuje pouze základní podporu CSS [18].

React není kompletním frameworkem a poskytuje pouze základní funkcionalitu. Pokud není požadovaná funkcionalita jeho součástí, je ji možné buď od základu implementovat, nebo využít komunitně vytvořených knihoven, které ovšem nezaručují dlouhodobou podporu dané funkcionality.

### ■ Angular a Ionic

Angular je oblíbeným frameworkem pro tvorbu webových aplikací. Je vyvíjen společností Google a umožňuje vytvářet kostry aplikací pomocí HTML v kombinaci s jazykem TypeScript, který představuje nadstavbu nad jazykem JavaScript. Narozdíl od výše zmíněného frameworku React je Angular navržen takovým způsobem, aby bylo možné oddělit aplikační logiku od uživatelského rozhraní a datového modelu aplikace. Je možné v něm využít návrhové

vzory jako jsou: *MVC* či *MVVM*. Dalším klíčovým rozdílem je i fakt, že je Angular představován jako kompletní framework, obsahující většinu potřebných technologií či knihoven, aniž by bylo nutné spoléhat se na open-source řešení pomocí komunitních knihoven. Z tohoto důvodu je tento framework náročnější pro pochopení a také se lépe uplatňuje zejména v komplexnějších projektech [19].

Ionic je framework vytvořený pro snazší vývoj hybridních mobilních aplikací. Uvádím jej s Angularem, jelikož byl původně vyvinut právě jako jeho rozšíření. V současné době je však možné tento framework využít s frameworkem ReactJs či Vue Js. Stejně jako Angular je i tento framework koncipován jako kompletní knihovna veškeré potřebné funkcionality a využívá návrhových vzorů *MVC* či *MVVM* [20].

### ■ Zvolené řešení

V rámci své práce jsem zvolil kombinaci frameworků ReactJs a React Native. React jsem zvolil především kvůli tomu, že má snazší učící křivku a je tedy možné obě aplikace vytvořit ve výrazně kratším čase než s využitím Angular. V případě, že bude nutné využít funkcionality, která není součástí frameworku ReactJs či React Native, budou využity knihovny třetích stran [21] [22].

### ■ 5.3.2 Globální správa stavů

Každá komponenta ve frameworku React má svůj vlastní stav, který představuje data, která se mohou v průběhu běhu aplikace měnit a při každé změně je určitá část komponenty znovu vykreslena. Určitá data je ale nutné poskytnout více částem aplikace, proto je třeba využít technologie, která zajišťuje systém globální správy stavů. Typická data nebo stavy, kvůli kterým je nutné tuto správu zavádět, jsou například informace o přihlášeném uživateli.

V této části zhodnotím dostupné technologie řešící problém sdílení dat mezi komponentami a vyberu vhodné řešení, které využiji v implementační části.

### ■ Redux Toolkit

Redux Toolkit je nadstavbou pro již existující knihovnu jménem Redux. Redux je velice oblíbenou a rozšířenou knihovnou, ale má zásadní problém - nutnost psaní stále opakujícího se kódu (dále v práci zmíněno jako *boilerplate kód*).

Redux poskytuje celé aplikaci komponentu *store*, která obsahuje stromový objekt. Listy tohoto stromu představují jednotlivé uložené stavy definované vývojářem. Veškerá data přidaná do *store* musí být serializovatelná a neměnná. Pokud má být *store* použit jako úložiště dat z API volání, je možné data získat voláním API ve zvolené komponentě a následně je vložit do *store* pomocí funkce *dispatch*. Další možností, jak data vložit do *store* představují *thunky*, což jsou Reduxem vytvořené middlewary umístěné v rámci *store*, které po zavolání určité akce provedou API volání a data odešlou přímo do něj.

Přístup s *thunky* usnadňuje zpracování chybových hlášek, jelikož umožňuje jejich zpracování na jediném místě v aplikaci.

Hlavním smyslem Redux Toolkitu je vyřešení problému s boilerplate kódem společně s uchováním a rozšiřováním základní funkcionality Redux [23].

## ■ React Context

React Context je metoda poskytnutá v rámci základní funkcionality Reactu. Umožňuje zapouzdřit určitou část aplikace do komponenty jménem *provider*, která poskytuje stavy pro všechny potomky komponenty. Pokud je nutné stavy získat, stačí zapouzdřit zvoleného potomka do komponenty jménem *consumer*.

React Context má zásadní nevýhodu, kterou je nutnost opětovného vykreslování všech potomků komponenty *provider*, při každé změně stavu [17].

## ■ Recoil

Recoil je experimentální knihovnou vytvořenou společností Meta pro správu globálních stavů. Oproti Redux je jeho značnou výhodou rozdělení jednotlivých stavů do nezávislých částí jménem *atomy*. Jednotlivý *atom* obsahuje stavy, které je možné získat pomocí selektorů a upravovat ve zvolené komponentě pomocí funkce *set*. Recoil také poskytuje možnost získávání dat z API volání uvnitř jednotlivých *atomů* [24].

## ■ Zvolené řešení

V rámci své práce jsem zvolil Redux Toolkit jako vhodné řešení správy stavů, jelikož oproti Recoil se jedná o stabilní knihovnu se stálou podporou a oproti React Context jsou v případě změny stavu znovu vykresleny jen určité komponenty, které daný stav používají. Redux Toolkit také poskytne možnost zjednodušení organizace API volání.

## ■ 5.4 Architektura platformy

Aby bylo zajištěno korektního fungování platformy, je třeba zvolit vhodné technologie, které umožní uložení dat, či komunikaci mezi serverem a mnou vytvořeným uživatelským rozhraním. Volbou těchto technologií se zabýval kolega Michal Šalaga v rámci své práce [10] a na základě jeho analýzy zde uvedu, jaké technologie bude implementovaná platforma vyžít a jak do celého konceptu zapadá má práce.

### ■ 5.4.1 MERN Stack

Na základě analýzy požadavků platformy byl jako základní soubor technologií vybrán tzn. MERN Stack. Jeho hlavní charakteristikou je umožnění implementace dynamických webových aplikací za pomoci jazyka Javascript, který

je známý pro svou efektivitu, nízkou paměťovou náročnost a škálovatelnost [25]. Základními částmi MERN Stack jsou:

- **MongoDB**

MongoDB je dokumentová databáze, která ukládá data ve formátu podobném JSON a mají svůj klíč a hodnotu. Databáze je typu NoSQL, který se od tradičních relačních databází liší absencí tabulek a v tomto případě jsou data ukládána do kolekcí.

- **Node.js**

Node.js je vývojové prostředí, které umožňuje exekuci jazyka Javascript mimo prostředí webového prohlížeče. Hlavním smyslem existence Node.js je umožnit tvorbu serverové části webových aplikací.

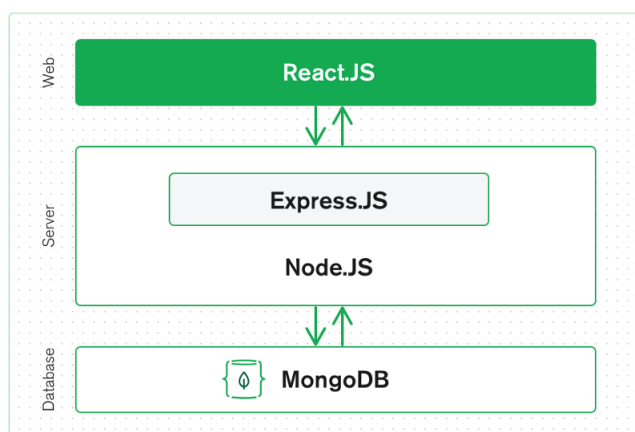
- **Express.js**

Express.js je framework pro výše zmíněný Node.js. Jeho hlavním cílem je poskytnutí dodatečné abstrakce pro snazší a rychlejší tvorbu serverových aplikací. Express.js také poskytuje řadu návrhových vzorů využitelných pro zlepšení škálovatelnosti aplikace či redukci přebytečného kódu. Příkladem těchto návrhových vzorů je například *middleware*, *MVC* či *router*.

- **React.js**

React.js je poslední částí zmíněného MERN Stacku a představuje klientskou část vyvíjené aplikace komunikující s Node.js serverem.

Obrázek níže zachycuje soubor technologií MERN Stack: 5.5 <sup>3</sup>



**Obrázek 5.5:** Grafický popis souboru technologií MERN Stack [9]

<sup>3</sup>Obrázek byl po domluvě s vedoucím práce ponechán v původním jazyce.

### ■ 5.4.2 REST

REST je architektura pro webové rozhraní, která umožňuje snadno a jednotně pracovat se zdroji ze serverové části (v našem případě za pomoci HTTP protokolu). Každý zdroj má svůj vlastní URI identifikátor a metodu, jak k němu lze přistoupit. S daty lze při použití REST manipulovat pomocí čtyř základních operací - Create, Read, Update a Delete, přičemž každá z těchto operací má svou vlastní implementaci. Metoda GET implementuje operaci Read, POST implementuje operaci Create, DELETE operaci Delete a metoda PUT operaci Update. Pro každou ze zmíněných metod existuje specifická serverová odpověď, na základě které lze ověřit, zda došlo k propagaci změn či byl zjištěn problém [26].

REST byl na základě kolegovy analýzy požadavků a zkušeností zvolen jako primární přístup ke zdrojům ze serverové části. Použití tohoto způsobu komunikace bude proto následně zohledněno v implementační části mé práce.

V alternativním případě by bylo možné místo REST využít knihovny GraphQL či Falcor.

### ■ 5.4.3 Socket.io

Na základě nefunkčního požadavku, týkajícího se částečné real-time komunikace, byla kolegou pro následnou implementaci zvolena technologie Socket.io. Knihovna Socket.io byla vybrána především na základě její oblíbenosti.

Socket.io je knihovna vytvořená v jazyce Javascript, určená pro oboustrannou komunikaci v reálném čase. Pro korektní oboustrannou komunikaci je nutné, aby klientská aplikace využívala část knihovny jménem Socket.io Client API a serverová knihovnu Socket.io Server API. Knihovna primárně využívá protokol WebSocket a v případě, kdy pomocí něj není možné navázat připojení, je toto připojení navázáno prostřednictvím HTTP mechanismu *long polling* [27].

Využití Socket.io se jeví jako vhodné řešení v případě implementace transakcí, které jsou součástí základní funkcionality celé platformy, jelikož je v tomto případě synchronizace mezi webovou a mobilní aplikací klíčová.

Jako vhodné alternativní řešení se jeví použití knihovny ws, SockJS či Sockette.

### ■ 5.4.4 OAuth 2.0

Aby bylo částečně vyhověno nefunkčnímu požadavku týkajícího se bezpečnosti platformy, byl vybrán protokol OAuth 2.0 jako vhodné řešení autorizace při vzájemné komunikaci mezi klientskou a serverovou částí platformy. Jeho využití bylo rovněž zohledněno na základě předchozích zkušeností při tvorbě webových aplikací.

Základním principem OAuth 2.0 je využití tokenů - *access tokenů* a případně i *refresh tokenů*. Tyto tokeny jsou distribuovány ve specifickém formátu klientské straně, která si je následně uloží, aby mohla pomocí nich přistupovat k autorizovaným zdrojům. Jelikož má *access token* pouze omezenou životnost,

je nutné ho v případě expirace nahradit novým. Nový *access token* lze získat pomocí specifického serverového požadavku, který bude obsahovat uložený *refresh token* [28].

V následné implementaci byla zohledněna nutnost uložení *access tokenu* a *refresh tokenu* na klientské straně kvůli existenci mobilní aplikace a proto byly oba tokeny uloženy v lokálním úložišti.

# Kapitola 6

## Implementace

V následující kapitole se věnuji popisu, jak byla klientská část implementována. Nejprve popíši architekturu projektů, následně uvedu jejich strukturu a nakonec představím způsob, jak jsem vyřešil komunikaci mezi serverovou a klientskou částí platformy.

### 6.1 Architektura klientských aplikací

Obě klientské aplikace, vytvořené v rámci mé bakalářské práce, dodržují stejnou architekturu, jelikož jsem chtěl dosáhnout co největší možnosti znovupoužitelnosti kódu a urychlit čas věnovaný vývoji aplikací.

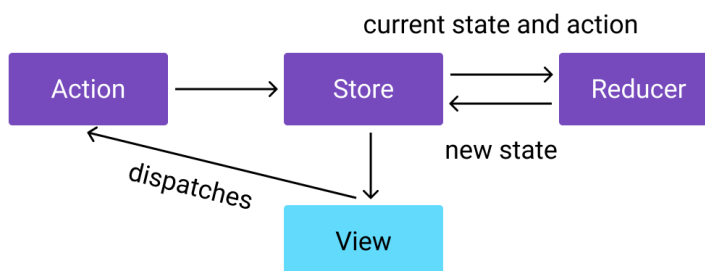
Architektura aplikací využívá vybranou technologii Redux Toolkit pro globální správu stavů [23]. Prakticky všechna data získaná ze serverové části jsou ihned propagována do objektu *store*, který je součástí komponenty *provider*. Jakýkoli element, který je potomkem komponenty *provider* smí přistoupit k datům uvnitř *store* a odeslat akce pomocí funkce *dispatch*, která poté zapříčiní obnovu dat ze serverové části.

V případě dat, která není třeba uložit, aby k nim mělo přístup více komponent, je uplatněn přístup lokálního uložení v rámci stavu konkrétní komponenty. Mezi tato data patří obsahy formulářových polí, či filtrovaná pole podle uživatelem zadaného vstupu.

Obrázek níže zachycuje datový tok knihovny Redux: 6.1 <sup>1</sup>

---

<sup>1</sup>Obrázek byl po domluvě s vedoucím práce ponechán v původním jazyce.



Obrázek 6.1: Grafický popis knihovny Redux Toolkit [8]

## 6.2 Struktura projektů

Stejně jako v případě architektury, je i struktura projektů aplikací velmi obdobná. Jelikož se ale jedná o mobilní a webovou aplikaci, bylo třeba využít v obou případech odlišné knihovny.

Oba projekty doržují kromě základních souborů, vytvořených při založení projektu, následující strukturu:

- **Složka Actions** - obsahuje soubory s výčtem akcí potřebných při práci s Redux Toolkit.
- **Složka Features** - obsahuje soubory, které odpovídají jednotlivým listům stromového objektu *store* a definují operace nad daty společně s jejich strukturou.
- **Složka Functions** - obsahuje soubory s pomocnými funkcemi, které je možné přepoužít v rámci celého projektu.
- **Složka Hooks** - obsahuje soubory s přepoužitelnými funkcemi typu *hook*, které pracují s životním cyklem komponent a spravují jejich stavy.
- **Složka Pages** - obsahuje složky s jednotlivými stránkami aplikace. Zmíněné složky obsahují:
  - Soubor `Page.js` - obsahuje zdrojový kód hlavní komponenty stránky
  - Soubor `Styled.Page.js` - obsahuje stylované elementy, které jsou určeny pro použití v komponentě obsahující stránku.
  - Soubor `ValidationSchema.js` - obsahuje validační podmínky pro formulář, který je součástí konkrétní stránky.
  - Složka `Components` - obsahuje zdrojové kódy méně komplexních komponent, které nelze přepoužít v jiné části projektu, kromě konkrétní stránky.



- **Složka Resources** - obsahuje obrázky a další soubory, které se nepovažují za zdrojový kód.
- **Složka Reusable components** - obsahuje zdrojové kódy méně komplexních komponent, které lze v projektu přepoužít dle potřeby. Ve složce se vždy nachází:
  - Soubor Component.js - obsahuje zdrojový kód znovupoužitelné komponenty.
  - Soubor Styled.Component.js - obsahuje stylované elementy, které jsou určeny pro použití v komponentě.
- **Složka Store** - obsahuje soubory, které implementují *middleware* rozšíření pro *store* a soubor se základní konfigurací *store*.

### 6.2.1 Mobilní aplikace

Z důvodu použití knihoven React Navigation, React Reanimated 2 a přítomnosti složek s nativním kódem je projekt, obsahující implementaci mobilní aplikace, doplněn o následující složky:

- **Složka Android** - obsahuje zdrojové kódy, které umožňují kompilaci React Native projektu do nativního kódu platformy Android.
- **Složka iOS** - obsahuje zdrojové kódy, které umožňují kompilaci React Native projektu do nativního kódu platformy iOS.
- **Složka Navigation** - obsahuje soubory, které definují posloupnost všech stránek v mobilní aplikaci a složku s komponentami, které úzce souvisí s navigací v aplikaci.

Kromě uvedených složek se na úrovni určitých komponent vyskytuje **složka Animation Hooks**, která obsahuje funkce typu *hook*. V tomto případě se jedná o *hooky*, které implementují animace pomocí knihovny React Reanimated 2.

### 6.2.2 Webová aplikace

Obdobně jako mobilní aplikace, využívá i webová aplikace knihovnu umožňující navigaci mezi jednotlivými stránkami - React Router. **Složka Navigation** obsahuje soubor s výčtem všech cest stránek, ke kterým lze v rámci aplikace přistoupit. Dále obsahuje komponenty, které omezují přístup k určitým stránkám, pokud nemá uživatel požadovanou roli a komponenty, které určují rozložení aplikace.

## 6.3 Komunikace se serverovou částí

V následující části popisují technologie, které byly využity pro navázání komunikace se serverovou částí. Kromě jejich popisu také uvedu, jak dané technologie zapadají do celkové architektury aplikací ??.

### 6.3.1 Fetch Base Query

Pro navázání REST komunikace se serverovou stranou jsem využil *fetch base query*, která vytváří dodatečnou abstrakci nad základní metodou *fetch* jazyka Javascript. *Fetch base query* je součástí knihovny Redux Toolkit, vybrané v rámci návrhové části bakalářské práce.

Využitím *fetch base query* lze standardizovat veškerá REST volání v rámci jediného objektu jménem *API slice*. Chybové i nechybové stavy provedného volání je možné poté zpracovat ve vlastním *middleware* rozšíření komponenty *store*. Kromě redukce psaného kódu poskytuje *fetch base query* možnost uložení dat z externích zdrojů do *cache*, odkud lze k získaným datům snadno přistoupit [23].

### 6.3.2 Socket.io Client API

Klientská část knihovny Socket.io je v rámci projektu využita pro operace, požadující získání a propagaci dat v reálném čase. Implementace Socket.io Client umožní přihlášenému uživateli získat informace o všech probíhajících transakcích, aniž by bylo nutné obnovit stránku internetového prohlížeče, či mobilní aplikace.

Přestože je většina API volání realizována prostřednictvím objektu *API slice*, není možné uplatnit stejný postup pro socketovou komunikaci, jelikož nepodporuje danou funkcionalitu. Z tohoto důvodu jsem vytvořil vlastní *middleware* pro komponentu *store*, aby bylo možné dodržet zásady zvolené architektury aplikací. V rámci vytvořeného *middleware* byly definovány všechny akce, na základě kterých má dojít k odeslání nebo přijetí zpráv ze serverové části. Jednotlivé akce jsou následně podle konkrétního požadavku odeslány prostřednictvím funkce *dispatch* a veškerá data z příchozích zpráv či odpovědí na odchozí zprávy jsou propagována do mnou definovaného *slice* objektu nazvaného **Transaction Slice** [27] [23].

# Kapitola 7

## Testování

V této kapitole se věnuji ověření základní funkcionality platformy pomocí uživatelského testování. Testování hraje v rámci vývoje software klíčovou roli, která spočívá v nalezení chyb ohrožujících chod software v reálném provozu. Testování pomocí uživatelských testů umožní ověřit, zda klientská strana korektně komunikuje se serverovou částí a umožní odhalit nedostatky, které mohou potencionálně narušit uživatelský zážitek. Současně byla otestována i funkcionality serverové části, kterou implementoval kolega Michal Šalaga v rámci své práce [10].

### 7.1 Průběh testování

Pro účely testování jsem sepsal deset scénářů do souboru na platformě Google Docs a vytvořil dotazník pro každý scénář s textovým polem pro popsání nalezené chyby. Následně jsem kontaktoval pět <sup>1</sup> spolužáků, se kterými jsem se dohodl na společném termínu testování. Každému spolužákovi jsem odeslal email obsahující unikátní přihlašovací údaje (heslo a jméno správce) pro webovou aplikaci společně s odkazy pro dotazník, testovací scénáře, adresu webové aplikace a odkaz pro stažení *apk* souboru mobilní aplikace.

V den testování jsem založil na platformě Microsoft Teams společný kanál a v domluveném čase zahájil hovor. Spolužáky jsem poté nechal pracovat na testovacích scénářích a s kolegou Michalem Šalagou jsem na celý proces dohlížel, kdyby došlo k chybě na serverové straně.

Testování trvalo přibližně dvě hodiny a od každého spolužáka jsem v průběhu obdržel formulář s výstupy testování. Spolužáci kromě průchodů podle testovacích scénářů zkoušeli úmyslně vkládat nevalidní data do formulářových polí, náhodně procházeli oběma aplikacemi a následně zapisovali nalezené chyby. Při testování bylo nutné, aby byli účastníci současně přihlášení jako správci platformy ve webové aplikaci a jako běžní uživatelé v mobilní aplikaci. Díky tomu bylo možné ověřit schvalování účtenek, či transakcí.

---

<sup>1</sup>Jedná se o odlišnou skupinu testovacích subjektů než u testování prototypů. 5.2.2

## 7.2 Testovací scénáře

### 7.2.1 Scénář 1 - Registrace nového uživatele mobilní aplikace

- **Vstupní data:** Neregistrovaný uživatel
- **Testovací kroky:**
  1. Mobilní aplikace
    - a. Uživatel stiskne tlačítko „Registrovat se“.
    - b. Uživatel vyplní formulářová pole a odešle jej kliknutím na tlačítko „Přihlásit se“.
- **Očekávaný výstup:** Po úspěšném odeslání formuláře je v databázi uložen nový uživatel a dojde k přesměrování v mobilní aplikaci na stránku „Události“.

### 7.2.2 Scénář 2 - Přihlášení uživatele mobilní aplikace

- **Vstupní data:** Nepřihlášený registrovaný uživatel
- **Testovací kroky:**
  1. Mobilní aplikace
    - a. Uživatel stiskne tlačítko „Přihlásit se“.
    - b. Uživatel vyplní formulářová pole a odešle jej kliknutím na tlačítko „Přihlásit se“.
- **Očekávaný výstup:** Po úspěšném odeslání formuláře dojde k přesměrování v mobilní aplikaci na stránku „Události“.

### 7.2.3 Scénář 3 - Přihlášení uživatele webové aplikace

- **Vstupní data:** Nepřihlášený správce nebo nepřihlášený provozovatel podniku
- **Testovací kroky:**
  1. Webová aplikace
    - a. Uživatel vyplní formulářová pole a odešle jej kliknutím na tlačítko „Přihlásit se“.
- **Očekávaný výstup:** Po úspěšném odeslání formuláře dojde k přesměrování ve webové aplikaci na stránku „Transakce“.

#### 7.2.4 Scénář 4 - Změna hesla uživatele

- **Vstupní data:** Přihlášený uživatel
- **Testovací kroky:**
  1. Mobilní aplikace
    - a. Uživatel přejde na stránku „Profil“.
    - b. Uživatel vybere stisknutím položku s názvem „Nastavení“.
    - c. Uživatel vybere stisknutím položku s názvem „Změna hesla“.
    - d. Uživatel vyplní formulářová pole validními daty a stiskne tlačítko „Potvrdit“.
- **Očekávaný výstup:** Po úspěšném odeslání formuláře dojde v databázi k přepsání aktuálního hesla za nové heslo a pokud bude již přihlášený uživatel odhlášen, bude muset použít k přihlášení nové heslo.

#### 7.2.5 Scénář 5 - Zablokování uživatele

- **Vstupní data:** Přihlášený správce
- **Testovací kroky:**
  1. Webová aplikace
    - a. Uživatel přejde na stránku „Uživatelé“.
    - b. Uživatel klikne na tlačítko „Zablokovat“ u uživatele, který není správce.
    - c. Uživatel potvrdí akci stisknutím tlačítka „Potvrdit“, nacházejícího se v nově otevřeném modálním okně.
- **Očekávaný výstup:** Zablokovaný uživatel se nebude moci přihlásit do mobilní či webové aplikace a pokud je již přihlášený, bude odhlášen.

#### 7.2.6 Scénář 6 - Vytvoření účtu podniku

- **Vstupní data:** Přihlášený správce
- **Testovací kroky:**
  1. Webová aplikace
    - a. Uživatel přejde na stránku „Podniky“.
    - b. Uživatel klikne na tlačítko „Vytvořit podnik“.
    - c. Uživatel vyplní všechna pole ve formuláři validními daty a stiskne tlačítko „Vytvořit“.
- **Očekávaný výstup:** Po úspěšném odeslání formuláře dojde k vytvoření nového objektu podniku a nového účtu provozovatele tohoto podniku. Nově vytvořený podnik bude možné nalézt ve webové aplikaci na stránce „Podniky“ a bude možné se do webové aplikace přihlásit pomocí přihlašovacích údajů provozovatele podniku.

### 7.2.7 Scénář 7 - Vytvoření události

- **Vstupní data:** Přihlášený správce
- **Testovací kroky:**
  1. Webová aplikace
    - a. Uživatel přejde na stránku „Podniky“.
    - b. Uživatel kliknutím zvolí podnik z načteného seznamu.
    - c. Uživatel klikne na tlačítko „Vytvořit událost“.
    - d. Uživatel vyplní všechna pole ve formuláři validními daty a stiskne tlačítko „Vytvořit“.
- **Očekávaný výstup:** Po úspěšném odeslání formuláře dojde k zápisu nové události do databáze. Nově vytvořenou událost bude možné nalézt ve webové i mobilní aplikaci na stránce „Události“.

### 7.2.8 Scénář 8 - Schválení účtenky

- **Vstupní data:** Přihlášený správce (Webová aplikace), přihlášený uživatel (Mobilní aplikace), probíhající událost
- **Testovací kroky:**
  1. Mobilní aplikace
    - a. Uživatel přejde na stránku „Události“ a klepnutím zvolí událost nacházející se v sekci „Probíhající“.
    - b. Uživatel stiskne tlačítko „Přidat účtenku“.
    - c. Uživatel zvolí možnost „Přidat z galerie“ z nově zobrazeného modálního okna.
    - d. Uživatel stisknutím nahraje vybranou fotografií z galerie mobilního zařízení.
  2. Webová aplikace
    - a. Uživatel přejde na stránku „Události“.
    - b. Uživatel klikne na filtrovací tlačítko obsahující nápis „Probíhající“.
    - c. Uživatel v seznamu kliknutím vybere totožnou událost, do které v předchozí části nahrál novou fotografii účtenky.
    - d. Uživatel najde v sekci „Účtenky“ fotografii nahrané účtenky z předchozí části a kliknutím ji zvolí.
    - e. Uživatel vyplní na základě fotografie formulář umístěný v pravé části nově zobrazeného modálního okna. Do pole „ID účtenky“ zapíše identifikační číslo účtenky, do pole „Hodnota“ zapíše celkovou utracenou částku z účtenky a stiskne tlačítko „Schválit“.

- **Očekávaný výstup:** Po úspěšném schválení účtenky dojde v databázi k přiřazení uživatele mobilní aplikace k události, ke které přidal fotografii účtenky. Účtenku nahranou k události bude možné zobrazit po stisknutí tlačítka „Historie“ na stránce s detailem události v mobilní aplikaci.

### 7.2.9 Scénář 9 - Ukončení události

- **Vstupní data:** Přihlášený správce

- **Testovací kroky:**

1. Webová aplikace
  - a. Uživatel přejde na stránku „Události“.
  - b. Uživatel klikne na filtrovací tlačítko obsahující nápis „Probíhající“.
  - c. Uživatel kliknutím zvolí probíhající událost.
  - d. Uživatel klikne na tlačítko „Ukončit událost“.
  - e. Uživatel potvrdí akci kliknutím na tlačítko „Potvrdit“ nacházejícím se v nově zobrazeném modálním okně.

- **Očekávaný výstup:** Po úspěšném ukončení události bude v databázi změněn stav události jako „Ukončená“ a nebude ji následně možné nalézt na stránce „Události“ v mobilní aplikaci. Ve stejnou chvíli dojde v případě vítězství v události k přiřazení odměny.

### 7.2.10 Scénář 10 - Potvrzení transakce

- **Vstupní data:** Přihlášený správce nebo přihlášený provozovatel podniku (Webová aplikace), přihlášený uživatel (Mobilní aplikace)

- **Testovací kroky:**

1. Webová aplikace
  - a. Uživatel přejde na stránku „Transakce“.
2. Mobilní aplikace
  - a. Uživatel přejde na stránku „Odměny“.
  - b. Uživatel kliknutím zvolí odměnu.
  - c. Uživatel vyplní formulářové pole v otevřeném modálním okně číselnou hodnotou a stiskne tlačítko „Aktivovat odměnu“.
3. Webová aplikace
  - a. Uživatel potvrdí novou transakci kliknutím na tlačítko „Potvrdit“.

- **Očekávaný výstup:** Po úspěšném potvrzení dojde k uzavření stránky s detailem transakce v mobilní aplikaci a dojde k odečtení veškerých potvrzených kreditů přihlášeného uživatele v rámci dotyčného podniku.

### 7.3 Výstupy testování

Následující tabulka 7.1 shrnuje výstupy všech testovacích scénářů (značení TS) pro každý subjekt (značení S). V případě, že test skončil úspěšně, použil jsem symbol ✓ a v opačném případě symbol ✗.

	S1	S2	S3	S4	S5
TS1	✓	✓	✓	✓	✓
TS2	✓	✓	✓	✓	✓
TS3	✓	✓	✓	✓	✓
TS4	✓	✓	✓	✓	✓
TS5	✓	✓	✓	✓	✓
TS6	✓	✓	✓	✓	✓
TS7	✓	✓	✓	✓	✓
TS8	✓	✓	✓	✓	✓
TS9	✓	✓	✓	✓	✓
TS10	✓	✓	✓	✓	✓

**Tabulka 7.1:** Výstupy uživatelského testování

Z poskytnutého formuláře vyplývá, že všechny testovací scénáře proběhly úspěšně. Na jejich základě jsem usoudil, že základní funkcionality platformy funguje správně.

Na druhou stranu byla odhalena chyby, které mohou potenciálně ohrozit funkci celé platformy. Nalezené chyby se týkají jak klientské, tak serverové části a v rámci této práce se budu zabývat pouze těmi, které se týkají klientské strany. Serverovými nedostatky se zabývá kolega Michal Šalaga ve své bakalářské práci [10]. Mezi chyby, nalezené na klientské straně, patří:

- Přítomnost obecných chybových hlášek, které uživateli nenaznačují, kde udělal chybu.
- Absence hlášek v případě úspěchu určité akce.
- Absence označení pro jednotlivé stavy událostí.
- Absence validačních podmínek u určitých formulářových polí.
- Nedostatečně vysvětlené validační chyby u formulářových polí.

Uvedené chyby budou opraveny až po odevzdání bakalářské práce.



## Kapitola 8

### Využití platformy na trhu

V této kapitole se zabývám zhodnocením využití navržené platformy na trhu. Nejprve provedu SWOT analýzu ze strany vývojářů a poté navrhnu první verzi jejího obchodního modelu. Provedení SWOT analýzy a navržení obchodního modelu bude potřebné pro situaci, ve které dojde k založení společnosti.

#### 8.1 SWOT analýza

SWOT analýza je metoda určená pro analýzu vnějších a vnitřních vlivů daného projektu, tématu či produktu. Díky znalosti těchto vlivů jsme schopni sestavit strategické cíle, které budou respektovat pozitivní či negativní okolnosti, ovlivňující průběh exekuce dané problematiky [29]. Mezi základní části matice SWOT analýzy patří:

- **Silné stránky** - jsou takové aspekty z vnitřního prostředí firmy, ve kterých je daná firma lepší než konkurence.
- **Slabé stránky** - jsou takové aspekty z vnitřního prostředí firmy, které snižují její vnitřní hodnotu. Slabé stránky je třeba minimalizovat.
- **Příležitosti** - jsou takové aspekty z vnějšího prostředí firmy, které jí mohou přinést posílení na trhu či lepší interní rozvoj.
- **Hrozby** - jsou takové aspekty z vnějšího prostředí firmy, které mohou ohrožit dosažení jejích strategických cílů.

V následující části provedu SWOT analýzu soutěžní platformy z mé a kolegovy strany pro případ, kdybychom chtěli platformu uvést do reálného provozu. Provedení analýzy usnadní tvorbu následného návrhu obchodního modelu.

##### 8.1.1 Silné stránky

- Nadšení do nových technologií
- Kreativní myšlení
- Zkušenosti z předchozích projektů

### ■ 8.1.2 Slabé stránky

- Malá velikost týmu
- Netrpělivost
- Nedostatek financí

### ■ 8.1.3 Příležitosti

- Pomoc s potlačením důsledků pandemické krize
- Vytvoření nového software
- Možnost podnikání

### ■ 8.1.4 Hrozby

- Opětovné zhoršení situace pro realizaci projektu
- Nezájem o nápady
- Konkurence

### ■ 8.1.5 Vyhodnocení SWOT analýzy

Provedením SWOT analýzy byly odhaleny vlivy, které mohou zásadním způsobem ovlivnit exekuci obchodního plánu společnosti. V následující části se budu věnovat návrhu obchodního modelu, který by mohl sloužit jako základ budoucího obchodního plánu společnosti.

## ■ 8.2 Obchodní model platformy

Obchodní model představuje rozvahu, jak daný podnik vytváří či předává hodnotu svým zákazníkům a jakým způsobem získává hodnotu pro sebe sama. Model dělí základní principy fungování podniku do jednotlivých segmentů, které popisují oblasti, ve kterých operuje.

V případě, že dojde k reálnému využití naší platformy, je nutné navrhnout odpovídající obchodní model. Vytvořený návrh poskytne přehled o fungování společnosti, což umožní lépe analyzovat a v budoucnu optimalizovat její aktivity v daných oblastech [30]. Při tvorbě obchodního modelu jsem se inspiroval šablonou *business model canvas* a jejími částmi.

### ■ 8.2.1 Klíčoví partneři

- Majitelé podniků
- **Poskytovatelé technologií** - implementace a správný chod celé platformy vyžaduje technologie vyvinuté třetími stranami.

### ■ 8.2.2 Klíčové aktivity

- **Organizace událostí** - společnost je zodpovědná za vytváření událostí ve spolupracujících podnicích, schvalování nahraných účtenek v nich a poskytnutí odměn pro vítěze daných událostí.
- **Vývoj a údržba platformy**
- **Hledání potencionálních klientů (majitelé podniků)**

### ■ 8.2.3 Klíčové zdroje

- **Soutěžní platforma**
- **Vývojáři**
- **Prodejní síla** - zaměstnanci, kteří se snaží získat nové spolupráce s podniky.

### ■ 8.2.4 Poskytovaná hodnota

- **Slevy na zboží** - uživatelé mohou ve spolupracujících podnicích získat slevy na zboží z menu, výměnou za získané kredity.
- **Zvýšení návštěvnosti a zisku ve spolupracujících podnicích**

### ■ 8.2.5 Vztahy se zákazníky

- **Zákaznická podpora**
- **Informování o novinkách událostí**
- **Informování o získaných odměnách**
- **Poskytování statistik o proběhlých událostech majitelům podniků**

### ■ 8.2.6 Distribuční kanály

- **Sociální sítě**
- **Reklamní předměty ve spolupracujících podnicích**
- **Webová stránka**
- **Mobilní aplikace**

### ■ 8.2.7 Zákaznické segmenty

- **Majitelé podniků**
- **Zákazníci podniků**

Při sestavování příjmů 8.2.9 a nákladů 8.2.8 8.2.10 zohledňuji zákazníky podniků, kteří se budou účastnit událostí a fotit účtenky, které bude nutné ukládat v systému. Oproti tomu budou majitelé podniků poskytovat prostory pro realizaci akcí a provize z útrat zákazníků, ze kterých bude společnost generovat příjmy.

### ■ 8.2.8 Struktura nákladů

V této části uvádím porovnání měsíčních nákladů dvou scénářů, ke kterým by mohlo potencionálně dojít v případě snahy o uvedení platformy na trh. Oba scénáře pracují se situací, ve které by působnost platformy zahrnovala pouze oblast Prahy.

Aby bylo možné pokrýt veškeré náklady obou scénářů, byly provedeny iterace, ve kterých došlo nejprve k určení všech fixních nákladů v projektu. S pomocí znalosti fixních nákladů jsem vypočítal, kolik bude třeba uspořádat událostí, aby je bylo možné pokrýt 8.2.9 a nakonec jsem dle počtu událostí stanovil náklady na cloudové technologie 8.2.10. Po uvedení těchto nákladů jsem vyhodnotil, zda je třeba navýšit zisky (uspořádat více událostí). Níže uvádím nejrealističtější iteraci pro každý scénář, ke které jsem došel.

### ■ Studentský projekt

Následující scénář zahrnuje náklady již existující společnosti v jejím počátku, přičemž je projekt realizován v týmu o dvou členech jako dosud.

- **Internet** - 461 Kč (Vodafone 300 Mbit včetně pronájmu modemu)
- **Náklady na dopravu** - 260 Kč (Lítačka se studentskou slevou pro dvě osoby)
- **Letáky** - 1000ks za 1400 Kč (jednostranné letáky rozměru A5 dle webové stránky letaky4u.cz)
- **Ostatní náklady**

Zde jsou uvedeny náklady, které nelze ihned vyčíslit, jelikož není platforma v reálném provozu a jejich hodnota se bude odvíjet buď od zisků nebo výkonu celé platformy. Z tohoto důvodu zde uvedu pouze orientační částky představující minimum, kterého chceme dosáhnout. V případě cloudových technologií bude částka určena v samostatné části kvůli návaznosti na část definující zdroje příjmů 8.2.10.

  - Reklama na sociálních sítích - 10 000 Kč
  - Odměny - 5000 Kč

- Cloudové technologie
  - Loggly - služba pro logování dat ze serveru
  - AWS S3 - služba poskytující úložiště objektů
  - AWS SES - služba umožňující odesílání emailů
  - Heroku - služba umožňující běh aplikací v cloudovém prostředí

Celkové náklady v tomto případě činí celkem **17 121 Kč**.

## ■ Startup

Tento scénář zahrnuje náklady již existující společnosti v případě rozšíření týmu, který by zahrnoval 4 členy s různými úlohami a bylo by třeba najít společný prostor pro práci. Rovněž se jedná o situaci, kdy již všichni členové pracují mimo prostředí školy.

- **Malá kancelář** - 4000 Kč (kancelář o rozměru 30m2 včetně internetu)
- **Platy** - 64 800 Kč (16 200 Kč mzda pro 4 lidi)
- **Náklady na dopravu** - 550 Kč (cena Lítačky pro člena, který shání nové potencionální klienty)
- **Letáky** - 1000ks za 1400 Kč (jednostranné letáky rozměru A5 dle webové stránky letaky4u.cz)
- **Ostatní náklady**

Zde jsou opět uvedeny náklady, které nelze ihned vyčíslit. Opět zde uvedu minimum, kterého chceme dosáhnout, přičemž cloudové technologie budou vyčísleny v oddělené části 8.2.10.

- Reklama na sociálních sítích - 25 000 Kč
- Cloudové technologie
  - Loggly - služba pro logování dat ze serveru
  - AWS S3 - služba poskytující úložiště objektů
  - AWS SES - služba umožňující odesílání emailů
  - Heroku - služba umožňující běh aplikací v cloudovém prostředí

Celkové náklady v tomto případě činí celkem **95 750 Kč**.

## ■ 8.2.9 Zdroje příjmů

- **Provize z prodaného zboží** - společnost získá dohodnutou provizi z každé platné nahrané účtenky, vyfotografované během události. Výše provize se bude odvíjet od úspěchu projektu a pokrytí nákladů, vyčíslených v předchozí části. Pokud by výše provize činila 10 procent z celkové útraty v soutěži, průměrná útrata činila 300 Kč a průměrně by se zúčastnilo 30 zákazníků, bylo by nutné měsíčně uspořádat přibližně následující počet událostí:

- **Studentský projekt** - 20 událostí (tj. nejméně 600 účtenek/zákazníků)
- **Startup** - 107 událostí (tj. nejméně 3210 účtenek/zákazníků)

Z výše uvedených dat vyplývá, že pro udržení projektu v případě startupu by bylo nutné pozměnit podnikatelskou strategii, jelikož by nebylo možné organizovat takové množství akcí v tak malém týmu. Na druhou stranu je projekt možné udržovat v provozu alespoň jako volnočasovou aktivitu během studia.

### ■ 8.2.10 Náklady za cloudové technologie

Aby bylo možné určit orientační výši nákladů za cloudové technologie, bude třeba vycházet z poznatků, které jsem uvedl v části věnující se zdrojům příjmů 8.2.9. Na základě minimálního počtu událostí jsem určil možnou výši nákladů za služby, která by pokryla měsíční potřeby pro daný scénář.

#### ■ Náklady pro studentský projekt

- Loggly - 1921,75 Kč (Standard plán zahrnující 3 zdrojové skupiny)
- AWS S3 - 62,76 Kč (3 GB uložených dat, která jsou průměrně 10-krát získána)
- AWS SES - 77,05 Kč (34 000 emailů - registrace uživatelů, notifikace o nových a ukončených událostech, notifikace o výsledcích transakcí)
- Heroku - 158,65 Kč (512 MB RAM - Hobby plán)

Pokud by náklady za cloudové technologie v případě studentského projektu činily celkem **2220,21 Kč**, bylo by nutné zvýšit počet organizovaných událostí o 2, aby byly pokryty všechny zmíněné náklady.

#### ■ Náklady pro startup

- Loggly - 1921,75 Kč (Standard plán zahrnující 3 zdrojové skupiny)
- AWS S3 - 271,99 Kč (13 GB uložených dat, která jsou průměrně 10-krát získána)
- AWS SES - 1536,28 Kč (678 000 emailů - registrace uživatelů, notifikace o nových a ukončených událostech, notifikace o výsledcích transakcí)
- Heroku - 566,64 Kč (512 MB RAM - Standard 1X)

Pokud by náklady za cloudové technologie v případě startupu činily celkem **4296,66 Kč**, bylo by nutné zvýšit počet organizovaných událostí o 5, aby byly pokryty všechny zmíněné náklady.

### ■ 8.2.11 Vyhodnocení obchodního modelu

Z navrženého obchodního modelu vychází, že je pro smysluplné využití platformy třeba dbát na vyvážení nákladů a zisků, aby nedošlo ke ztrátám při využití cloudových služeb.

# Kapitola 9

## Závěr

Cílem této bakalářské práce byla analýza, návrh, implementace a testování základní funkcionality klientské části soutěžní platformy pro podporu gastronomického odvětví.

V úvodu práce jsem uvedl, jak jsem s kolegou Michalem Šalagou [10] dospěl k nápadu vytvořit soutěžní platformu pro podporu gastronomických podniků, která řeší nízkou návštěvnost v nich. Poté jsem provedl rešerši existujících řešení, která se zabývají danou problematikou a vyhodnotil jsem, zda má naše řešení smysl implementovat. Zjistil jsem, že neexistuje žádné ekvivalentní řešení vůči námi navržené soutěžní platformě. Poté jsem se věnoval návrhu klientské strany platformy a kolega ve své práci části serverové [10]. Pro implementaci aplikací jsem zvolil framework ReactJs (webová aplikace) a React Native (mobilní aplikaci). Rovněž jsem provedl rešerši potřebných knihoven a uvedl, jak má práce navazuje na práci kolegy Šalagy [10]. Dále jsem se věnoval výběru technologií pro tvorbu prototypů klientských aplikací. Po vytvoření low-fidelity prototypů a jejich testování jsem se věnoval high-fidelity prototypům, aby bylo možné snáze stylizovat budoucí klientské aplikace. Následně jsem se věnoval implementaci aplikací a popsal jsem jejich strukturu a architekturu. Výsledné aplikace jsem otestoval prostřednictvím uživatelského testování. Nakonec jsem provedl SWOT analýzu z naší strany a navrhl prvotní verzi obchodního modelu platformy.

V rámci své práce jsem splnil všechny body ze svého zadání a celý projekt byl pro mne velkou zkušeností, jelikož jsem se naučil pracovat s novými technologiemi a mohl jsem je vyzkoušet na mnohem komplexnějším projektu, než jsem do této doby implementoval. Současně jsem splnil stanovený cíl, který spočíval ve tvorbě a implementaci řešení, které by mohlo pomoci s problematikou nízké návštěvnosti v gastronomických podnicích. Rovněž jsem se naučil lépe pracovat v týmu.

### 9.1 Budoucí stav

Obě klientské aplikace jsou v aktuálním stavu plně funkční. Během testování ale byly odhaleny chyby, které mohou potenciálně ohrozit běh platformy v reálném provozu. Před jejím případným zveřejněním bude třeba opravit nalezené chyby.

Před uvedením platformy na trh bude také nutné sestavit vhodný podnikatelský plán, který bude navazovat na navržený obchodní model a provést jeho revizi a případně jej upravit.



## Literatura

- [1] [Silueta člověka]. In: cleanpng.com [online]. USA: Cleanpng.com, 2021 [cit. 2022-05-10]. Dostupné z: <https://www.cleanpng.com/png-transparency-man-silhouette-person-adobe-photoshop-7354561/>.
- [2] [Obrázek slevy]. In: kindpng.com [online]. USA: Shenzhen BestAI Internet, 2019 [cit. 2022-05-10]. Dostupné z: <https://www.kindpng.com/free/discount-icon/>.
- [3] [Obrázek odměny]. In: adobe.com [online]. USA: Adobe, 2022 [cit. 2022-05-10]. Dostupné z: <https://stock.adobe.com/cz/search/images?k=trophy>.
- [4] [Obrázek účtenky]. In: thenounproject.com [online]. USA: Noun Project, 2022 [cit. 2022-05-10]. Dostupné z: <https://thenounproject.com/icon/receipt-37120/>.
- [5] [Obrázek kreditů]. In: pinclipart.com [online]. USA: PinClipart.com, 2018 [cit. 2022-05-10]. Dostupné z: <https://www.pinclipart.com/maxpin/ibhTmmJ/>.
- [6] [Obrázek události]. In: icon-library.com [online]. USA: Free Icons Library, 2019 [cit. 2022-05-10]. Dostupné z: <https://icon-library.com/icon/event-icon-27.html>.
- [7] [Obrázek šipky]. In: thenounproject.com [online]. USA: Noun Project, 2022 [cit. 2022-05-10]. Dostupné z: <https://thenounproject.com/icon/long-arrow-1337191/>.
- [8] [Obrázek redux]. In: dd.engineering.com [online]. USA: DD.ENGINEERING, 2020 [cit. 2022-05-10]. Dostupné z: <https://dd.engineering/blog/a-practical-introduction-to-using-redux-with-react>.
- [9] [Obrázek mern]. In: mongodb.com [online]. New York: MongoDB, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.mongodb.com/mern-stack>.

- [10] ŠALAGA Michal. *Návrh a tvorba backend strany platformy pro podporu gastronomie* [bakalářská práce, draft]. 2022.
- [11] *Rozdíl mezi funkčními a nefunkčními požadavky* [online]. USA: Strehphonsays, 2022 [cit. 2022-05-10]. Dostupné z: <https://cs.strehphonsays.com/functional-and-non-functional-requirements-3325>.
- [12] *Lekce 2 - UML - Use Case Diagram* [online]. Česká republika: itnetwork.cz, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>.
- [13] *Mapování procesu* [online]. Česká republika: Lean Six Sigma, 2022 [cit. 2022-05-10]. Dostupné z: <https://lean6sigma.cz/mapovani-procesu/>.
- [14] *Figma* [online]. USA: Figma, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.figma.com/>.
- [15] *Adobe XD* [online]. USA: Adobe, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.adobe.com/cz/products/xd.html>.
- [16] *Prototyping 101: The Difference between Low-Fidelity and High-Fidelity Prototypes and When to Use Each* [online]. USA: Adobe, 2022 [cit. 2022-05-10]. Dostupné z: <https://blog.adobe.com/en/publish/2017/11/29/prototyping-difference-low-fidelity-high-fidelity-prototypes-use>.
- [17] *React* [online]. USA: Meta Platforms, 2022 [cit. 2022-05-10]. Dostupné z: <https://reactjs.org/>.
- [18] *React Native* [online]. USA: Meta Platforms, 2022 [cit. 2022-05-10]. Dostupné z: <https://reactnative.dev/>.
- [19] *Angular* [online]. USA: Google, 2022 [cit. 2022-05-10]. Dostupné z: <https://angular.io/>.
- [20] *Ionic* [online]. USA: Ionic, 2022 [cit. 2022-05-10]. Dostupné z: <https://ionicframework.com/>.
- [21] *Angular vs React 2022 : Which JS Framework your Project Requires?* [online]. Česká republika: Simform, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.simform.com/blog/angular-vs-react/>.
- [22] *Is native, React Native, Flutter or PWA right for you?* [online]. USA: STRV, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.strv.com/blog/is-native-react-native-flutter-or-pwa-right-for-you-engineering-business>.
- [23] *Redux Toolkit* [online]. USA: Dan Abramov, 2022 [cit. 2022-05-10]. Dostupné z: <https://redux-toolkit.js.org/>.

- [24] *Recoil* [online]. USA: Facebook, 2022 [cit. 2022-05-10]. Dostupné z: <https://recoiljs.org/>.
- [25] *MERN stack* [online]. New York: MongoDB, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.mongodb.com/mern-stack>.
- [26] *REST API* [online]. Česká republika: itnetwork.cz, 2022 [cit. 2022-05-10]. Dostupné z: <https://www.itnetwork.cz/programovani/nezarazene/stoparuv-pruvodce-rest-api/>.
- [27] *Socket.io* [online]. USA: Socket.IO, 2022 [cit. 2022-05-10]. Dostupné z: <https://socket.io>.
- [28] *OAuth 2.0* [online]. USA: OAuth 2.0, 2022 [cit. 2022-05-10]. Dostupné z: <https://oauth.net/2/>.
- [29] *SWOT analýza* [online]. USA: ManagementMania.com 2.0, 2016 [cit. 2022-05-10]. Dostupné z: <https://managementmania.com/cs/swot-analyza>.
- [30] *Obchodní model* [online]. USA: ManagementMania.com 2.0, 2016 [cit. 2022-05-10]. Dostupné z: <https://managementmania.com/cs/obchodni-model-business-model>.

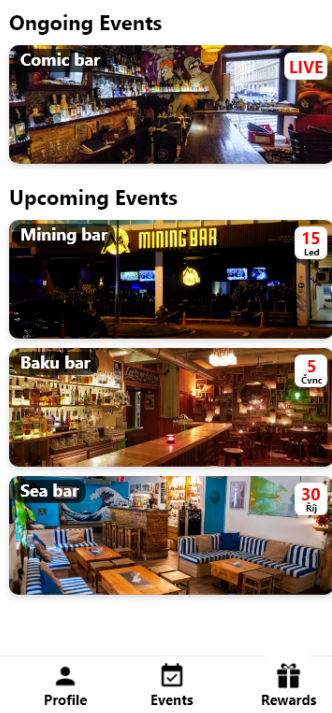


# Příloha A

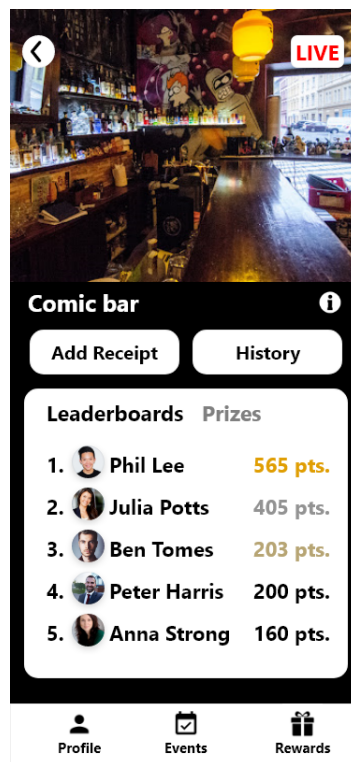
## Low-fidelity prototypy

Prototypy zobrazené v následující části obsahují pro ilustrační účely obrázky získané z platformy Google. V případě, že bude nutné prototypy využít při reálném provozu platformy, budou nahrazeny obrázky vlastními.

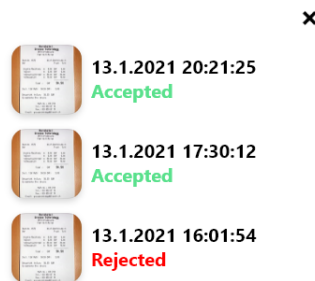
### A.1 Low-fidelity prototypy mobilní aplikace



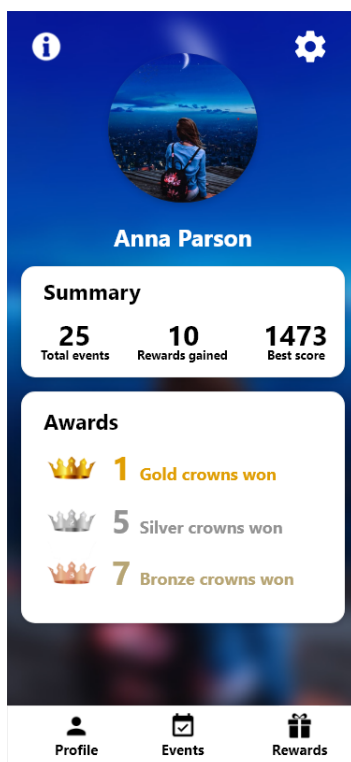
Obrázek A.1: Prototyp úvodní stránky s událostmi



Obrázek A.2: Prototyp stránky s detailem události



Obrázek A.3: Prototyp stránky s účenkami



Obrázek A.4: Prototyp profilové stránky

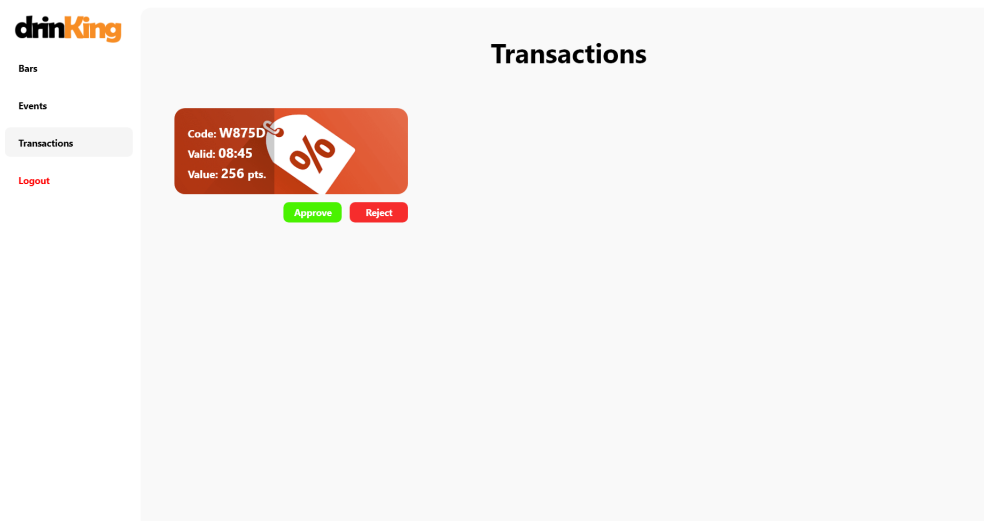


Obrázek A.5: Prototyp stránky s odměnami



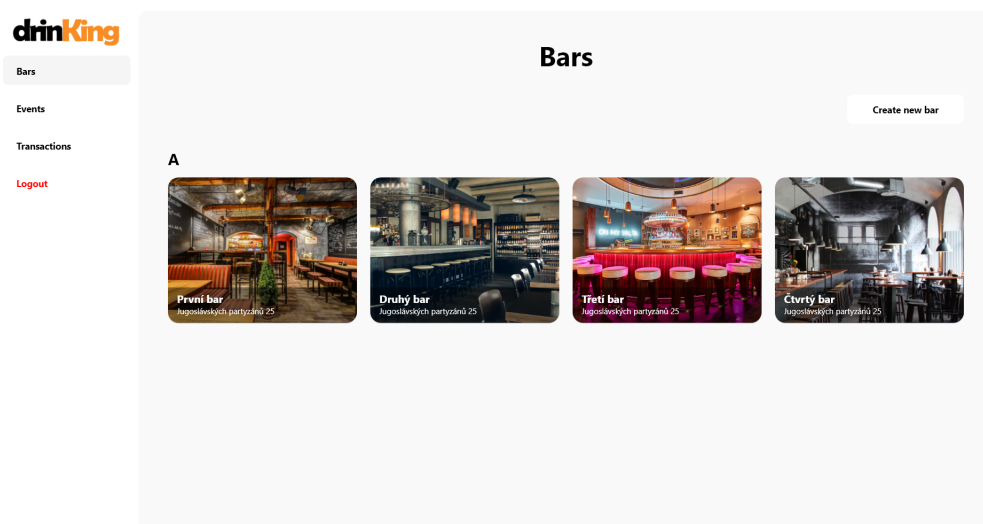
Obrázek A.6: Prototyp stránky s aktivovanou odměnou

## A.2 Low-fidelity prototypy webové aplikace

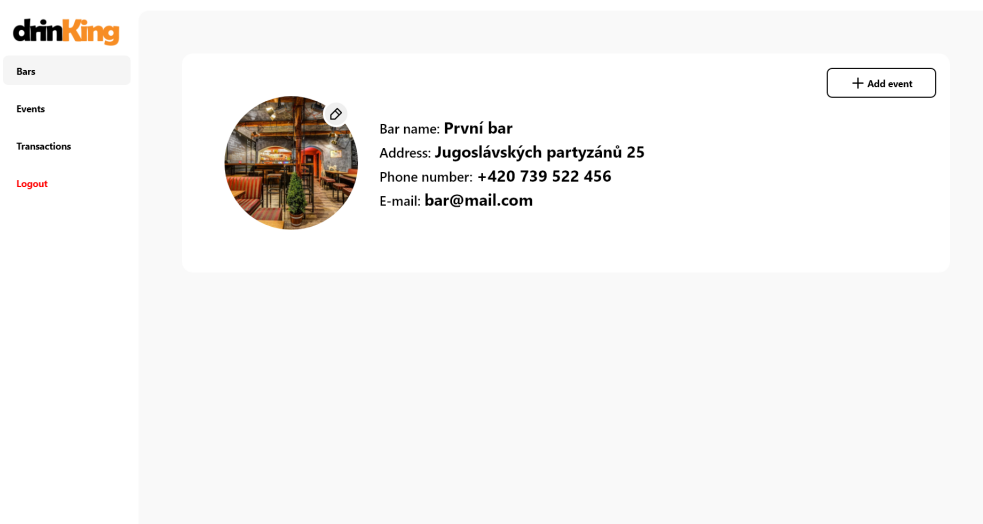


Obrázek A.7: Prototyp úvodní stránky s transakcemi

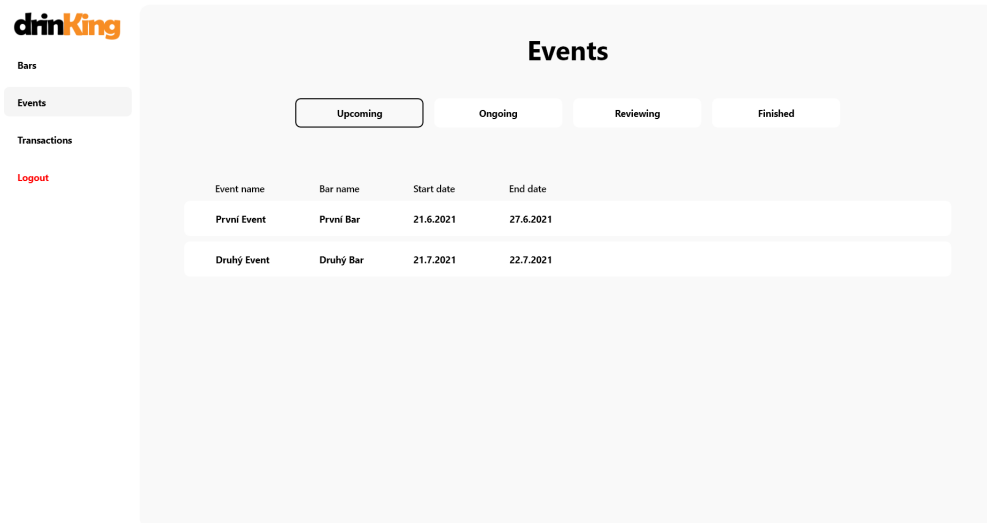




Obrázek A.8: Prototyp stránky s podniky



Obrázek A.9: Prototyp stránky s detailem podniku



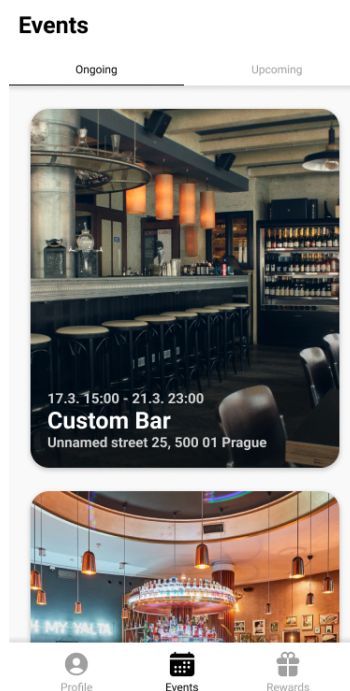
Obrázek A.10: Prototyp stránky s událostmi

## Příloha B

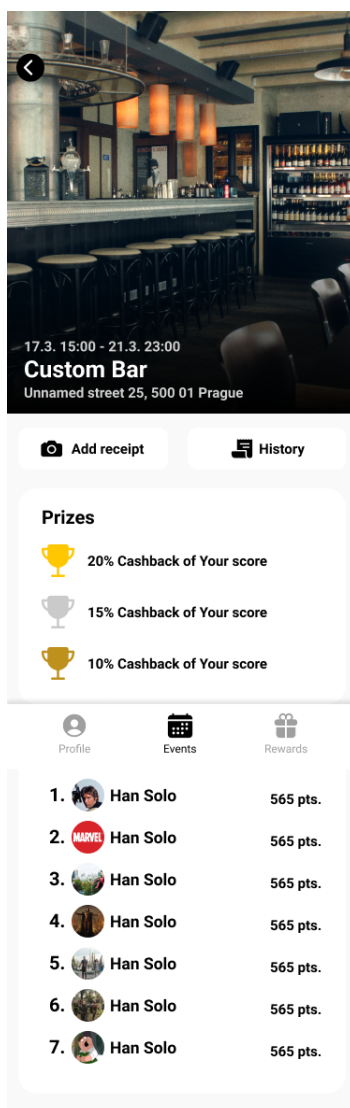
### High-fidelity prototypy

Prototypy zobrazené v následující části obsahují pro ilustrační účely obrázky získané z platformy Google. V případě, že bude nutné prototypy využít při reálném provozu platformy, budou nahrazeny obrázky vlastními.

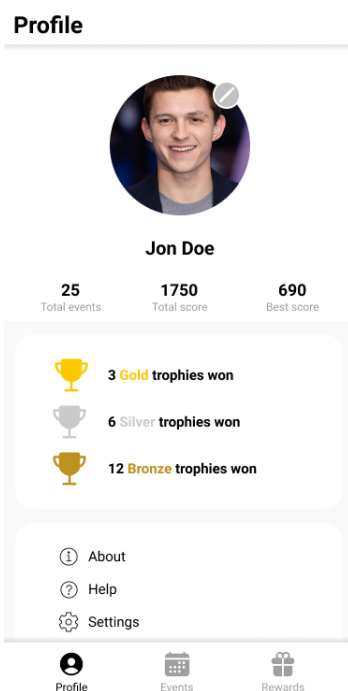
#### B.1 High-fidelity prototypy mobilní aplikace



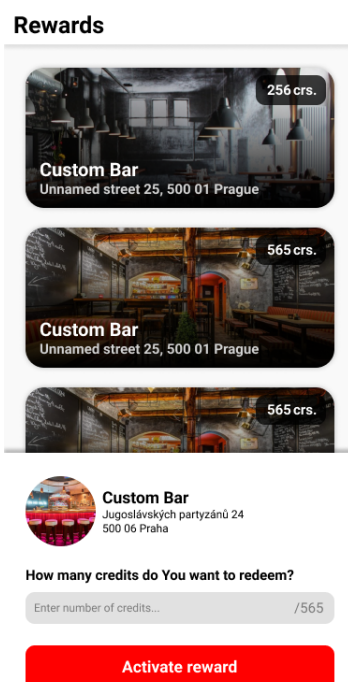
Obrázek B.1: Prototyp úvodní stránky s událostmi



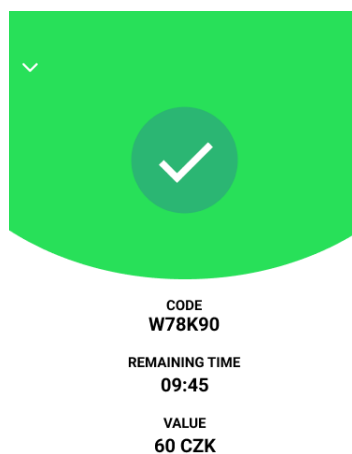
Obrázek B.2: Prototyp stránky s detailem události



Obrázek B.3: Prototyp profilové stránky



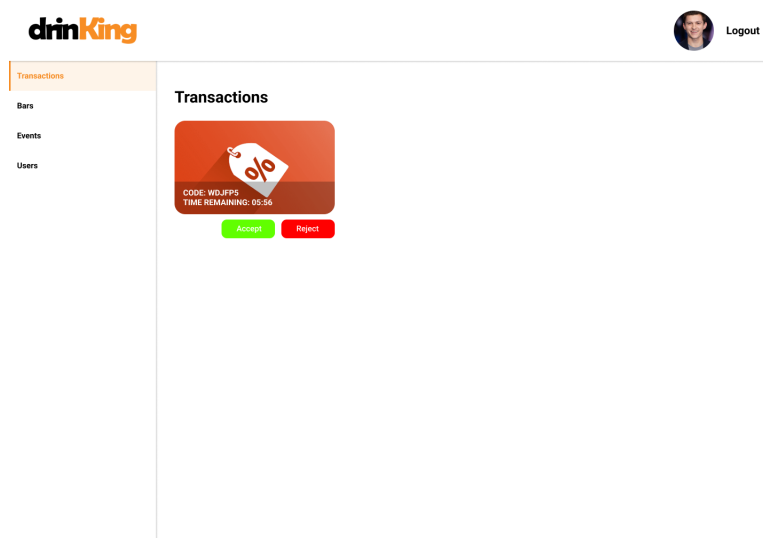
Obrázek B.4: Prototyp stránky s odměnami



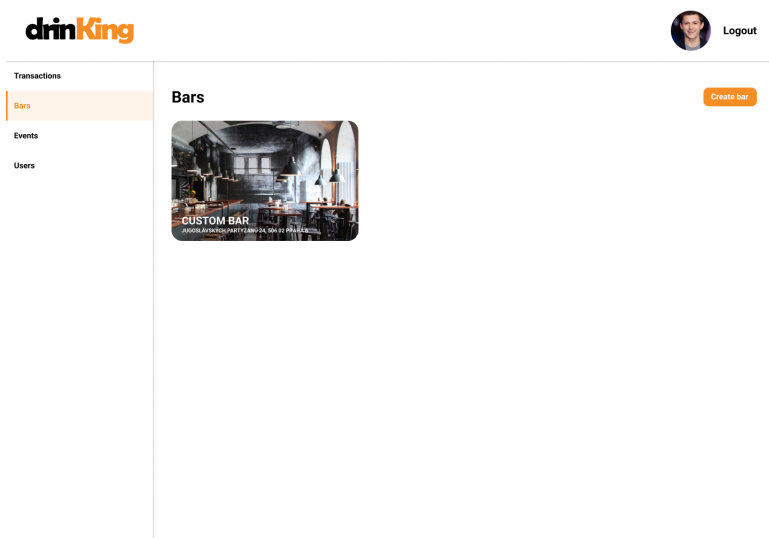
Cancel transaction

Obrázek B.5: Prototyp stránky s aktivovanou odměnou

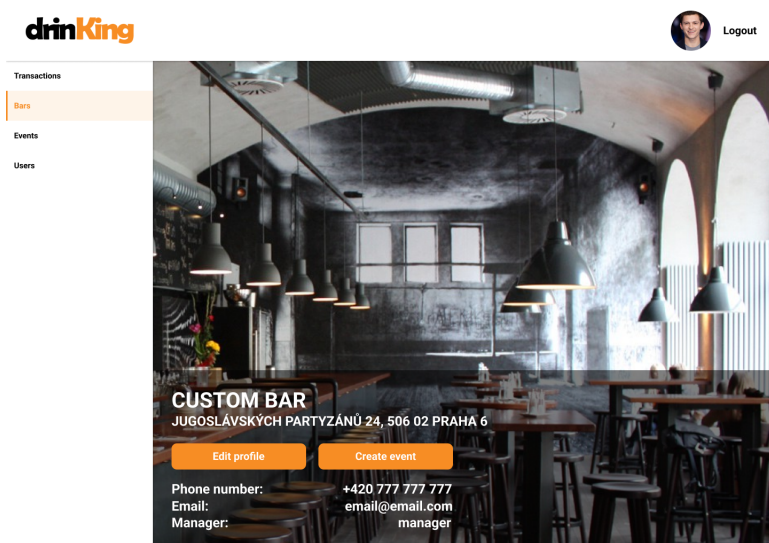
## B.2 High-fidelity prototypy webové aplikace



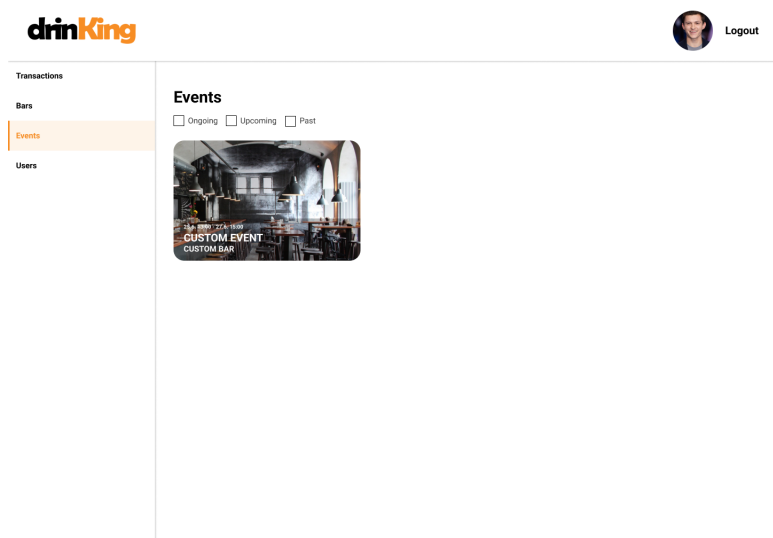
Obrázek B.6: Prototyp úvodní stránky s transakcemi



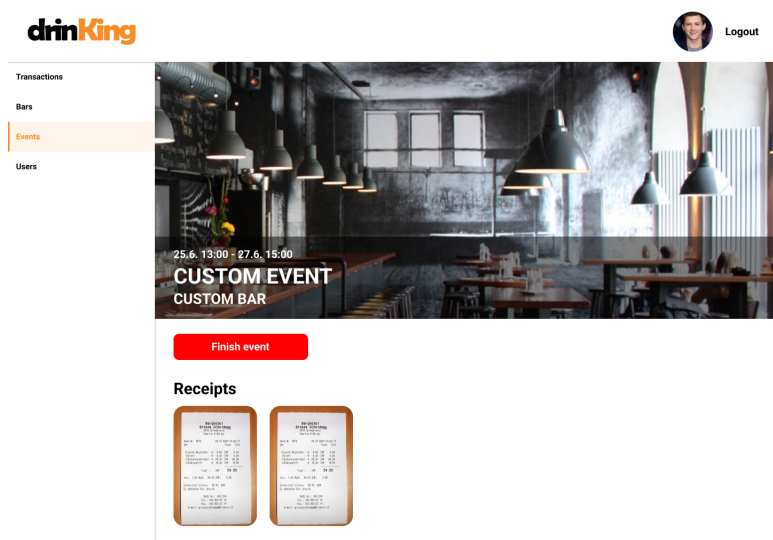
Obrázek B.7: Prototyp stránky s podniky



Obrázek B.8: Prototyp stránky s detailem podniku



Obrázek B.9: Prototyp stránky s událostmi



Obrázek B.10: Prototyp stránky s detailem události





## Příloha C

### Seznam použitých zkratk

- **API** Application Programming Interface
- **CSS** Cascading Style Sheets
- **HTML** Hypertext Markup Language
- **HTTP** Hypertext Transfer Protocol
- **JSX** JavaScript Extensible Markup Language
- **MVC** Model-view-controller
- **MVVM** Model–view–viewmodel
- **REST** Representational State Transfer
- **UI** User Interface
- **URI** Uniform Resource Identifier
- **XML** Extensible Markup Language



## Příloha D

### Obsah přiloženého komprimovaného souboru

- **mobile\_app\_source\_code** - zdrojový kód mobilní aplikace
- **web\_app\_source\_code** - zdrojový kód webové aplikace
- **obrázky** - obrázky uvedené v práci
  - prototypy - návrhy obrazovek
  - diagramy - diagramy použité při návrhu aplikací
  - screenshoty - snímky obrazovek aplikací třetích stran
  - ostatní - další obrázky uvedené v práci
- **thesis.pdf** - text práce ve formátu PDF
- **thesis\_source.zip** - zdrojové kódy práce ve formátu LaTeX