



ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	ElateMe: Analýza a implementace internetových plateb
Student:	Luka Lukašević
Vedoucí:	Ing. Petr Pauš, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2018/19

Pokyny pro vypracování

Cílem práce je analýza významných platebních online metod. Vybrané metody implementujte do aplikace ElateMe.

Analyzujte:

- technické a implementační vlastnosti plateb kartou, Payment Request API, QR kódem, Paypalem a bitcoiny a jejich transakční náklady,
- funkční a nefunkční požadavky pro platby, použijte FURPS+,
- požadavky popište vhodnými UML diagramy.

Navrhněte:

- platbu interními kredity v aplikaci,
- platformě specifický a nesespecifický model vyvíjených součástí a subsystémů.

Implementujte:

- platbu interními kredity, bitcoin, paypal, Payment Request API, QR kód,
- grafické uživatelské rozhraní dle dodaného návrhu.

Otestujte:

- vytvořte unit testy,
- proveďte uživatelské testování,
- ověřte zabezpečení plateb.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 8. února 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLÓGIÍ
ČVUT V PRAZE**

Bakalářská práce

Analýza a implementace internetových plateb v rámci aplikace ElateMe

Luka Lukašević

Katedra softwarového inženýrství

Vedoucí práce: Ing. Petr Pauš, Ph.D.

10. května 2018

Poděkování

Chtěl bych poděkovat panu Michalu Maněnovi, za navržení tématu bakalářské práce a za cenné rady při vypracovávání práce. Dále bych chtěl poděkovat vedoucímu práce panu Ing. Petru Paušovi, Ph.D., za pomoc se strukturou práce a za vedení při jejím vypracování.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 10. května 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Luka Lukašević. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Lukašević, Luka. *Analýza a implementace internetových plateb v rámci aplikace ElateMe*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Bakalářská práce se zabývá analýzou a implementací vybraných internetových plateb do aplikace ElateMe. V práci je provedena analýza jednotlivých platebních metod a následně jejich návrh a implementace. Jednotlivé metody jsou dopodrobna popsány a je také popsáno, jak je třeba postupovat pro zprovoznění dané metody v internetové aplikaci.

Klíčová slova ElateMe, platby na internetu, PayPal, QR kód, platební karta, bitcoin, Google Pay

Abstract

This bachelor thesis deals with analysis and implementation of selected payment methods in ElateMe. Each payment method is analyzed and subsequently, its design and implementation are described. All individual methods are described in detail and the right procedure how to make these methods function correctly in an internet application is also described.

Keywords ElateMe, internet payments, PayPal, QR code, credit card, bitcoin, Google Pay

Obsah

1	Úvod	1
1.1	Cíl práce	2
1.2	ElateMe	2
2	Analýza	5
2.1	Použité platební metody	5
2.2	Google Pay	6
2.3	Platba platební kartou	11
2.4	Bitcoin Payment	18
2.5	QR Code payment	28
2.6	PayPal poháněný systémem Braintree	29
2.7	Diagram případů užití	33
3	Návrh	35
3.1	Návrh platby QR kódem	37
3.2	Návrh platby PayPal s využitím systému Braintree	42
3.3	Návrh platby platební kartou	45
3.4	Návrh platby pomocí bitcoinu	47
3.5	Návrh platby pomocí Google Pay	50
4	Implementace	53
4.1	Implementace platby pomocí QR kódu	53
4.2	Implementace platby kreditní kartou	55
4.3	Implementace platby pomocí bitcoinů	56
4.4	Implementace platby PayPal poháněná systémem Braintree	58
4.5	Implementace platby Google Pay	60
5	Testování	63
5.1	Testování platební metody PayPal	64

Závěr	69
Literatura	71
A Seznam použitých zkratk	75
B Návod na vyzkoušení platebních metod v aplikaci ElateMe	77
C Obsah přiloženého CD	79

Seznam obrázků

2.1	Stručný proces Payment Requestu. [1]	8
2.2	Uživatelské rozhraní Payment Request API na mobilní platformě.[1]	9
2.3	1. fáze platebního procesu platby kartou na internetu s názvem Povolení. [2]	13
2.4	2. fáze platebního procesu platby kartou na internetu s názvem Dávkování. [2]	14
2.5	3. fáze platebního procesu platby kartou na internetu s názvem Zúčtování. [2]	14
2.6	4. fáze platebního procesu platby kartou na internetu s názvem Financování. [2]	15
2.7	Detail informací k zaplacení pomocí bitcoinů od společnosti BitcoinPay. [3]	21
2.8	Krok 1 při platebním procesu, který dodává společnost BitPay. [4]	23
2.9	Krok 2 při platebním procesu, který dodává společnost BitPay. [4]	23
2.10	Krok 3 při platebním procesu, který dodává společnost BitPay. [4]	24
2.11	Sekvenční diagram platby pomocí bitcoinů [5].	27
2.12	Průběh vytvoření platební transakce s Braintree [6].	32
2.13	Diagram případů užití při interakci uživatele s platebními metodami.	33
3.1	Krok 1 při procesu přispívání - výběr přispívané částky.	36
3.2	Krok 2 při procesu přispívání – zvolení platební metody.	36
3.3	Krok 3 při procesu přispívání – poděkování zákazníkovi za platbu.	37
3.4	Detail vygenerovaného QR kódu k přání v aplikaci ElateMe.	40
3.5	Diagram aktivit klientské části platby pomocí QR kódu.	41
3.6	Diagram aktivit serverové části platby pomocí QR kódu.	42
3.7	Detail vyskakovacího okénka pro platbu PayPalem.[7]	44
3.8	Diagram aktivit platby PayPal.	45
3.9	Detail platební brány dodávané společností First Data pro účely Fio banky.	46
3.10	Diagram aktivit pro platbu kreditní kartou[8].	47

3.11	Diagram aktivit pro platbu pomocí bitcoinů.	50
3.12	Detail zobrazení Google Pay metody pomocí payment requestu. [9]	51
3.13	Diagram aktivit pro platbu Google Pay za použití systému Braintree.	52
5.1	Počáteční krok testování, detail přání.	65

Seznam tabulek

3.1	Seznam použitých parametrů pro vytvoření vyhovujícího QR kódu.[10]	39
4.1	Seznam atributů pro komponentu <QRCode />. [11]	54

Úvod

V dnešní době stále přibývá plateb na internetu. I přes to, že co se statistik týče, stále převládá typ platby zboží na internetu dobírkou, je platba na internetu na vzestupu a brzy zcela ovládne způsob plateb za zboží prodávané na internetu. S tím přichází i spousta možností jak na internetu platit. Nejedná se pouze o přímočarou platbu kreditní kartou tak, jak to všichni dělají v kamenných obchodech. I když je způsob platby kreditní kartou nejpoblárnější, vznikají zde nové platební metody, které mají především usnadnit a zabezpečit průběh platby na internetu. Ať už se jedná o elektronické peněženky, které si ukládají informace o kreditní kartě uživatele a zrychlují tak proces platby, jako je například PayPal. Nebo se objevují nové virtuální měny jako bitcoin, které fungují jako virtuální peníze a dá se s nimi na internetu platit. Metod a způsobů, jak na internetu platit, přibývá, jak ale poznat, které doopravdy stojí za to ve vaší internetové aplikaci používat a které pouze vystupují jako výhodné řešení, ale ve výsledku tomu tak vůbec není? V rámci aplikace ElateMe, na které jsem se podílel při vývoji v čtvrtém a pátém semestru na Fakultě informačních technologií, jsem dostal za úkol implementovat vybrané platby na internetu. Dostal jsem za úkol platby nejen implementovat, ale také se podívat blíže na platební procesory, které dané platby umožňují provozovat, a vybrat z nich ty nevhodnější pro aplikaci, jako je ElateMe. Nejedná se pouze o to vybrat způsob platby, který je implementačně nejjednodušší, je třeba uvážit zde i faktory, jako je podpora platebních metod i v zahraničí, jak jsou služby fungující jako platební procesory výhodné, jak jsou platební metody bezpečné, abychom co nejvíce snížili riziko napadnutí a mnoho dalšího. Tím vším se moje bakalářská práce bude zabývat a budu se snažit o to, aby kdokoliv kdo bude implementovat internetové platební metody do své internetové aplikace, po přečtení této práce věděl, jakým směrem se vydat.

1.1 Cíl práce

Cílem této bakalářské práce je zanalyzovat a implementovat vybrané internetové platby do aplikace ElateMe. Především se budu soustředit na správnou a podrobnou analýzu jednotlivých plateb. Po správné analýze poté stačí pouze využít dokumentace, kterou dodávají platební procesory k integraci platebních metod s internetovými aplikacemi. Je třeba vybrat správné platební procesory pro jednotlivé platby, neboť právě platební procesory jsou základním kamenem pro implementaci internetových plateb. Existuje více platebních procesorů a je tedy důležité vybrat takový, který nejvíce vyhovuje požadavkům aplikace ElateMe. Dále bude mým úkolem správně využít poznatky z analýzy a platby správně implementovat pro použití v aplikaci ElateMe. Na konci práce jednu vybranou platební metodu otestuji, kdy provedu testování od rozhodnutí uživatele pro platbu tak po připsání finančního obnosu na účet ElateMe.

1.2 ElateMe

Společnost ElateMe s.r.o. je startup založený v červenci roku 2017 panem Matějem Macháčkem a Michalem Maněnou. Na vývoji webové a mobilní aplikace ElateMe.com pracují její zakladatelé od února roku 2016. Tato aplikace poskytuje inovativní řešení v situacích, kdy dojde k události, při níž se očekává příjem daru a nebo když chcete darovat vy něco někomu. Takové události jsou nejčastěji narozeniny, Vánoce, svatby a spousty dalších událostí, při kterých dochází k darování darů. Uživatelé aplikace ElateMe.com mohou vytvářet přání a sami přispívat na přání svých přátel.

Společnost zakoupila autorská práva na aplikaci od studentů ČVUT, kteří se na vývoji jak mobilní tak webové aplikace podíleli v rámci předmětu Softwarový týmový projekt, který je vyučován na Fakultě informačních technologií na ČVUT a v rámci svých bakalářských prací, které studenti dovedli do zdárného konce a obhájili je před komisí u státních závěrečných zkoušek.

1.2.1 Co je to Wowiee?

Wowiee je oficiální název aplikace ElateMe.com. Jelikož ale zakladatelé aplikace chtějí proniknout i na zahraniční trh, zjistili, že název ElateMe je těžko zapamatovatelný a ne tak zvučný co se marketingového hlediska týče. Na druhou stranu, nebylo možné použít Wowiee jako název domény, jelikož slovo wowiee má v anglickém jazyce slangový význam pro výraz “wow“ a tím pádem je cena domény značně vyšší než za doménu ElateMe. Z toho důvodu je název webové služby Wowiee a název domény stejný jako jméno společnosti ElateMe.

Uživatel se prostřednictvím Facebooku přihlásí do aplikace a na úvodní stránce vidí feed nejnověji vytvořených přání od svých přátel. Na každé přání je možno kliknout a tím se zobrazí detail tohoto přání. Na detailu přání je vidět

požadovaná věc, částka kterou je potřeba vybrat, částka kterou se už podařilo vybrat, popis k přání a případné komentáře ostatních přátel, či uživatelů aplikace Wowee. Na úvodní stránce feedu jsou zobrazeny i nejbližší narozeniny přátel, což může podnítit aktivnější přispívání na přání uživatele, který vytvořil přání v rámci svých narozenin. Každý uživatel má kruh svých přátel, kteří mají možnost vidět a přispívat na přání vytvořené uživatelem. Uživatel ale může právo přátel na přispívání a prohlížení jeho vytvořených přání odebrat a stejně tak může tato práva přidat uživatelům, které nemá v okruhu přátel. Žádost o povolení prohlížení přání a na jeho přispívání je možné uskutečnit na detailu přání, pokud je tento detail neúplný, to znamená, že je očividné že uživatel nemá potřebná práva. Pokud se uživatel rozhodne pro darování příspěvku na přání, tak tato akce probíhá ve čtyřech na sebe navazujících krocích. Prvním krokem je výběr částky, kterou chceme na dané přání přispět. Po zvolení částky se uživatel dostane k druhému kroku, kde si zvolí způsob platby z vybraných platebních metod. Po vybrání metody dojde ke zpracování samotné platby, které probíhá na základě vybrané metody. Posledním krokem je poděkování za příspěvek k přání.[12]

Analýza

Analýza je nejdůležitější částí této práce, jelikož na jejím základě se budou poté odvíjet části, jako je návrh a implementace. Seznámím čtenáře s platebními metodami, které jsme se rozhodli pro aplikaci ElateMe implementovat. Dále zde popíši co je vše potřeba pro správný průběh plateb, a jak jsme při výběru správných platebních procesorů postupovali.

2.1 Použité platební metody

Pro realizaci přispívání peněz na přání bylo v rámci aplikace ElateMe zapotřebí realizovat funkční platební uživatelské rozhraní. Tato část aplikace je zřejmě nejdůležitější částí celé aplikace, jelikož příspěvky uživatelů na přání jsou hlavní myšlenkou a hlavní funkcí aplikace.

Proto bylo velmi důležité vybrat vhodné platební metody, aby průběh plateb byl co nejplynulejší, nejbezpečnější, nejrychlejší a celkově uživatelsky nejvíce vyhovující. Aplikace ElateMe nemá mít pouze tuzemský rozsah, tudíž z toho důvodu se kromě výše zmíněných kritérií, které by platby zvolené pro platební rozhraní měly splňovat, muselo ještě přihlížet k různým technickým a právním omezením pro mimorepublikové platby. Pro představu uvedu příklad, že pokud by se pro vývoj platby platební kartou mělo použít REST API společnosti PayPal tak by to nebylo možné, jelikož PayPal API již nadále nepodporuje integrace s platebními kartami. Vyvinuli ale vlastní produkt s názvem Braintree, který poskytuje potřebné API pro integraci s platebními kartami. Tato omezení budou blíže popsány v kapitole Technologické a tržní omezení.

Po zvážení všech kritérií byly pro realizaci platebních transakcí zvoleny následující platební metody a standardy,

- Platební karta,
- Google Pay,

- Bitcoin Payment,
- QR Payment,
- PayPal poháněný Braintree.

Tyto vybrané metody budou stručněji popsány v následujících kapitolách a to jak po analytické stránce tak později v kapitole Implementace i po implementační stránce.

2.2 Google Pay

Google pay je nová platební metoda vytvořená společností Google, kterou je možno platit v kamenných obchodech a v mobilních aplikacích pomocí Android mobilního zařízení, tabletu nebo chytrých hodinek. V lednu roku 2018 se platební metoda Android Pay a služba Google Wallet sjednotili do jedné platební metody nazvané Google Pay. Android Pay změnil značku i název a používá nyní nový název Google Pay. Služba Google Pay si zachová všechny funkčnosti služby Android Pay a doplní je o funkčnosti služby Google Wallet. Tato služba funguje pouze na Android zařízeních a na internetu se dá použít pouze v prohlížeči Android Chrome od společnosti Google. Při použití platby Google Pay v kamenném obchodě služba využívá technologii NFC k přenosu platebních informací pro platbu a informace o kreditní kartě jsou uloženy v telefonu zákazníka. NFC je zkratka pro Near Field Communication technologii což je soubor protokolů, které umožní komunikovat dvěma elektronickým zařízením na vzdálenost 4 cm [13]. Při použití Google Pay na internetu je využíváno standardu Payment Request, který uchovává informace o kreditní kartě v prohlížeči. Jelikož je naše aplikace webová, budeme používat Google Pay s využitím Payment Requestu.[14]

2.2.1 Payment Request API

Payment Request API je novým standardem pro platby na internetu, který v současnosti vyvíjí mezinárodní konsorcium W3C, které má na starost vývoj webových standardů.

Tento nový standard má zvýšit bezpečnost internetových plateb a především usnadnit jejich průběh. Hlavním úkolem tohoto nového standardu je urychlit a usnadnit platební proces pro uživatele. To znamená, že místo klasických 3 až 4 kroků a více než dvou kliknutí, kdy chceme něco na internetu zaplatit kartou nebo jiným platebním způsobem, který daná stránka podporuje, nám Payment Request API toto umožní v jednom kroku s pouhými pár kliknutími.

Payment Request API je navržen tak, aby byl tak zvaně vendor-agnostic, což znamená, že nevyžaduje použití konkrétního platebního systému. Je tedy navržen spíše obecně, abychom ho mohli integrovat do většiny už používaných

plateb na internetu. Nejedná se o nový způsob platby, ani se přímo netýká platebních procesorů. Jedná se spíše o způsob, jak dodat platbu uživatele a informace o dopravě obchodníkovi s následujícími cíli:

- Nechá vystupovat konkrétní internetový prohlížeč jako prostředníka mezi obchodníkem, uživatelem a platební metodou.
- Usnadňuje a standardizuje tok průběhu platební komunikace co nejvíce to jde.
- Bezproblémově podporuje rozdílné bezpečné platební metody.
- Pracuje bezproblémově na jakémkoliv prohlížeči, zařízení nebo platformě mobilní či jiné.

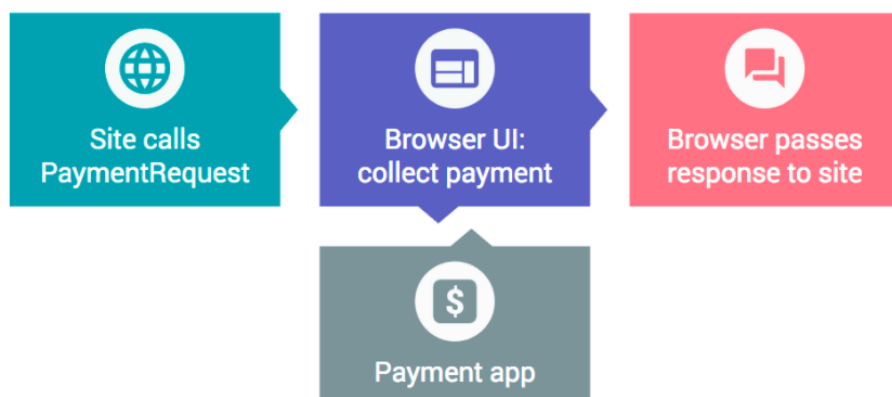
Payment Request API je otevřený standard, podporovaný vícero internetovými prohlížeči, který zaměřuje tradiční, zdlouhavý průběh platby na internetu tím, že umožňuje obchodníkovi požadovat a přijímat jakoukoliv platbu v rámci jednoho API volání. API rozhraní umožňuje webové stránce si vyměňovat informace s uživatelským agentem, který vystupuje jako vyskakovací okno v prohlížeči s informacemi potřebnými k dokončení platby, zatímco uživatel poskytuje vstup před schválením nebo zamítnutím žádosti o platbu.

Jak už bylo zmíněno, tak největší inovací se kterou přichází Payment Request API je rychlost a jednoduchost z pohledu uživatele. To je způsobeno především tím, že zde prohlížeč vystupuje jako prostředník. Všechny informace potřebné k rychlému zaplacení tak mohou být uloženy přímo v prohlížeči. Tím pádem může uživatel pouze platbu potvrdit a provést, vše pouze s jedním kliknutím. Pokud uživatel použije na určitém prohlížeči platbu poprvé, tak bude vyzván k doplnění potřebných informací k provedení platby jako jsou především informace o kartě a informace o způsobu dopravy, prohlížeč si informace zapamatuje a do budoucna bude platba probíhat pouze kontrolou a jedním kliknutím. [1]

2.2.2 Proces platební transakce Payment Requestu

Používáním Payment Request API je proces přenosu peněz vytvořen co nejsouvisleji, jak je to jen možné, a to pro obě dvě strany, jak pro uživatele tak pro obchodníka. Platební proces jako takový začíná ve chvíli, kdy uživatel zvolí typ platby pomocí payment requestu. Jednoduše spustí akci kliknutím na tlačítko "Google Pay". Poté co uživatel klikne na tlačítko, tak se na webové stránce obchodníka vytvoří paymentRequest a předá internetovému prohlížeči potřebné informace k provedení platby. Tyto informace jsou kupříkladu částka, která bude odečtena z účtu zákazníka a přičtena na účet obchodníka po provedení platby, jaká měna byla zvolena a jaké platební metody webová stránka dovoluje. Internetový prohlížeč poté zkontroluje kompatibilitu mezi vybranými platebními metodami, které bude moct zákazník použít při payment requestu

2. ANALÝZA



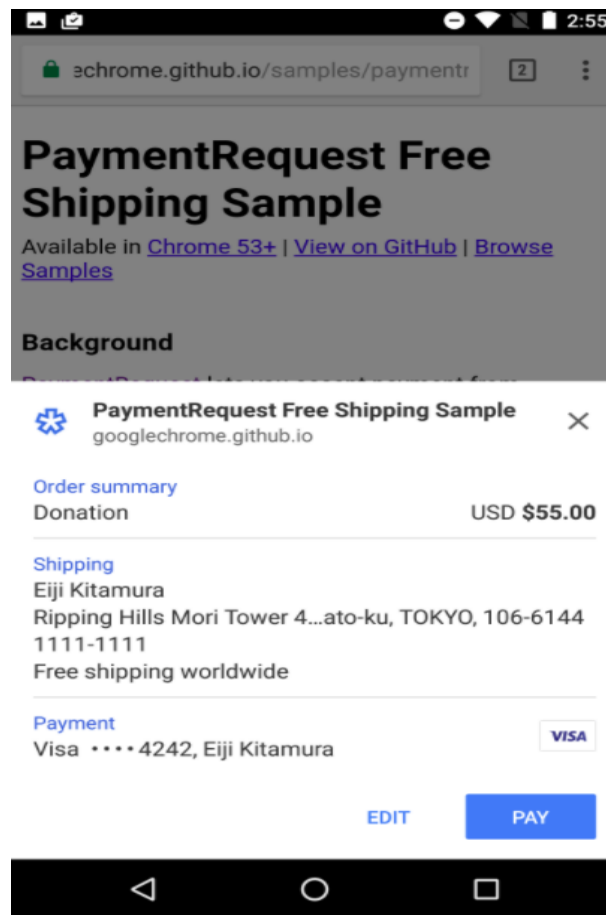
Obrázek 2.1: Stručný proces Payment Requestu. [1]

a mezi metodami, které má zákazník nainstalované na příslušném zařízení. Například, pokud obchodník umožní zákazníkovi pomocí payment requestu zaplatit platební metodou Google Pay, tak by zákazník měl používat zařízení s operačním systémem Android.

Prohlížeč poté zobrazí zákazníkovi vyskakovací formulář, který představuje uživatelské rohraní pro zkontrolování platebních informací a údajů o platbě. Pokud zákazník už někdy v minulosti použil Payment Request API v prohlížeči ze zařízení, ze kterého chce platbu uskutečnit, pak pouze vybere platební metodu, pomocí které chce akci dokončit, a zadá autorizační údaje, což je například v případě zvolení platby kreditní kartou třímístný CVC kód. Pokud zákazník používá Payment Request API na daném prohlížeči, z daného zařízení poprvé, bude vyzván k vyplnění doručovacích údajů, informací o kreditní kartě a případně osobních informací, jako jsou jméno a email.

Jako platební metoda zde může být jak přímočará platba kreditní kartou, která je prohlížečem uložena nebo uložena bude a nebo aplikace třetí strany, která je vyvinuta speciálně pro doručení platebních informací na potřebnou webovou stránku. Poté co zákazník autorizuje platební transakci, jsou všechny údaje zpracovány na straně obchodníka pomocí Payment Request API. Dále proces pokračuje v z pohledu uživatele v několika sekundách, kdy se zákazníkovi zobrazí okénko s tím, zda byla platba přijata či nikoliv. Na straně procesu platby a zpracování informací je proces složitější.

Jak bylo výše zmíněno, tak druhou hlavní výhodou Payment Request API je bezpečnost. Při platbě dochází k výměně citlivých informací z pohledu zákazníka jako je číslo kreditní karty, datum expirace karty, CVC kód, jméno držitele či jeho kontaktní údaje. V nynější době jsou tyto informace, které jsou vyměňovány mezi zákazníkem, obchodníkem a bankou, velmi dobře chráněny pomocí bezpečnostních protokolů, aby nedošlo k útoku na průběh transakce



Obrázek 2.2: Uživatelské rozhraní Payment Request API na mobilní platformě.[1]

ze strany hackerů. Pořád jsou ale informace viditelné pro stranu obchodníka, kde je bezpečnost citlivých informací zákazníka založena pouze na důvěře k obchodníkovi. Jedním ze způsobů, jak citlivé informace zákazníka skrýt je princip tokenizace.

2.2.3 Princip tokenizace u Payment Requestu

Tokenizace se dá popsat jednoduše a složitěji. Jednoduchý popis je takový, že tokenizace je proces, při kterém dojde k záměně citlivých informací za náhodně generované znaky stejného formátu s tím, že token, který je vygenerován, nemá žádnou matematickou souvislost s daty, ze kterých vzniknul. Data, ze kterých token vzniknul, jsou uložena v cloudové webové službě, která token vygenerovala. Což je také jediná možnost, jak provést opačnou operaci zvanou de-tokenizace, kdy z tokenu získáme reálné informace. Složitější popis

tokenizace zabíhá více do hloubky a tato práce se jím nebude blíže zabývat.

Tento způsob ochrany dat je primárně používán při platebních procesech z toho důvodu, že jediné co, se posílá mezi bankami, platebními branami a webovými službami je právě vygenerovaný token. Tudíž případný útočník jediné co dokáže z komunikace odposlechnout je pouze token, ze kterého má velmi malou pravděpodobnost získat zakódované informace. Princip tokenizace používají především technologie Apple Pay, Android Pay, Samsung Pay a Google Pay. Payment Request funguje na stejném principu jako vyjmenované technologie s tím rozdílem, že místo aplikace, do které se uloží informace o kartě, kterou používají dané technologie, používá Payment Request internetový prohlížeč.

Zatímco zákazník zkontroluje platební informace v uživatelském rozhraní, které vygeneruje Payment Request API, a potvrdí platbu, dostane odpověď za pár sekund, zda transakce proběhla úspěšně či nikoliv. Za tím se ale odehrává hned několik akcí na straně platebního procesu. Když uživatel potvrdí platbu, Payment Request API zpracuje dodané informace a pošle požadavek o token do Googlu, Google se spojí s bankou zákazníka podle informací z karty a zažádá o vygenerování tokenu. Banka zkontroluje informace z karty, které jí předal Google, a na základě toho vygeneruje token, který si k příslušným datům uloží. Pošle token zpátky Googlu a Google vrátí token jako výsledek Payment Requestu. Obchodník tak jediné co má k dispozici, je token, který odešle na zvolenou platební bránu. Platební brána se spojí s bankou, která působí jako držitel karty nebo anglicky acquiring bank například držitel karty VISA. Banka, u které je karta uživatele registrována se spojí s bankou zákazníka a pošle token s dotazem, zda existuje v bance takto vygenerovaný token. Jestliže nastane shoda banka pošle kladnou odpověď do banky držitele karty, banka držitele pošle kladnou odpověď na platební bránu, kde se provede platba a vrátí kladnou odpověď obchodníkovi, že platba proběhla v pořádku a obchodník zobrazí akceptaci platby zákazníkovi. Jestliže nedojde v bance uživatele ke shodě tokenů, pošle se zpátky odpověď taktéž, v tomto případě však negativní a zákazník bude informován o neprovedení platby, kde v odpovědi banky bude udáno, z jakého důvodu nemohlo dojít k provedení platby.

2.2.4 Výběr platebního procesoru

Jelikož je Payment Request pouhým standardem, který nám usnadňuje platební proces a zaručuje anonymizaci dat, je třeba použít platební procesor, jenž informace o platbě odešle po síti a spojí banky obchodníka a zákazníka, aby mohlo dojít k přenosu peněz. Existuje velká spousta společností, jež dodávají funkční platební procesory na internetu. Mezi největší určitě patří Stripe, Global Payments, Go Shippo a nebo Braintree. Každá z těchto společností funguje na trhu už pár let a těší se z velké oblíbenosti zákazníků. Fungují jako platební procesory pro celou řadu platebních metod a ne jenom pro Google Pay. Jelikož je jednou z vybraných platebních metod, kterou budu analyzo-

vat a implementovat metoda PayPal, automaticky bude tato metoda používat jako platebního procesora Braintree, jelikož je Braintree dceřinou společností společnosti PayPal, což popíše blíže u samotného popisu metody. Braintree funguje jako platební procesor i pro Google Pay a dokonce poskytuje integraci pro samotný Payment Request. Rozhodli jsme se proto, že jako platebního procesora využijeme právě společnost Braintree. Především z důvodu, že už bude Braintree fungovat jako platební procesor pro platební metodu PayPal. Více o Braintree v kapitole o PayPal [2.6].

2.3 Platba platební kartou

Možnost platit na internetu platební kartou by neměla chybět v žádné webové aplikaci, kde dochází k platebním procesům. Je to nejvíce a nejdéle používaný typ platby na internetu, kterému důvěřuje většina zákazníků, a v dnešní době každý platební kartu vlastní a aktivně jí využívá při platbách na internetu. Je proto velmi důležité pro obchodníka zvolit nejlepší cestu pro přijímání plateb pomocí platební karty a zvolit správného platebního procesora, který platby zprostředkuje. Byznys, který společnost ElateMe chce provozovat, je založen na přijímání plateb a je proto kladen veliký důraz na správnou implementaci a vyřešení hladkého průběhu plateb. Na co chceme klást především důraz při implementaci platby platební kartou je bezpečnost.

- Více jak 1/3 zákazníků váhá nakupovat na internetu z bezpečnostních důvodů.
- Průměrný podnik, který funguje na internetu, ztrácí každoročně 5% z výnosu kvůli podvodům na internetu.
- 50% začínajících podniků se stanou obětí internetového podvodu v nějaké chvíli podnikatelského cyklu.[?]

Je proto velmi důležité využít správného platebního procesora, který nám zaručí co nejvyšší bezpečnost. Předtím ale než se dostanu k popisu vyhovujících platebních procesorů a metod, které se při implementaci platby platební kartou používají a dodržují, by každý obchodník, který provozuje platební procesy ve své aplikaci, měl vědět jak takový proces platební kartou obecně vypadá a k jakým operacím a pohybům na síti při tom dochází.

2.3.1 Pochopení procesu platby platební kartou na internetu

V této sekci vysvětlím proces platby platební kartou na internetu, aby čtenář dosáhl vyššího pochopení procesů a použitých termínů. Popíše proces od zadání platebních údajů uživatelem až po odečtení zaplacené částky ze zákaznickova účtu a připsání obnosu na obchodníka účet. Proces se týká sedmi entit,

2. ANALÝZA

kteřé budou později blíže popsány a jejich role v procesu budou vysvětleny a zobrazeny na obrázkovém diagramu pro lepší pochopení procesu.

Entity jsou následující.

- Zákazník,
- obchodník,
- platební brána,
- platební procesor,
- síť platebních karet,
- emitující banka (issuing bank),
- banka akceptující platební karty (acquiring bank).

Držitel karty neboli zákazník se rozhodne pro platbu platební kartou na webové aplikaci, kterou provozuje obchodník a na níž poskytuje služby, o které má zákazník zájem. Zákazník si zvolí produkt a rozhodne se ho zaplatit pomocí platební karty. Obchodník poskytující tuto službu vytvoří uživatelské rozhraní ve formě formuláře, kam zákazník vyplní údaje o platební kartě, jako je číslo platební karty, datum expirace a třímístný bezpečnostní kód. Dále pak kontaktní informace o zákazníkovi a způsob dopravy.

Poté co zákazník tyto informace vyplní, stiskne potvrzovací tlačítko a obchodník tyto informace odešle na platební bránu. Platební brána je software, který se chová jako most mezi obchodníkem a bankami, komunikující s citlivými informacemi transakce jako jsou například údaje o platební kartě zákazníka. Platební brána tyto citlivé informace zašifruje a odešle je platebnímu procesoru ke zpracování.

Platební procesor je společnost, která je napojena na banku akceptující platební karty neboli acquiring bank a na účet obchodníka. Platební procesor odvádí veškerou těžkou práci v procesu platby platební kartou, zahrnující autorizaci platby, kontrolu dostupných platebních prostředků na účtu zákazníka, detekci případných pokusů o podvody a mnoho dalších věcí.

Vystupují zde dvě banky, které jsou zapojeny do každého procesu platby platební kartou na internetu. Jsou to emitující banka, neboli issuing bank, a banka akceptující platební karty, neboli acquiring bank. Emitující banka je banka, která vydala platební kartu držitele karty v zastoupení Visa nebo MasterCard, v tomto případě kartu zákazníka. Banka akceptující platební karty, neboli acquiring bank, zpracovává platební transakce v zastoupení obchodníka a poskytuje mu obchodní účet. Každá síť platebních karet má vlastní soubor regulí a politik, které určují, jak by platby na internetu měly probíhat, jak má probíhat zpětné zúčtování a práva a zodpovědnost obchodníků a držitelů karet.[2]

Jakmile držitel karty dokončí online objednávku, začíná proces platby. Jsou zde 4 fáze procesu transakce.

- Povolení,
- dávkování,
- zúčtování,
- financování.

Autorizace

Skrze platebního procesora dostává obchodník oprávnění k prodeji. Toto oprávnění zaručuje, že platební karta nebyla nahlášena jako ukradená a že držitel karty má na účtě dostatečný zůstatek pro provedení platby. Povolení je uděleno v rámci několika sekund. V případě, že je transakce povolena, prodej může pokračovat k dalšímu kroku.[2]



Obrázek 2.3: 1. fáze platebního procesu platby kartou na internetu s názvem Povolení. [2]

Dávkování

Obchodník povolené a provedené platby ukládá a obvykle na konci dne zpracovává dávku. [2]

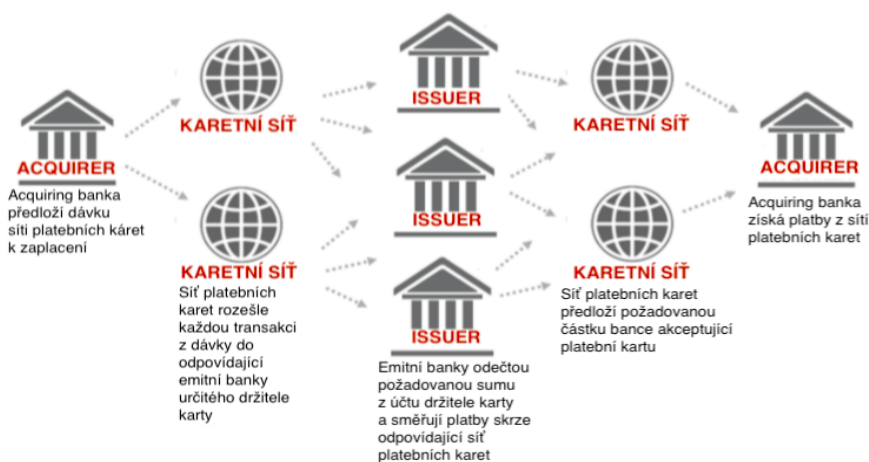
2. ANALÝZA



Obrázek 2.4: 2. fáze platebního procesu platby kartou na internetu s názvem Dávkování. [2]

Zúčtování

Poté, co je dávka zpracována, odešle banka akceptující platební karty (acquiring bank) platební transakce do odpovídajících sítí platebních karet ke zpracování. Tyto karetní sítě předají informace o transakcích do odpovídajících emitních bank (issuing banks), které strhnou odpovídající obnosy z účtů držitelů karet, a poté vyšlou obnosy, které strhly z účtů držitelů karet, zpět do banky akceptující platební karty (acquiring bank). [2]



Obrázek 2.5: 3. fáze platebního procesu platby kartou na internetu s názvem Zúčtování. [2]

Financování

Banka akceptující platební karty (acquiring bank) vloží finanční prostředky z dávky na účet obchodníka, snížené o případné poplatky, které si účtovaly entity v průběhu platebního procesu jako jsou emitní banka (issuing bank),

banka akceptující platební karty (acquiring bank) , platební procesor, platební brána a síť platebních karet. Jakmile jsou peníze uloženy na obchodníkův účet, jsou obchodníkovi k dispozici pro použití. [2]



Obrázek 2.6: 4. fáze platebního procesu platby kartou na internetu s názvem Financování. [2]

Toto jsou 4 fáze platebního procesu, kdy pro nás je v tomto případě nejdůležitější fáze první, a to celkové povolení a zpracování platebního požadavku. Pro aplikaci ElateMe je ze všeho nejdůležitější vybrat vyhovující platební procesor tak, aby se dalo říci, že je spolehlivý, má důvěru zákazníků, je vyhovující pro začínající aplikaci, jejichž obrat prozatím není zvláště velký, dodržuje PCI bezpečnostní standardy, což jsou standardy jak nákladat s citlivými informacemi platebních karet, funguje mezistátně, což znamená, že lze platit i jinou měnou než je koruna česká a především, aby poplatky za služby platebního procesora vyhovovaly potřebám aplikace ElateMe. Je tu mnoho věcí, které je třeba uvážit předtím, než se rozhodneme pro toho správného a námi vyhovujícího platebního procesora. Není žádná správná a špatná odpověď, ale pouze to, co odpovídá nejlépe naší situaci.

2.3.1.1 Typy platebních procesorů

Předtím, než se dostanu k samotnému výběru správného platebního procesora, je třeba uvést, že existují tři typy platebních procesorů.

- Obchodníkův účet + platební brána.
- Všechno v jednom řešení.
- Zjednodušený procesor kreditní karty.

Do těchto tří kategorií spadají veškeré platební procesory a předtím, než vybereme platební procesor pro aplikaci ElateMe, tyto tři typy popíšeme.

Obchodníkův účet + platební brána

Tradiční metoda pro přijímání online plateb je kombinací obchodníka účtu a platební brány. Zprostředkovatelé online platebních služeb jako jsou SecureNet (USA), Authorize.Net (USA) nebo eWay (Austrálie a UK) do této kategorie spadají. Obchodní účet je speciální typ byznys bankovního účtu, který dovoluje přijímat platby kreditní kartou. Online platební brána propojuje obchod a obchodníkův účet, usnadňuje zpracování platební transakce mezi různými zainteresovanými stranami včetně banky a banky vydavatele karty (issuing bank). Jsou zde ovšem poplatky spojené s obchodním účtem i s platební bránou, je zde ale mnoho možností jak se poplatky účtují, některé si účtují více konkurenčních poplatků za transakce nebo neúčtují instalační poplatky.

Jedna věc, kterou je třeba si uvědomit o použití tohoto typu platebního procesoru, je taková, že je třeba požádat o přístup jak u obchodního účtu tak u platební brány a to většinou vyplňováním formulářů o společnosti obchodníka, který o službu žádá, a poskytováním určitých finančních informací o společnosti. Zpracování takových požadavků může trvat až několik dní, což znamená, že obchodník nebude moci přijímat platby ihned. Po schválení obchodního účtu i platební brány, je potřeba připojit obchodní účet k platební bráně a poté připojit platební brána na obchod (internetovou aplikaci) obchodníka. Obvykle to zahrnuje konfiguraci vašeho internetového obchodu pomocí API klíčů, sdílených tajemství a tokenů.

Výhodou při používání platební brány je to, že většinou máte jako zákazník využívající tyto služby lépe udržovanou a osobní zákaznickou podporu, jelikož platební brány většinou dodávají velké společnosti, které mají velkou uživatelskou podporu. Často se budete moci dovolat na zákaznický servis, kde odborník vyslechne vaše otázky a pokusí se vám s nimi pomoci.

Platební brány mohou také nabídnout větší kontrolu nad bezpečnostními pravidly a lepší přizpůsobení na vaši internetovou aplikaci. Tento typ platebního procesoru se hodí více pro větší společnosti, které potřebují řešení šité více na míru.

Všechno v jednom řešení

Tento typ platebního procesoru, který zahrnuje společnosti jako jsou PayPal nebo 2Checkout, kombinuje obchodníkův účet a platební bránu do jednoho řešení, což může mít za výsledek rychlejší a snadnější založení a použití v internetové aplikaci. Umožňují přijímat všechny hlavní kreditní karty jako jsou Visa, MasterCard nebo AmericanExpress a obvykle mají výhodné transakční sazby. U tohoto typu platebního procesoru, u základního účtu, není třeba platit měsíční poplatky a nebo poplatky za nastavení. Za některé přidané funkce u některých služeb je ovšem třeba měsíční poplatky platit.

V minulosti bylo jednou potencionální nevýhodou, že ne vždy nabízejí bezproblémový podklad pro zadání platebních informací. V závislosti na tom, jak

bylo řešení nastaveno, zákazníci internetového obchodu mohli být přesměrováni na jinou stránku pro zadání platebních informací a provedení platby. To už ale dnes neplatí. Od té doby, co je PayPal poháněn technologií Braintree a technologií One-touch, rozhodujete vy o tom jak bude zákaznickova zkušenost s platbou vypadat. Toto je velmi důležité, jelikož je PayPal velmi populární. Má více než 100 milionů uživatelů a pohání více jak 60 % online transakcí. Je doporučeno nakombinovat PayPal s další službou, aby měl zákazník možnost si vybrat, jakým způsobem platbu provede.

Zjednodušený procesor kreditní karty

Služby, jako je například Stripe odstranili, většinu složitostí z procesu příjímání plateb online. Z pohledu obchodníka, je tento typ platebního procesoru velmi podobný typu všechno v jednom řešení, který jsem popsal výše, obvykle se ale bez problému integruje s obchodníkovou internetovou aplikací, což znamená že zákazník nikdy neopustí aktuální stránku, na které nakupuje a provádí platbu, ale vše se odehrává na téže stránce. Tento typ platebního procesoru je především už nastaven a umožňuje tedy obchodníkovi přijímat různé platby s otázkou pár minut. Není zde tedy potřeba použití obchodního účtu nebo platební brány, což znamená méně poplatků pro obchodníka. Ceny jsou konkurenceschopné s typem procesoru všechno v jednom řešení a většina těchto nových platebních procesorů také neúčtuje měsíční a instalační poplatky. Je to jednoduché, zákazník zadá údaje o kreditní kartě, potvrdí platbu a platba se zpracuje, to je vše.

Jedna věc, kterou je třeba uvážit, než se pro tento typ platebního procesoru obchodník rozhodne, je, že ne vždy a u každé společnosti je dostačující zákaznická podpora. Mají tendenci být lehké k použití a instalaci, tudíž se předpokládá, že psaná dokumentace k produktu bude dostačující. Může se ale stát, že obchodník narazí na problém s společností zpracovávající platební karty, v tom případě je při řešení problému obchodník odkázán na články nebo na emailovou komunikaci se společností. Většina společností, poskytující tento typ procesoru, ani nemá oddělení s zákaznickou podporou.[15]

2.3.2 Výběr platebního procesoru pro ElateMe

Ještě předtím, než došlo na samotný výběr platebního procesoru, bylo zapotřebí si pro potřeby aplikace zhotovit podnikatelský účet u některé banky. Při výběru vhodné banky se přihlíželo i k tomu, aby daná banka poskytovala řešení pro příjímání internetových plateb v podobě kreditní karty. Platební brány poskytované bankou mají dvojnásobně menší provize z plateb než procesory, které se zaměřují přímo a pouze jenom na dodávání platební brány. To bylo jedním z hlavních důvodů, proč vybírat platební bránu poskytovanou bankou. Takovým kritériím vyhovovalo hned několik českých bankovních poskytovatelů jako jsou Raiffeisen banka, Česká Spořitelna, UniCredit banka,

ČSOB nebo Fio banka. Všechny banky až na Fio banku odmítly dodat platební bránu a poskytnout obchodní účet z toho důvodu, že měly problém s charakterem služby ElateMe, což je crowdfunding. Crowdfunding je způsob financování, kde více jednotlivců přispívá menším obnosem k cílové částce [16]. Tyto banky poskytují platební brány pouze internetovým prodejnám, což jim nakazují vnitřní pravidla banky. Jelikož Fio banka začala poskytovat platební bránu jako poslední, musela se zaměřit kromě na eshopy i na ostatní formy společností, které potřebují platební bránu.[17] Díky tomu, že si pan Maněna, jednatel společnosti ElateMe, u Fio banky založil obchodní účet, byl proces s vyjednáváním platební brány pro příjímání platebních karet hladký. Tento typ platebního procesoru spadá do druhé kategorie, jenž jsem výše popsal, a to do kategorie všechno v jednom řešení. Zde funguje platební brána zároveň jako platební procesor, který komunikuje se sítí platebních karet. Platební brána, dodávána Fio bankou, je poháněna a vytvořena společností First Data, která také má na starost zákaznickou podporu co se týče chodu platební brány.

Výhodou zvoleného platebního procesoru je především možnost integrace s obchodním účtem uživatele, provoz platební brány je z důvodu založení obchodního účtu u Fio banky zcela zdarma, poplatky za transakce jsou nižší než u některých konkurentů, technická podpora ze strany společnosti First Data je dodávána 24 hodin denně a 7 dní v týdnu a platby jsou zabezpečeny technologií 3D Secure. Zabezpečení 3D Secure je forma zabezpečení sloužící k ochraně informací o platební kartě. Při každé platbě u obchodníků, kteří využívají tento protokol, je zákazníkovi zaslán SMS kód, kdy platba proběhne až po jeho zadání [18].

Nevýhodou u námi vybraného řešení je především to, že Fio banka umí přijímat platby pouze z českého účtu, což znamená, že v případě že k platbě dojde v jiné měně než je česká koruna, tak dojde ke konverzi z dané měny na českou korunu. Za tuto konverzi si Fio banka účtuje poplatky a může dojít ke ztrátě peněz při nevýhodném kurzu měny. Předpokládáme tedy, že zahraniční uživatelé využijí spíše jiné formy platby na naší stránce, jako je například PayPal nebo pomocí QR kódu. Další nevýhodou může být to, že při platbě dojde k přesměrování na platební bránu, která není na stejné stránce, kde uživatel provádí platbu. Sice se po zadání platebních údajů a potvrzení dojde zpět k přesměrování uživatele na stránku, kde platba započala, ale toto může v uživateli vyvolat nedůvěru.

2.4 Bitcoin Payment

Bitcoin je senzací 21. století co se týče elektronických peněz neboli krypto-měn. Je to internetová open-source P2P platební síť a zároveň i v této síti používaná měna. P2P, což je zkrácený název pro peer-to-peer, je počítačová síť typu client-client. V P2P síti nevystupuje žádný server, všichni totiž vystupují jako klienti a komunikují mezi sebou. Hlavní výhodou tohoto modelu

je, že v případě růstu počtu uživatelů roste celková přenosová kapacita sítě, kdežto v případě client-server, s narůstajícím počtem uživatelů průměrná přenosová rychlost klesá. Je to dáno tím, že v případě client-server se uživatelé musí dělit o konstantní kapacitu serveru, kdežto u P2P sítě její kapacita roste s přibývajícím počtem uživatelů.[19]

Sít bitcoin byla založena a funguje od roku 2009, kdy jí vytvořil člověk nebo skupina lidí s názvem Satoshi Nakamoto. Dodnes se neví, o jakou osobu se přesně jedná nebo jaké osoby patřily do skupiny, která síť založila.

Hlavním rysem, kterým se bitcoin odlišuje od ostatních platebních sítí, je jeho plná decentralizace. Což znamená, že síť byla navržena tak, aby nikdo nemohl měnu ovlivňovat, padělat, zabavovat účty, ovlivňovat platební toky nebo jinak zasahovat do procesů sítě, dokonce ani sám autor. Je to dáno tím, že v síti neexistuje žádný centrální bod, ani nikdo, kdo by o síti mohl rozhodovat. Jediné, co je známo a předem dané, je celkové množství bitcoinů, jejichž vypouštění do oběhu sítě je definováno v samotném zdrojovém kódu. Proto není možné vytvořit umělou inflaci vyvoláním většího množství bitcoinů.

Napříč sítí peer-to-peer funguje distribuovaná databáze, která slouží pro provádění transakcí mezi uživateli v síti. K zabezpečení celkové sítě je použita kryptografie, která umožňuje používat pouze peníze, které uživatel vlastní a zabraňuje znovupoužití peněz již utracených.

Uživatelé si bitcoiny, které vlastní ukládají do takzvaných peněženek/adres vygenerovaných pro příchozí transakci. Každý uživatel vlastní jednu nebo více peněženek/adres. Doporučuje se pro každou transakci vygenerovat novou adresu, z důvodu větší bezpečnosti. Ke každé vygenerované peněženice vlastní uživatel soukromý klíč, díky kterému je zůstatek uživatele skryt pro ostatní uživatele sítě. Adresy vygenerované pro nové transakce si uživatel ukládá do peněženek, ke kterým vlastní soukromý klíč. Celkový obnos peněz je pak uživateli zobrazen na peněženice jako součet peněz z adres, které v dané peněženice jsou.

2.4.1 Integrace do aplikace ElateMe

Možnost přispívat peníze na přání v aplikaci ElateMe v podobě bitcoinu jsme zvolili z toho důvodu, že je platba pomocí bitcoinu novým způsobem platby na internetu, který je na vzestupu a stále více internetových aplikací, které implementují platební procesy, daný způsob platby využívá. Jelikož bude mít aplikace ElateMe globální rozsah, doufáme, že možnost využít platby bitcoinem ocení hlavně zahraniční zákazníci. Češi vlastní bitcoiny a obchodují s nimi, nepoužívají ale bitcoiny pro platby tak aktivně jako zahraniční státy. V české internetové obchodní sféře implementuje platbu pomocí bitcoinů například Alza.cz, která tuto možnost vytvořila v květnu minulého roku, kde do svých provozoven zavedla bitcoin bankomaty od společnosti General Bytes a možnost platit touto kryptoměnou pomocí platební brány BitcoinPay.

Měli jsme dvě možnosti, jak do aplikace ElateMe implementovat platbu pomocí bitcoinů. První možností bylo si celý interface a logiku, které obsahuje generování platebních ID, přesuny bitcoinů mezi peněženkami, ukládání bitcoinů do účtů a peněženek, převody bitcoinů na realné měny a následně jejich posílání na odpovídající bankovní účty a mnoho dalších operací sami naimplementovat. Tato možnost nám nepřišla jako vyhovující pro naši začínající aplikaci z mnoha důvodů. Hlavním problémem by byla zejména časová a finanční náročnost zhotovení potřebného backendu pro platbu bitcoiny včetně vyřešení bezpečnostní stránky transakce, což je v případě virtuální měny nelehký úkol. Dalším problémem by byla nedůvěra zákazníků. Jelikož bude aplikace ElateMe na internetu nováčkem, nemůžeme se spolehnout na to, že uživatelé budou důvěřovat námi vytvořenému API pro bitcoinové platby. Tato možnost by se vyplatila spíše pro aplikace, které jsou obsáhlejší a mají především už důvěru zákazníků.

Druhou možností a také tou, kterou jsme pro naši aplikaci zvolili, je možnost využít API nebo knihovny, které poskytují webové aplikace třetích stran, které jsou již prověřené a internetovými aplikacemi využívány. Tyto aplikace poskytují interface pro platební procesy s bitcoiny ve formě API nebo knihoven, a fungují jako prostředník při transakcích s bitcoiny. Většinou je hlavní podmínkou k využívání těchto služeb vytvoření uživatelského účtu u dané aplikace, založení peněženky a uchovávání virtuálních peněz v této peněžence. Dále si pak společnost, která API nebo knihovny poskytuje, odebírá z každé provedené transakce určité procento za poskytování transakčních služeb.

Po prozkoumání internetu, kde jsme hledali nejvhodnější společnosti pro poskytování bitcoinových plateb, jsme zvolili k užšímu průzkumu a výběru dvě aplikace, a to

- Bitcoin Pay[3],
- Bit Pay[20].

Tyto dvě společnosti jsme vybrali k bližšímu prozkoumání z toho důvodu, že obě dvě mají důvěru zákazníků, nějakou dobu už aktivně fungují a jejich služby využívají větší e-shopy jak v České Republice v případě Bitcoin Pay tak v zahraničí co se týče Bit Pay. Obě dvě společnosti dodávají k operacím s bitcoiny API, potřebné pro vývojáře k integraci na webových stránkách, kde budou poskytovat zákazníkům platby pomocí bitcoinů a především jsou tyto API velmi dobře zdokumentované.

Níže popíšeme obě dvě společnosti detailněji, vytknu klady a zápory a zdůvodním, proč jsme si danou společnost pro poskytování bitcoin platebních služeb vybrali.

Bitcoin Pay

Bitcoin Pay je Česká společnost, která poskytuje přijímání bitcoin platby pro obchodníky a odesílají peníze na účty obchodníků už do jednoho dnu. Společnost funguje od roku 2014, kdy v roce 2013 odstartovala se získáním druhého místa v prestižní soutěži podnikatelského akcelérátoru Starcube. Od roku 2014 si společnost získala důvěru více jak 400 obchodních partnerů, kde mezi největší patří Alza.cz nebo Moneypolo.com.

Společnost poskytuje velmi dobře dokumentované API pro vývojáře, kteří mohou díky tomu na webových aplikacích poskytovat jednoduchý a rychlý platební proces pro uživatele. Jedním kliknutím bude uživatel přesměrován na stránku, kde se mu zobrazí adresa peněženky kam má poslat obnos bitcoinů, který je na stránce taky zobrazen. Informace jsou dostupné i v podobě QR kódu, tudíž je možné platit i pomocí aplikace pro snímání QR kódů. Po zaplacení potřebného obnosu na danou adresu bude zákazník na téže stránce informován o zaplacení. Na platbu má uživatel 15 minut a to z důvodu stálosti kurzu bitcoinů. Hodnota bitcoinu velmi rychle mění kurz, tudíž musí platba proběhnout právě do daných patnácti minut, kdy je společnost schopna zajistit stálý kurz. Kromě API a platebních tlačítek dále společnost poskytuje integraci bitcoinových plateb i do platebních terminálů, do plateb emailem a poskytují pluginy do e-shopů, kde k jejich použití není zapotřebí programátorských znalostí, ale pouze si daný plugin nainstalovat podle návodu, který je k nalezení na webové stránce.



Obrázek 2.7: Detail informací k zaplacení pomocí bitcoinů od společnosti BitcoinPay. [3]

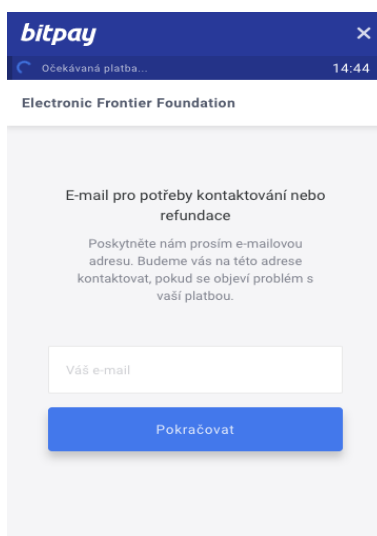
Společnost dále umožňuje bitcoiny převádět do světových měn jako je USD, EUR, CZK nebo polské Złote PLN a ve zvolené měně peníze na účty obchodníků. Kde jsou ke konverzi využity aktuální světové kurzy. Společnost Bitcoin Pay si za poskytování těchto služeb účtuje 0,8 % z každé provedené platby. Díky správnému know-how a analýze blockchainu umožňuje tato společnost garantovat platbu mnohem rychleji než je běžné. Potvrzení bitcoinové sítě netrvá minuty ale pouze sekundy. Další výhodou Bitcoin Pay je, že veškerá dokumentace k API je k dispozici i v českém jazyce a zákaznická podpora funguje v jazyce českém taktéž.

Bit Pay

BitPay je americká společnost založena v roce 2011, která dodává bitcoinové platební procesy na internetu i v kamenných prodejnách. BitPay je jednou z největších společností na světě pro dodávání bitcoinových platebních procesů. V roce 2014 se partnery BitPay staly firmy jako je například Microsoft, Warner Bros nebo PayPal. Bitcoin, které BitPay zpracuje lze zasílat na účty zákazníků v peněžní podobě dokonce až v osmi světových měnách s poplatkem za transakci 1 % z posílané částky.

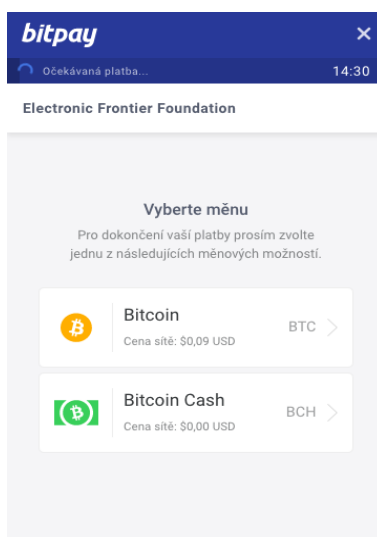
BitPay poskytuje integrace s platebními aplikacemi, které není potřeba nijak programovat, ale stačí je do e-shopu pouze nainstalovat. Dodává open-source pluginy, BitPay vydává knihovny pro populární programovací jazyky jako jsou například NodeJS, python, C nebo Java, vyvinuli open-source bitcoinové projekty jako jsou například Bitcore nebo Copay a mnoho dalších způsobů, jak BitPay využít pro platby pomocí bitcoinu ve webové aplikaci.

Co se týče samotné interakce uživatele s uživatelským rozhraním, které společnost BitPay poskytuje, tak se dělí na rozdíl od toho, které poskytuje společnost BitcoinPay na tři části. Poté co zákazník klikne na tlačítko "Zaplatit bitcoiny" se zobrazí vyskakovací okénko na té stejné stránce s požadavkem o zadání kontaktního emailu plátce, aby bylo možné uživatele po dokončení platebního procesu dodatečně informovat, že platba proběhla správně či nikoliv. Shledáváme tuto funkčnost jako velmi důležitou, jelikož v případě, že zákazník zaplatí menší nebo větší částku, bude o tom ihned informován v podobě emailu a může danou situaci ihned řešit.

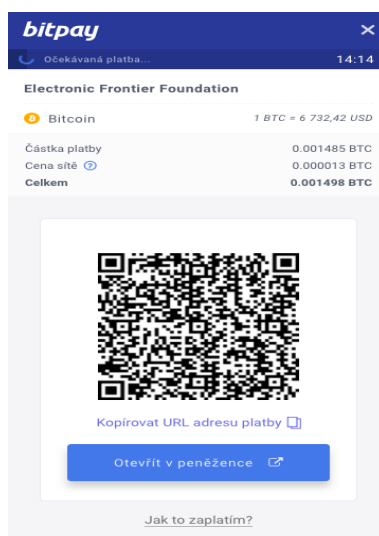


Obrázek 2.8: Krok 1 při platebním procesu, který dodává společnost BitPay. [4]

Po zadání emailové adresy se zákazník dostane na další stránku, na které si vybere, zda zaplatí přímo bitcoiny a nebo pro platbu zvolí bitcoin cash, což je kryptoměna, která vznikla v minulém roce v srpnu odštěpením od kryptoměny bitcoin.



Obrázek 2.9: Krok 2 při platebním procesu, který dodává společnost BitPay. [4]



Obrázek 2.10: Krok 3 při platebním procesu, který dodává společnost BitPay. [4]

Posledním krokem je zobrazení celkového obnosu bitcoinu, které musí zákazník zaplatit, včetně poplatku za zprostředkování platby a převod do peněžní hodnoty, který si společnost BitPay účtuje. Dále jsou v tomto posledním kroku zobrazeny tři možnosti, jak bitcoiny převést na obchodníkův účet a to pomocí

- QR kódu,
- zkopírovat kód transakce a poslat bitcoiny z vlastní peněženky na webu,
- zaplatit pomocí peněžní aplikace, pokud jí má zákazník nainstalovanou na zařízení které používá pro platbu.

Stejně jako u platebního procesu poskytovaného společností BitcoinPay, tak i zde má zákazník na vyřízení platby pouze 15 minut, jelikož to je doba, po kterou společnost zaručuje stálý kurz bitcoinu. Pokud dojde k poklesu nebo nárustu bitcoinu v těchto patnácti minutách, nijak to nepostihne obnos bitcoinů, který musí zákazník zaplatit, celá odpovědnost za případné dopláčení platby padá na společnost. Po zaplacení obnosu zákazníkem se po určité chvíli na samém okně pro placení zobrazí informace o schválení transakce či jejím zamítnutí. Toto by nemělo trvat více než pár sekund od doby, kdy zákazník odešle bitcoiny na správnou adresu.

Co jsme si vybrali

Po pečlivém prozkoumání všech pro a proti mezi dvěma společnostmi, které dodávají funkční API pro platby pomocí bitcoinů, jsme se rozhodli využít

služby společnosti BitPay.

I přes to, že má o 0,2 % vyšší poplatek za zprostředkování každé platby a dokumentace a zákaznická podpora funguje pouze v anglickém jazyce, jsme se pro tuto společnost rozhodli především kvůli dobrému jménu a důvěře zákazníků, kterých je po celém světě mnoho a také proto, že jak už jsem výše zmiňoval, tak aplikace ElateMe má být globálně zaměřena. Společnost BitPay podporuje převod bitcoinů až do osmi různých světových měn, kdežto společnost BitcoinPay pouze do čtyř, proto chceme vyjít vstříc zahraničním zákazníkům a umožnit jim větší výběr pro převod bitcoinů na jejich tuzemské účty v jejich měně.

Dále se nám zamlouvá skutečnost, že kromě API poskytují i jiná řešení integrace platebního procesu do webových aplikací v podobě pluginů, integrovaných řešení jako je například Shopify nebo 3DCart, knihovny k nejpoužívanějším programovacím jazykům jako jsou třeba Java, NodeJS, Python nebo PHP a celkově dalších 40 různých integrací, kterých stále přibývá. V případě, že v budoucnu společnost BitPay přestane dodávat API pro své služby, tak nebude problém využít nějakou z dalších integrací, které společnost poskytuje, a nebude hrozit ztráta získané důvěry zákazníků při změně zprostředkovatele platebních služeb pro bitcoinové platby.

2.4.2 Proces platební transakce platby pomocí bitcoinů

Proces bitcoinové platby se spouští v ten okamžik, kdy zákazník zvolí možnost, že chce platit pomocí bitcoinů. V tu chvíli na straně serveru obchodníka dojde k vytvoření platebního příkazu pomocí volání API, které nám, jak jsme výše rozhodli, bude dodávat společnost BitPay. Obchodníkův server musí tomuto volání dodat potřebné informace pro vytvoření platebního požadavku, jako je obnos, který bude zákazník platit, a měna, ve které bude platba provedena. V případě, že měna nebude označena jako BTC (bitcoiny) dojde k automatické konverzi částky podle aktuálních kurzů na straně serveru BitPay na bitcoiny. Tyto dva parametry jsou povinné, existuje pak řada dalších parametrů které povinné nejsou, ale doporučují se, jako je například notifikační URL, což je URL stránky, kde si bude server obchodníka odchyťávat změny stavu transakce. Výsledkem tohoto volání jsou informace o transakci, kde nás hlavně zajímá ID transakce, které budeme předávat funkci, kterou dodává společnost dodávající API, pro zobrazení vyskakovacího okénka, které bude uživatele navigovat k platbě, jak jsem popsal výše.

Po zobrazení okénka s informacemi o platbě uživatel zvolí jednu ze tří možností, jak bitcoiny zaplatit, a pomocí adresy, kterou zobrazilo uživatelské rozhraní dodané společností BitPay, uživatel s pomocí peněženky, na které vlastní dostatečný obnos bitcoinů provede platbu. Tím kontaktuje bitcoinovou P2P síť, kde se drží informace o všech platbách, transakcích a celkově o uložených a dostupných bitcoinech. Díky API tokenu, který se váže k uživatelskému účtu obchodníka u společnosti BitPay, který musí být dodán ke

každému API volání na server BitPay, je vygenerovaná adresa přímo vázána na peněženku obchodníka, kterou si obchodník vytvořil u společnosti BitPay, což bylo podmínkou k přijímání plateb. Díky tomu poté, co zákazník provede a potvrdí platbu bitcoiny z vybrané peněženky, bitcoivá síť naváže spojení s peněženkou obchodníka u společnosti BitPay a započne přenos bitcoinů. Server BitPay pošle na notifikační URL adresu, která byla doporučena pro vyplnění při vytváření požadavku na platbu. Na této adrese si server obchodníka zkontroluje stav platby. Pokud stav platby není "complete", což je jediný validní stav, který nám říká, že platba proběhla v pořádku a na účet obchodníka byla připsána platba, posílá server obchodníka v určitých časových intervalech požadavek o kontrolu stavu na server BitPay, kde výsledkem je stav transakce. Tato akce se opakuje do té doby, dokud není stav transakce "complete" nebo jeden z negativních příznaků stavu, a to "expired" a nebo "invalid".

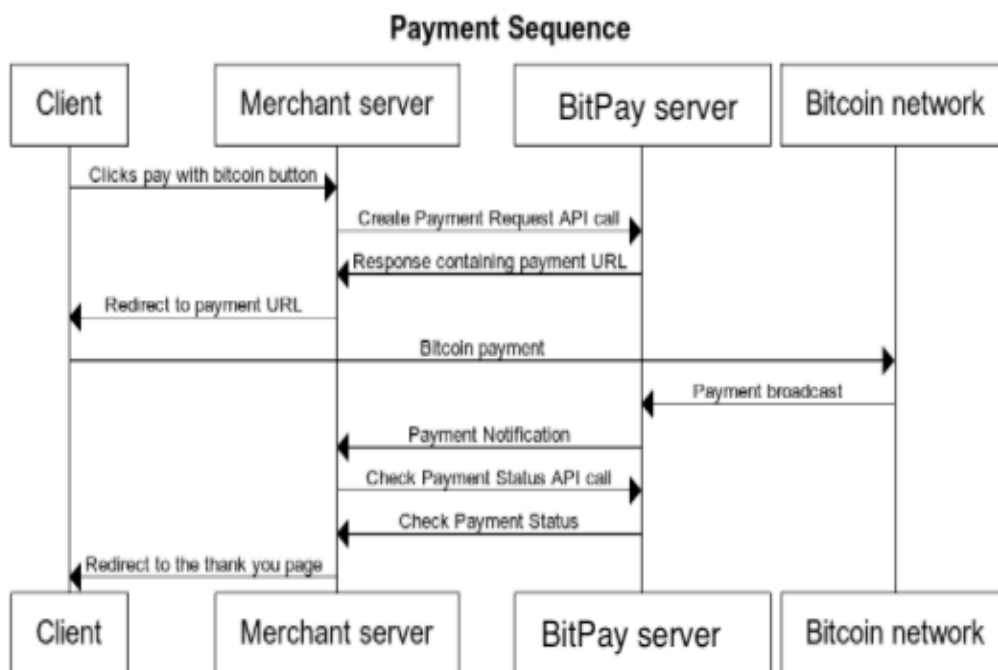
V každém případě bude zákazník informován obchodníkovým serverem o výsledku platební transakce, ať už kladně, v případě stavu "complete", kdy se uživateli zobrazí informace v podobě vyskakovacího okénka a nebo v případě, kdy stav bude "expired" či "invalid". Stav "expired" nastane, pokud zákazník neprovede platbu do patnácti minut, které běží od chvíle, kdy se zobrazí okénko s informacemi k platbě, a stav "invalid" nastane pokud někde při přenosu došlo k chybě. Pokud nastane stav "invalid", tak zákazníkovi nebudou strhávány odeslané bitcoiny, k odečtení odeslaných bitcoinů dojde teprve pokud budou úspěšně připsány na účet obchodníka. Uživatel bude informován o neúspěchu transakce opět v podobě vyskakovacího okénka.

Pro shrnutí tedy, hlavními aktéry v procesu platby pomocí bitcoinů jsou uživatel (zákazník), server obchodníka, server dodavatele API (BitPay), bitcoinová P2P síť. A hlavní akce a proces můžeme popsat stručně následující sekvencí kroků.

1. Zákazník zvolí platbu pomocí bitcoinů.
2. Server obchodníka voláním BitPay API požádá o vytvoření faktury.
3. Server BitPay posílá odpověď v podobě JSON (Javascript Object Notation).
4. Server obchodníka zpracuje odpověď a zobrazí zákazníkovi uživatelské rozhraní pro platbu.
5. Zákazník pošle obnos bitcoinů na vygenerovanou adresu.
6. Bitcoinová síť započne převod na danou adresu.
7. Server Bitpay pošle notifikační url na obchodníkův server o stavu transakce.

8. Server obchodníka kontroluje stav transakce dokud není validní a nebo neseleže, poté o výsledku informuje zákazníka.

Popsané kroky a aktéry procesu znázorňuje následující diagram.



Obrázek 2.11: Sekvenční diagram platby pomocí bitcoinů [5].

2.5 QR Code payment

QR kód, což je zkratka z anglického názvu “Quick Response“, je prostředek, který slouží k automatizaci sběru dat. QR kód je nástupcem klasického čárového kódu, který je na každém výrobku, který si můžeme koupit, a nese informace o daném výrobku. QR kód slouží ke stejnému účelu s rozdílem, že dokáže pojmout daleko větší množství informací o produktu než čárový kód. Jedná se o takzvaný dvoudimenzionální kód (čtvercová matice), který obsahuje černé čtverečky ve čtvercové mřížce na bílém pozadí, a dokáže ho přečíst a zobrazit zakódované informace jakékoliv zařízení, které obsahuje kameru. QR kód je od roku 2000 standardem ISO 18004. Tento standard byl naposledy upraven v roce 2006. Tyto kódy jsou určeny pro počítačové zpracování, jelikož využívají velké množství technik, které předcházejí chybám v interpretaci. Musí být odstraněna nebo znečištěna větší část kódu aby se kód stal nečitelným. QR kódy jsou definovány až v čtyřiceti různých velikostech (1 až 40) a mohou obsahovat číslice, písmena nebo japonské znaky kandži. Kód verze v je tvořen čtvercovou mřížkou o velikosti $17 + 4 * v$. Do takového QR kódu se dá zakódovat skoro jakákoliv informace. Používají se například na kódování internetových odkazů, na které se po vyfocení fotoaparátem na chytrém telefonu dá automaticky přejít, kódování textových zpráv a především to co nás zajímá nejvíce a to, že pomocí QR kódu lze zakódovat informace potřebné k provedení platební transakce, k čemuž slouží speciální aplikace pro načítání QR kódu.

2.5.1 Použití v aplikaci ElateMe

Implementovat platbu pomocí QR kódu jsme se do aplikace ElateMe rozhodli proto, protože to je rychlý a jednoduchý způsob, jak provést platbu převodem, kdy do QR kódu zakódujeme informace o účtu, kam se mají peníze poslat. Jediné co je potřeba, je, aby zákazník, který si zvolí tento způsob platby, na svém mobilním zařízení, kterým nascanuje QR kód, vlastnil aplikaci banky, jež podporuje funkci zřídit platbu naskenováním QR kódu. Zákazník tedy naskenuje pomocí mobilního zařízení QR kód a v aplikaci banky se vytvoří šablona pro převod peněz. Informace o transakci si aplikace vezme právě z QR kódu, které jsou v něm zakódované. Vznikne tedy šablona, která bude obsahovat číslo účtu, na který se budou peníze odesílat, částka a v případě QR kódu, který pro uživatele k určitému přání vytvoříme, bude obsahovat ještě variabilní symbol a specifický symbol platby.

Zákazník, až dorazí v donaci ke kroku pro výběr platební metody, zvolí typ platby QR kódem a na další stránce se mu zobrazí vygenerovaný QR kód, který pouze nascanuje a ve své mobilní aplikaci provede platbu. Kromě QR kódu zde bude zobrazen účet, kam se peníze posílají, variabilní symbol a specifický symbol platby, pro případnou kontrolu, zda načtení QR kódu do aplikace proběhlo správně. Uživatel bude ihned vpuštěn na další stránku s poděkováním. Z

důvodu bankovního převodu z účtu na účet nemůžeme prozatím kontrolovat, zda platba proběhla správně či nikoliv, jelikož dle regulí ČNB by se měl tuzemský převod peněz mezi bankovními účty provést do 24 hodin od odeslání a to neuvažují zahraniční platby, kde tato doba může být značně delší. Z toho důvodu je uživatel vpuštěn na stránku s poděkováním okamžitě a díky API, které poskytuje banka, u které má společnost ElateMe s.r.o. založený účet a kam se peníze posílají, v tomto případě API FIO banky, se dá získat výpis transakcí a kontrolovat příchozí platby. Ve chvíli, kdy na účtu ElateMe přibude nová transakce, bude o této akci server ElateMe informován a dle variabilního symbolu a specifického symbolu dokážeme potvrdit platbu, která proběhla. Na základě získaného variabilního symbolu transakce zjistíme, na jaké přání byla částka přispěna a uživatele, který přání založil, informujeme o tom, že částka na jeho přání se změnila. Na základě specifického symbolu získaného z transakce informujeme příslušného uživatele, že platba kterou provedl byla provedena a připsána na přání.

Server ElateMe bude zároveň informovat příslušné uživatele i v případě, že dojde k neúspěchu platby z nějakého důvodu. V případě, že uživatel zvolí platbu pomocí QR kódu a dojde až na stránku s poděkováním, zavazuje se tím, že platbu odeslal. Jednáli se o tuzemský převod, platba by měla být provedena do jednoho pracovního dne. Pokud se jedná o převod mezi stejnými bankami, bude převod proveden do několika sekund, pokud se provádí převod mezi rozdílnými tuzemskými bankami, může převod trvat až jeden pracovní den. V případě, že je platba prováděna ze zahraničního účtu, je to už o něco déle. Zde záleží na větším množství faktorů, kde hlavním je především měna. Pokud se jedná o Eura, převod se provede cca do dvou pracovních dnů. Pokud jde o Libry, tak ty se převedou do čtyř pracovních dnů, a v případě ostatních měn může převod trvat 5 až 10 pracovních dnů[21]. Každá platba bude zaznamenána v databázi ElateMe s informací, do kolika dnů by měla platba proběhnout. V případě že tato doba splatnosti vyprší a platba nebude z API FIO banky detekována, bude uživatel, který prováděl platbu, informován o neprovedení platby a na přání nebude připsána daná částka. Dorazí-li platba po splatnosti bude uživateli, který platbu provedl vrácena.

2.6 PayPal poháněný systémem Braintree

Paypal je Americká společnost poskytující online platební systém, který podporuje online peněžní transakce, a slouží jako elektronická alternativa například k šekům nebo složenkám. Společnost byla založena v roce 1998 pod názvem Confinity a v roce 2002 společnost přišla na trh se svými akciemi pod názvem PayPal. Později v tom samém roce se společnost PayPal stala vlastnictvím Amerického internetového giganta eBay. V roce 2014 eBay oznámila, že by chtěla z PayPal udělat samostatnou společnost, a v roce 2015 se tak doopravdy stalo.[22] PayPal ve zkratce umožňuje přenos peněz mezi jednotli-

vými PayPal účty, kde jsou tyto účty identifikovány pouze emailovou adresou. Každý účet je propojen s jednou nebo více platebními kartami (Visa, MasterCard, American Express), kde tyto karty musí mít povolené internetové platby. Alternativou platby pomocí PayPalu je platit přímo kreditní kartou, zde je ale zvýšené riziko podvodu, jelikož je třeba zadat informace o kartě přímo obchodníkovi, kdežto společnost PayPal je roky prověřená společnost s vysokou důvěrou a velmi dobrým zabezpečením. V roce 2013 PayPal koupil společnost Braintree, který napomáhá k lepší uživatelské spokojenosti a hladkosti provedení platby.[23]

Braintree je dceřiná společnost společnosti PayPal. Braintree se specializuje na dodávání platební brány a obchodního účtu v jednom pro uživatele mobilních a webových aplikací. Kromě toho, zaručuje opakované účtování, uložení kreditních karet, podporuje mobilní aplikace a mezinárodní platby a dodržuje PCI bezpečnostní standardy.[24]

PayPal poháněný Braintree je tedy platební brána, která kombinuje kreditní a debetní karty s použitím technologie zvané In-Context Checkout, což je interface, který dovolí zákazníkům zaplatit pomocí jejich PayPal účtu, aniž by museli opustit stránku, ze které si platbu zvolili. Zahrnuta je technologie společnosti PayPal zvaná One Touch, umožňující opakujícím se zákazníkům, platit jejich PayPal účtem, aniž by se museli znovu přihlašovat. Kromě toho, že je služba PayPal poháněna systémem Braintree optimalizována pro desktopové i mobilní nakupování, nabízí také pokročilé funkce, jako je ochrana proti podvodům, 3D Secure a možnost autorizovat platby před zachycením, což poskytuje větší flexibilitu při získávání plateb od zákazníků. Dále umožňuje si nastavit účet tak, aby bylo možné přijímat více než jednu měnu. Tím pádem dojde k zabránění potenciální ztráty peněz, kterou může zapříčinit nestálost kurzů jednotlivých měn.[25]

2.6.1 Co je potřeba pro přijetí plateb pomocí PayPal

Obchodník, jenž chce využívat službu PayPal poháněnou systémem Braintree, musí udělat potřebné kroky předtím, než začne přijímat platby tímto způsobem. Jako první, co musí obchodník udělat, je, že si založí tři účty.

- **Obchodní bankovní účet** – tento účet si obchodník založí u banky dle svého výběru a na tento účet mu budou posílány automaticky všechny provedené transakce.
- **Braintree účet** – Braintree účet si obchodník založí na stránkách společnosti Braintree a je zcela zdarma. Skrz tento účet získá obchodník zároveň získá i obchodní účet, platební bránu a tokeny potřebné k vytváření transakcí. Doporučuje se vytvořit si před tímto účtem, účet na Braintree Sandbox, což je testovací účet, na kterém si obchodník může

otestovat průběh platby s testovacími daty, a až poté může přejít na produkční účet, který je s tímto testovacím skoro identický. V tomto účtu si obchodník také zvolí v jaké měně se budou získané peníze posílat na obchodní bankovní účet.

- **PayPal účet** – I přes to, že je Braintree dceřinou společností společnosti PayPal, je třeba mít kromě Braintree účtu i PayPal účet. Pokud obchodník vlastní PayPal účet, lze se přihlásit do Braintree pomocí tohoto účtu, obráceně to nelze. PayPal účet se s Braintree účtem prováže na kontrolním panelu Braintree účtu a nadále je průběh hladký.[26]

Zřízení Braintree účtu a jeho zpravování je zcela zdarma a co se týče PayPal účtu, je třeba zaplatit symbolický poplatek ve výši zhruba padesáti korun českých. Tato částka je hrazena z důvodu nezneužívání účtů a zabránění zakládání zbytečných účtů. Částka, která je při přihlašování uhrazena, bude v plné výši vrácena při první úspěšné transakci. Po založení těchto účtů se může obchodník dát do samotné integrace s aplikací. Braintree dodává doplňující klientský a serverový SDK (Sada vývojových nástrojů) pro dokončení integrace s aplikací.

- Klientská část SDK umožňuje bezpečně získat platební informace od zákazníka.
- Serverová část SDK umožňuje zpracovat získané platební informace a vytvořit platební transakci.

Braintree dodává SDK pro následující klientské platformy a serverové jazyky

- **Klientské SDK:** Android, iOS, JavaScript
- **Serverové SDK:** Java, .Net, Python, Node.js, PHP, Ruby

Klientská část SDK vyžaduje oprávnění aby mohla komunikovat s platební bránou Braintree. Braintree nabízí dvě formy prokázání oprávnění.

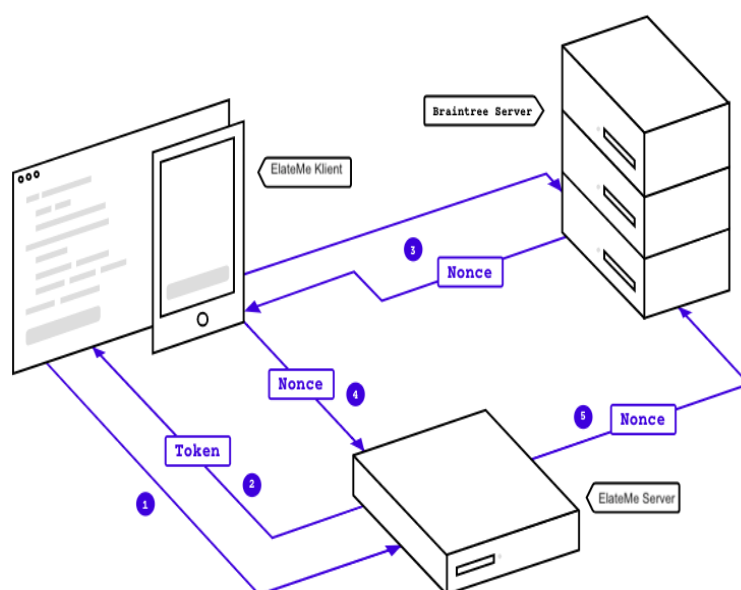
- **Tokenizační klíč** - jednoduchá, znovupoužitelná hodnota, která povoluje tokenizaci platební metody. Tento klíč se vygeneruje na kontrolním panelu účtu na Braintree.
- **Klientský token** - hodnota na jedno použití, která schvaluje tokenizaci platební metody, vyhledání platebních metod a uschování na klientské části. Tento klíč je vygenerován pomocí knihoven serverové části SDK.

Jak serverová tak klientská část SDK používají tak zvaný Payment method nonce jako odkaz na platební informace.

Payment method nonce

Payment method nonce je bezpečnostní, jednorázový odkaz na platební informace, jako jsou informace o kreditní kartě nebo citlivé informace zákazníka. Je to klíčový prvek, který umožňuje serveru obchodníka posílat citlivé platební informace serveru Braintree, aniž by se dotýkal dat samotných. Dalo by se tedy říci, že jde o jakési zašifrování citlivých informací, aby nedošlo k jejich zneužití. [27]

2.6.2 Proces komunikace s Braintree



Obrázek 2.12: Průběh vytvoření platební transakce s Braintree [6].

1. Aplikace nebo webový front-end obchodníka zažádá serverovou část o klientský token obchodníka, aby bylo možné inicializovat klientskou část aplikace.
2. Server obchodníka vygeneruje klientský token a pošle ho jako odpověď na klientskou část.
3. Jakmile je klient inicializován pomocí klientského tokenu a zákazník potvrdil informace o platbě v uživatelském rozhraní, klient naváže komunikaci s Braintree serverem. Braintree server v odpovědi posílá Payment method nonce.
4. Klientská část pošle získaný Payment method nonce na server obchodníka.

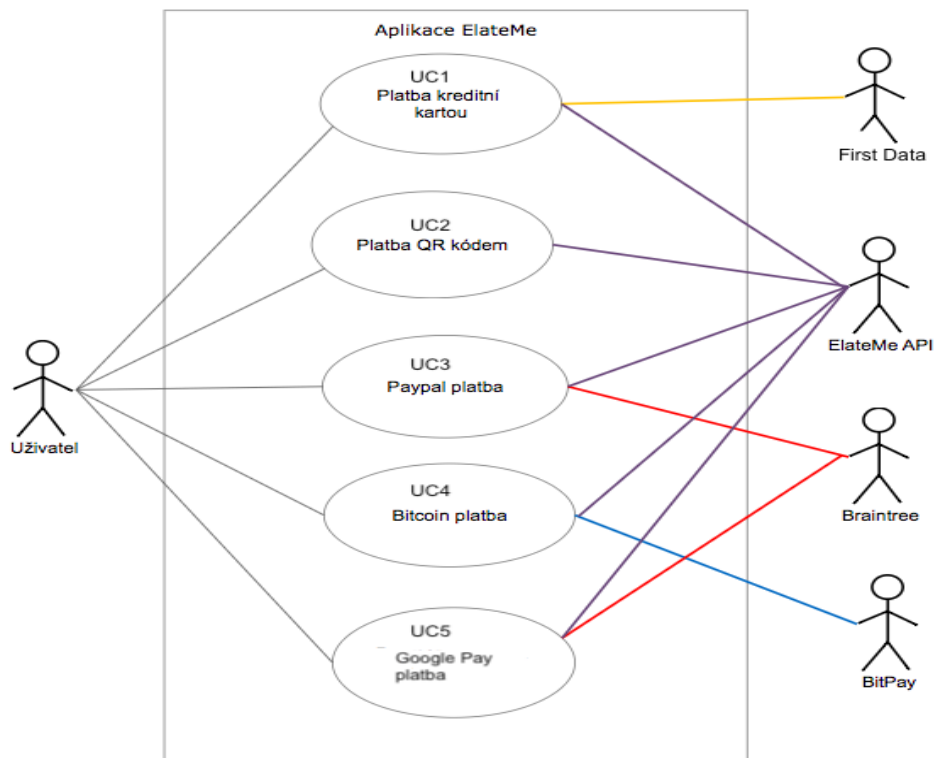
5. Server poté, co získá Payment method nonce, může vytvořit transakci a odeslat vytvořenou transakci na server Braintree, kde se transakce zpracuje a proběhne.

2.7 Diagram případů užití

Diagram případů užití je jedním z UML diagramů. Diagramy tohoto typu slouží pro znázornění interakce uživatele se systémem, kde je uživatel napojen na jednotlivé případy užití. Diagram případů užití identifikuje různé typy uživatelů a různé typy případů užití. Hlavními aktéry jsou zde dvě entity.

- **Actor** – Actor může být uživatel systému a nebo systém třetí strany, který se nějakým způsobem podílí na fungování systému.
- **Use Case** – Use Case, neboli případ užití, je akce, kterou provádí Actor.

Interakci uživatele s jednotlivými platbami jsem popsali v diagrama případů užití na obrázku 2.13.



Obrázek 2.13: Diagram případů užití při interakci uživatele s platebními metodami.

2. ANALÝZA

- **Uživatel** – Hlavním aktérem, který vystupuje ve všech případech a má interakci se všemi případy užití, je uživatel aplikace. Uživatelem se rozumí zákazník, který chce prostřednictvím příspěvku pomoci ke splnění přání jinému uživateli.
- **Braintree** – Braintree je společnost dodávající platební bránu pro PayPal a Google Pay.
- **First Data** – First Data je společnost dodávající platební bránu pro Fio banku a tím pádem i pro platby kreditní kartou.
- **ElateMe API** – Se všemi případy užití je spojeno API ElateMe serveru. Při každé akci komunikuje klient ElateMe se serverem.
- **BitPay** – BitPay je společnost poskytující služby pro bitcoin platby.
- **UC1** – Příklad kdy zákazník zvolí platbu kreditní kartou.
- **UC2** – Příklad kdy si zákazník nechá vygenerovat účet k odeslání peněz QR kódem.
- **UC3** – Příklad kdy zákazník zaplatí svým PayPal účtem.
- **UC4** – Příklad kdy zákazník zaplatí pomocí bitcoinů.
- **UC5** – Příklad kdy zákazník použije platbu Google Pay.

Návrh

V této části se budu zabývat návrhem implementace vybraných plateb. Popíši zvolené programovací jazyky a technologie, nastíním, jak bude vypadat uživatelské rozhraní, které jsem také implementoval, popíši postupy které využiji v implementační části a tyto postupy doplním o diagramy aktivi u metod u kterých to bude možné.

Pro zobrazení dostupných možností jakým způsobem na přání přispět bylo třeba implementovat uživatelské rozhraní. Rozhodli jsme se pro vyskakovací okno, které se otevře při stisku zákazníka na tlačítko “Přispět“. Zákazník si zvolí jaký obnos chce přispět na přání a na další stránce, v tom samém vyskakovacím okně, budou na výběr dostupné platby. Proces přispívání jsme zvolili ve formě vyskakovacího okénka z toho důvodu, že zákazník zůstane po celou dobu procesu na stránce přání.

Pro implementaci uživatelského rozhraní jsme se rozhodli použít JavaScriptovou knihovnu zvanou React, technologii HTML a pro vzhled a styl okénka jsme použili preprocesor CSS zvaný Sass.

React

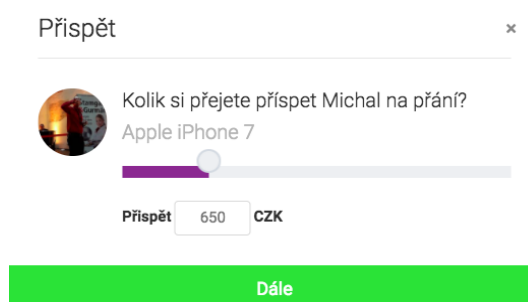
React je JavaScriptová knihovna pro vytváření webových komponent. V MVC architektuře jde o vrstvu “V“ neboli view vrstvu a mohli bychom ho přirovnat například k Latte v Nette. React v roce 2013 opensoveroval Facebook a předtím ho sám interně používal. React přichází s zásadní změnou paradigmatu a to takovou, že už se nepíše kód, který něco mění, ale kód, který popisuje jak by měl vypadat výsledek. React spojuje JavaScript s HTML, kdy se klasické HTML dá používat jako návratová hodnota JavaScriptových funkcí, ukládat se do proměnných a předávat se do funkcí a mnoho dalšího. [28]

3. NÁVRH

Sass

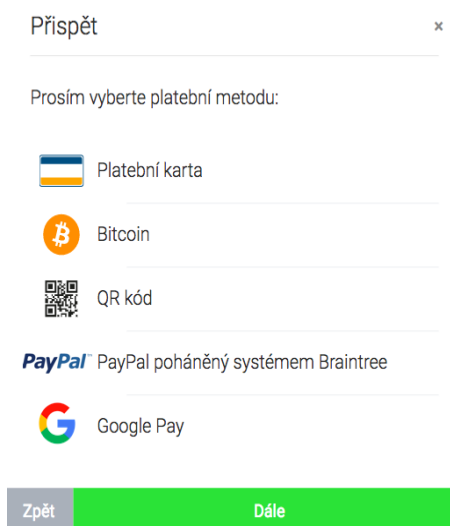
Sass je kompilovaný jazyk, který rozšiřuje syntaxi CSS o proměnné, cykly, funkce a jiné. Šetří čas, množství napsaného kódu a je přehlednější. Používá se syntaxe SCSS (Sassy CSS) a před každým použitím je SCSS soubor zkompilovat do CSS pomocí Sass kompilátoru, jelikož prohlížeč nezná SCSS syntaxi. Všechny SCSS jsou kompatibilní s CSS, ale obráceně to neplatí.[29]

Jakmile zákazník stiskne na tlačítko “Přispět“, zobrazí se první stránka vyskakovacího okénka, kde bude informace o přání, na které se přispívá, a posuvná osa, na které si zákazník zvolí, kolik chce na přání přispět. Dále



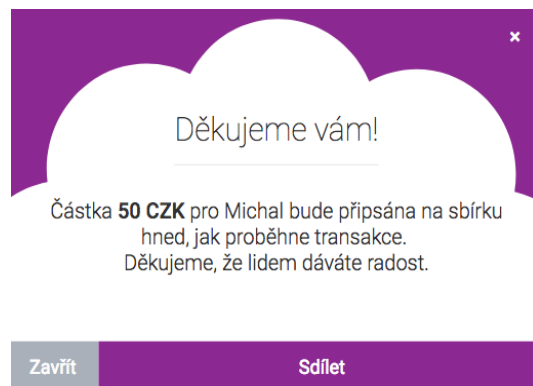
Obrázek 3.1: Krok 1 při procesu přispívání - výběr přispívané částky.

po zadání částky uživatel přejde na další stránku, kde už budou nabízené platební metody. Pro každou platební metodu jedno aktivační tlačítko, které při kliknutí odkáže zákazníka na zadání platebních informací.



Obrázek 3.2: Krok 2 při procesu přispívání – zvolení platební metody.

Po kladném zadání platebních údajů a zpracování transakce platebním procesorem bude zákazník propuštěn na poslední stranu vyskakovacího okna. Toto okénko má za úkol zákazníkovi poděkovat za provedenou platbu a informovat ho o tom, na čí přání přispěl a jaký obnos.



Obrázek 3.3: Krok 3 při procesu přispívání – poděkování zákazníkovi za platbu.

Dále budou popsány návrhy implementace jednotlivých platebních metod.

3.1 Návrh platby QR kódem

Implementace této platby se ukázala být jako nejjednodušší. Jediným úkolem aplikace ElateMe je v případě této platby vygenerovat validní QR kód s informacemi o platbě a následně odchytit pomocí volání API obchodníkovy banky, v tomto případě pomocí API Fio banky, jestli k dané platbě došlo či nikoliv.

Formát hodnoty QR kódu

Jak už bylo zmíněno v analytické části, QR kód dokáže zakódovat určité množství informací, kdy zvolenému generátoru předáme tyto informace v textové podobě a dostaneme QR kód. Formát tohoto textu, který se generátoru předává, bychom nemuseli brát v potaz, pokud bychom chtěli zakódovat pouze jednu informaci. My potřebujeme do QR kódu zakódovat informaci hned několik a to číslo účtu, částku, kterou uživatel přispěje, variabilní symbol a konstantní symbol platby. Z toho důvodu existuje standard, který určuje v jakém formátu by se měly bankovní informace do QR kódu posílat, aby z toho vzniknul validní QR kód, jenž umožní zákazníkovi provést platbu. Formát řetězce je navržen tak, aby byl kompaktní co se velikosti obsažených dat týče. Výhodou navrženého formátu je relativně dobrá lidská čitelnost a potenciální rozšiřitelnost o specifické atributy.

3. NÁVRH

Řetězec musí být vždy zahájen fixní hlavičkou SPD*. Následuje verze protokolu zakončena hvězdičkou, například 1.0*. Poté už řetězec obsahuje jednotlivé atributy platby ve formátu:

- $\{\text{klíč}\}:\{\text{hodnota}\}^*$

$\{\text{klíč}\}$ – Klíč daného atributu je zapsán vždy z množiny znaků [A-Z-]. Seznam přípustných klíčů je dlouhý a z toho důvodu uvedu níže v tabulce pouze ty, které byly použity pro náš účel.

$\{\text{hodnota}\}$ – Hodnota daného atributu může obsahovat libovolné znaky, ale musí být zároveň v přípustném formátu pro dané pole. Tento formát bude popsán v tabulce níže, kde budou vypsány použité klíče.

3.1. Návrh platby QR kódem

Klíč	Délka	Formát	Hodnota
Základní parametry, které platí globálně			
ACC	Max 46 znaků (IBAN + BIC)	IBAN, BIC	Hodnota se skládá ze dvou komponent oddělených znaménkem + První komponenta je číslo účtu ve formátu IBAN Druhou komponentou je identifikace banky neboli BIC ve formátu SWIFT podle ISO 9362. Tato druhá komponenta je volitelná.
AM	Max 10 znaků	Desetinné číslo. Tečka slouží jako oddělovač desetinných míst kde jsou povolena maximálně dvě desetinná místa	Výše částky platby.
CC	Právě 3 znaky	ISO 4217, 3 znaky, velká písmena	Měna platby
Rozšířené parametry, které se používají pouze v ČR			
X-VS	Max 10 znaků	Celé číslo	Variabilní symbol platby
X-SS	Max 10 znaků	Celé číslo	Specifický symbol platby

Tabulka 3.1: Seznam použitých parametrů pro vytvoření vyhovujícího QR kódu.[10]

Z parametrů popsaných v tabulce 3.1 se vytvoří textový řetězec s odpovídajícími hodnotami a ten se předá jako argument generátoru QR kódu.

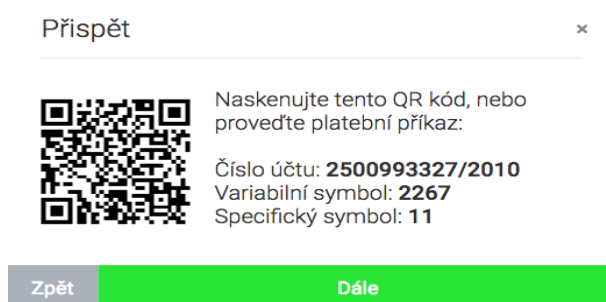
Generování QR kódu

Je mnoho způsobů jak v nynější době QR kódy generovat. Pro aplikaci ElateMe jsme se rozhodli použít JavaScriptový balíček pro generování kódů s názvem “qrcode.react“. Tento balíček nám umožňuje použít předdefinovanou

3. NÁVRH

komponentu napsanou v Reactu, která nám vygeneruje QR kód pro platbu. Jediným naším úkolem v tomto případě je komponentu importovat pro použití a předat ji povinný argument s názvem “value“. Tento argument, je textový řetězec obsahující platební informace ve správném formátu, pro generování QR kódu.

Poté co zákazník na stránce s nabízenými platbami, viz. obrázek 3.2, zvolí typ platby QR kódem, je spuštěno generování QR kódu a zákazník je přesměrován na speciální stránku ve vyskakovacím okénku pro zobrazení vygenerovaného QR kódu.



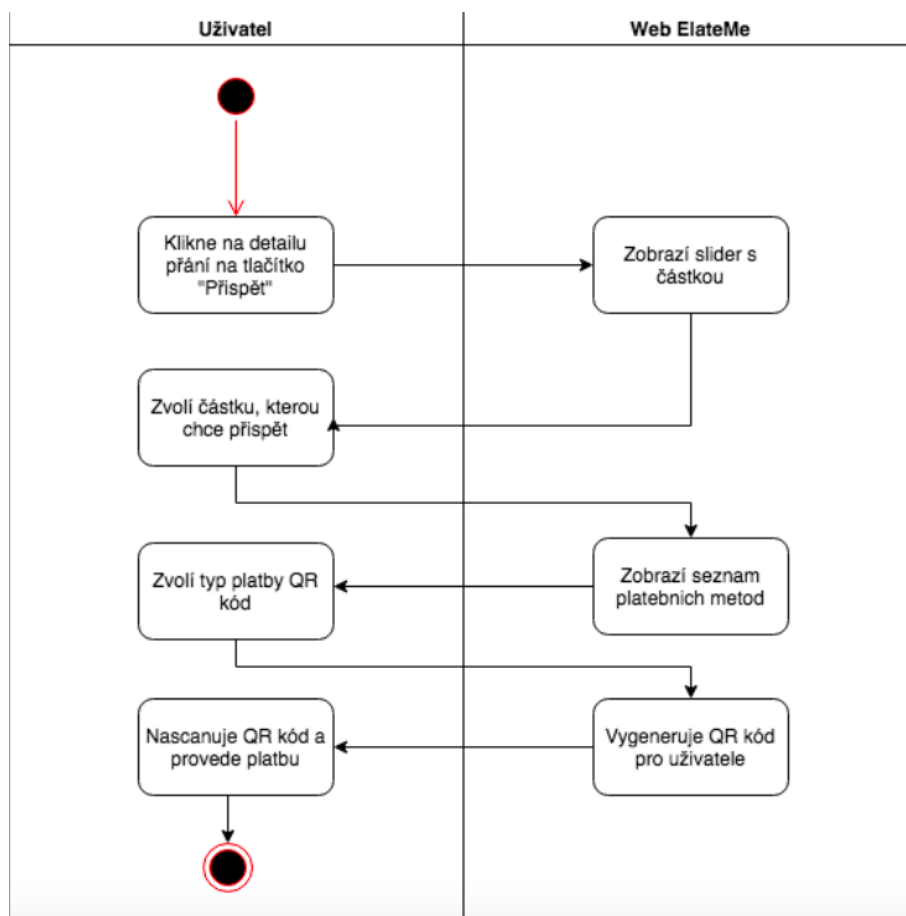
Obrázek 3.4: Detail vygenerovaného QR kódu k přání v aplikaci ElateMe.

Zde na detailu je ještě kromě QR kódu vidět číslo účtu, kam se bude částka posílat, a variabilní a specifický symbol platby, jak jsem popisoval výše v analytické části. Variabilní symbol je ID přání, na které se přispívá, a specifický symbol je ID přihlášeného uživatele, který provádí platbu. Tyto informace musí platba obsahovat, jelikož na tento typ platby neexistuje žádný platební procesor, který by nám automaticky potvrdil, zda platba proběhla či nikoliv. Zde je na zákazníkovi, aby si QR kód naskenoval a provedl platbu. Z toho důvodu je při vygenerování QR kódu na serverové části aplikace tato operace zapsána do databáze s dobou splatnosti 10 pracovních dnů. Na serverové části aplikace ElateMe běží program, který má na starost pomocí API Fio banky, posílat v určitých intervalech dotazy na stavy transakcí. Na základě údajů, jako jsou variabilní a specifický symbol platby, zjistí, zda uživatel provedl platbu v daném časovém intervalu. Časový interval 10 dní na provedení platby jsme nastavili v závislosti analýzy rychlosti zahraničních bankovních převodů na české bankovní účty v analytické části. Pokud platba proběhne do daného intervalu, budou na základě variabilního a konstantního symbolu informováni odpovídající uživatelé o úspěšném příspěvku na přání. Konkrétně podle variabilního symbolu, což je ID přání, informujeme uživatele, který přání založil, že někdo přispěl na přání a podle specifického symbolu informujeme uživatele, který příspěvek provedl, že vše dopadlo dobře. Pokud nedojde k zaplacení ze strany zákazníka do 10 dnů, program běžící na serveru danou situaci odchytlí, pošle uživateli, který měl platbu provést informaci a zrušení platby a

v případě, že platba dorazí se zpožděním, bude tato částka zákazníkovi, který platbu provedl, vrácena na účet.

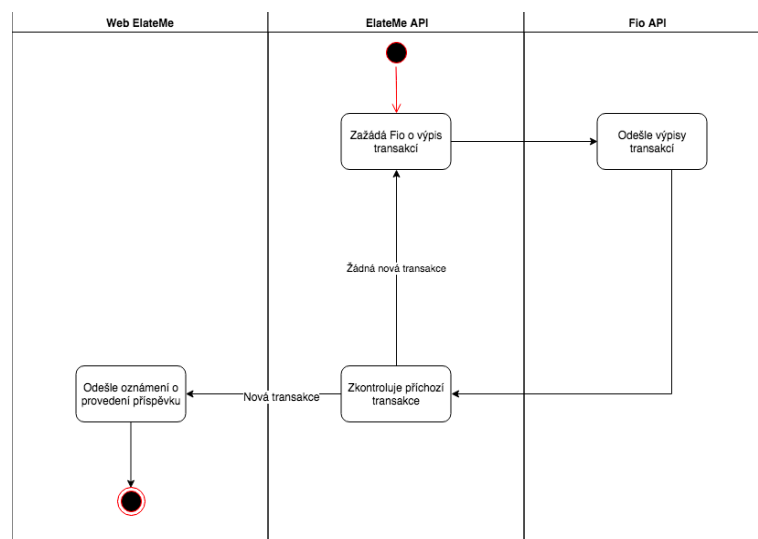
3.1.1 Diagram aktivit

Následující diagramy znázorňují aktivity, které se dějí na straně aplikace ElateMe při procesu platby pomocí QR kódu. Jsou zde dva diagramy, jelikož při této platbě probíhají dva procesy nezávisle na sobě, a z toho důvodu to bylo potřeba rozdělit na dvě části. Jeden diagram se týká klientské části aplikace ElateMe a druhý té serverové části.



Obrázek 3.5: Diagram aktivit klientské části platby pomocí QR kódu.

3. NÁVRH



Obrázek 3.6: Diagram aktivit serverové části platby pomocí QR kódu.

3.2 Návrh platby PayPal s využitím systému Braintree

Využitím společnosti Braintree pro zprostředkování platby pomocí PayPalu, se nám značně usnadní návrhová a implementační část této platební metody. Společnost Braintree dodává řadu vývojových nástrojů a výbornou dokumentaci pro snazší integraci platebních metod s aplikací. Jak už jsem popisoval v analytické části, společnost Braintree dodá jak serverovou část, tak klientskou část pro rychlou integraci s aplikací. Jelikož v mé bakalářské práci řeším pouze klientskou část plateb, nebudu se serverovou částí dodávanou společností Braintree více zabývat. Co se týče klientské části, je třeba uvést, že před tím než započne samotná integrace klientské části s aplikací, aby bylo možné přijímat PayPal platby, je zapotřebí mít vyřešených pár prerekvizit.

- Je třeba vytvořit, ověřit a propojit náš PayPal Business účet s Braintree účtem. Toto propojení se provede v kontrolním panelu Braintree účtu.
- Server ElateMe vygeneruje klientský token, pomocí kterého se následně na straně klienta ElateMe zahájí platební proces.

Pro základní použití komponent, které Braintree nabízí, je tyto komponenty potřeba do klientské části aplikace ElateMe nějakým způsobem importovat. Nabízejí se zde dvě možnosti, jak tyto komponenty importovat.

- Načíst komponenty pomocí scriptových tagů, které se používají v HTML pro importování JavaScriptových knihoven a balíčků.

3.2. Návrh platby PayPal s využitím systému Braintree

- Nainstalovat si komponenty v podobě JavaScriptových balíčků pomocí npm (node package manager) neboli správce JavaScriptových balíčků, které jsou volně stažitelné a připravené k použití.

Jelikož využíváme Reactu pro akce na klientské straně a celkově pro generování uživatelského rozhraní, rozhodli jsme se pro řešení v podobě JavaScriptových balíčků importovaných a nainstalovaných pomocí npm. Použití takového způsobu importu balíčků pro nás bude jednodušší s ohledem na používaný programovací jazyk.

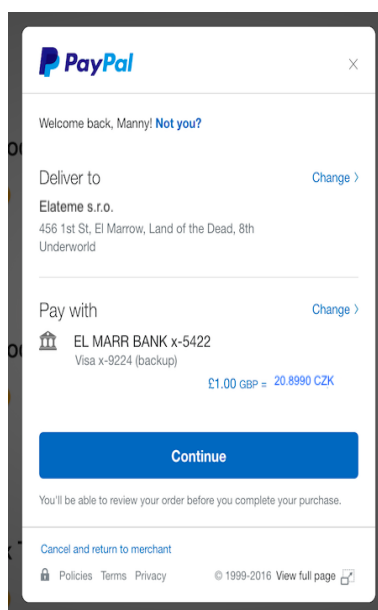
Pro navázání spojení s Braintree a vytvoření payment method nonce [2.6.1], který odešleme na server ElateMe, budeme potřebovat balíčky, kde jeden vyvinula společnost Braintree pro integraci klientské části v zákaznickově aplikaci a druhý společnost PayPal pro snadné a přehledné provedení platby. Tyto balíčky jsou

- **braintree-web** – slouží pro vytvoření instance klienta.
- **paypal-checkout** – slouží k zobrazení vyskakovacího PayPal okna, které provede zákazníka platebním procesem.

Po kliknutí uživatelem na tlačítko platby pomocí PayPalu na obrazovce s nabízenými platbami 3.2, dojde na straně klienta aplikace ElateMe k zaslání požadavku na server o vytvoření klientského tokenu. Dále se pomocí importovaného balíčku braintree-web vytvoří instance klienta. Klientský token, který byl obsažen v odpovědi serveru, je předán jako argument do funkce, které vytváří instanci klienta. Dalším argumentem je pak funkce, kde se vytváří logika, která zobrazí PayPal uživateli v podobě vyskakovacího okénka a následně zpracuje průběh dokončení platby pomocí PayPalu. Toto vyskakovací okénko je výsledkem balíčku paypal-checkout, kdy pokud je uživatel už přihlášen do svého PayPal účtu v jiném okénku prohlížeče, budou jeho kontaktní informace vyplněny automaticky a pokud ne, bude vyzván k přihlášení do PayPal účtu a okénko bude poté mít stejnou podobu. Jedním z argumentů funkce pro vytvoření tohoto okénka je objekt, který musí obsahovat dva povinné atributy a to hodnota platbu a měnu platby. Tyto informace jsou pak zobrazeny zákazníkovi a jsou neměnné. Dalším argumentem je pak funkce, která má na starost kontrolu výsledku transakce. Transakce může nabýt třech stavů a každý stav se musí řádně zpracovat.

- **onAuthorize** – Tento stav nastane, když zákazník potvrdí informace o transakci v PayPal okénku. Tento stav vyvolá funkci, která vygeneruje payment method nonce [2.6.1], který se odešle na server aplikace ElateMe. Server payment method nonce odešle na Braintree server, který odpoví zda transakce proběhla či nikoliv.

3. NÁVRH



Obrázek 3.7: Detail vyskakovacího okénka pro platbu PayPalem.[7]

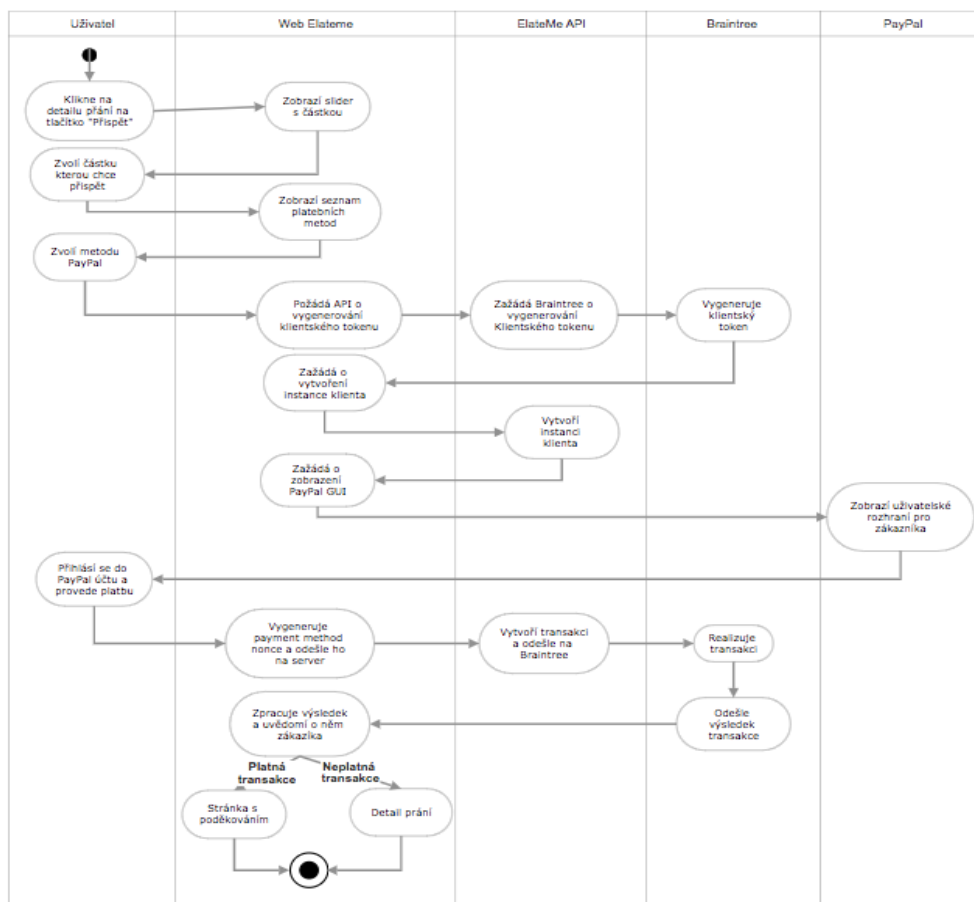
- **onCancel** – Tento stav nastane, když zákazník zruší transakci. To může nastat po stisknutí tlačítka “Cancel and return to merchant“ ve vyskakovacím PayPal okénku a nebo po stisknutí křížku v pravém horním rohu.
- **onError** – Tento stav nastane, pokud dojde k nějaké chybě v průběhu generování okénka nebo přímo při interakci uživatele s okénkem.

Poté, co server ElateMe dostane odpověď ze serveru Braintree o stavu transakce, odešle ji na klientskou část aplikace. Pokud vše proběhlo správně a transakce byla schválena, je zákazník přesměrován na stránku s poděkováním [3.3]. V případě, že nastala chyba při schvalování transakce, bude zákazníkovi sdělena příčina chyby a bude odkázán zpět na obrazovku s výběrem platebních metod [3.2].

3.2.1 Diagram aktivit

Následující diagram znázorňuje aktivity, které probíhají při zvolení platební metody PayPal.

3.3. Návrh platby platební kartou



Obrázek 3.8: Diagram aktivit platby PayPal.

3.3 Návrh platby platební kartou

Implementace této metody spadá převážně na serverovou část aplikace ElateMe. Budu popisovat, jak jsem postupoval pouze při návrhu klientské části této platební metody. Průběh zpracování dat a celého procesu, včetně části na serveru bude popsán níže, v odpovídajícím UML diagramu.

Mým jediným úkolem, co se týče této platební metody, bylo přeměřovat uživatele na URL, vygenerované ze strany ElateMe API. Poté co zákazník klikne na obrazovce s platebními metodami [3.2] na tlačítko “Kreditní karta“, dojde na klientské straně aplikace ElateMe k zaslání dotazu na ElateMe API o vygenerování URL adresy. Výsledkem dotazu je URL, na které je uživatel přeměřován. Na tomto URL se nachází webová stránka obsahující platební bránu, kterou pro účely provedení platební transakce pomocí kreditní karty poskytuje Fio banka, respektive společnost First Data. Na této stránce uživa-

3. NÁVRH

tel zadá informace ke své platební kartě. Na stránce je kromě formuláře pro vyplnění informací o kreditní kartě zobrazeno ID přání, na které je přispíváno, a informace o obchodníkovi. Dále zaručení Fio banky, že údaje o kartě nebudou sdíleny s obchodníkem a že služba je zabezpečena technologií 3D secure. Poté, co zákazník vyplní údaje o kreditní kartě, zmáčkne tlačítko “Zaplatit“ a

Fio banka Platební brána 3D Secure

Podporované karty:

Jméno na kartě:

Číslo karty (bez mezer):

Platnost karty (měsíc/rok): /

CVC2/CVV2 kód: CVC2/CVV2 kód najdete na zadní straně karty

Zaplatit 50.00 CZK [Zpět na objednávku](#)

Jméno obchodníka Wowe
ElateMe s. r. o.
Rybářská 716/24
110 00 Praha 1

Popis transakce
Prispevek na prani ID 2328

Vaše platební údaje nikdy nesdílíme s obchodníkem.

Verified by **VISA** **MasterCard** SecureCode [learn more](#)

Po vyplnění údajů o platební kartě, stisknete tlačítko "Zaplatit" a vyčkejte, než systém informace zpracuje (to může trvat až 1 minutu, neobnovujte stránku). V případě, že máte aktivovanou službu MasterCard® SecureCode™ nebo Verified by VISA, můžete být požádáni o poskytnutí dodatečných informací ohledně Vaší totožnosti (heslo nebo jiné informace). V případě, že nevíte uvedené heslo nebo jinou požadovanou informaci, prosím, kontaktujte Vaši banku.

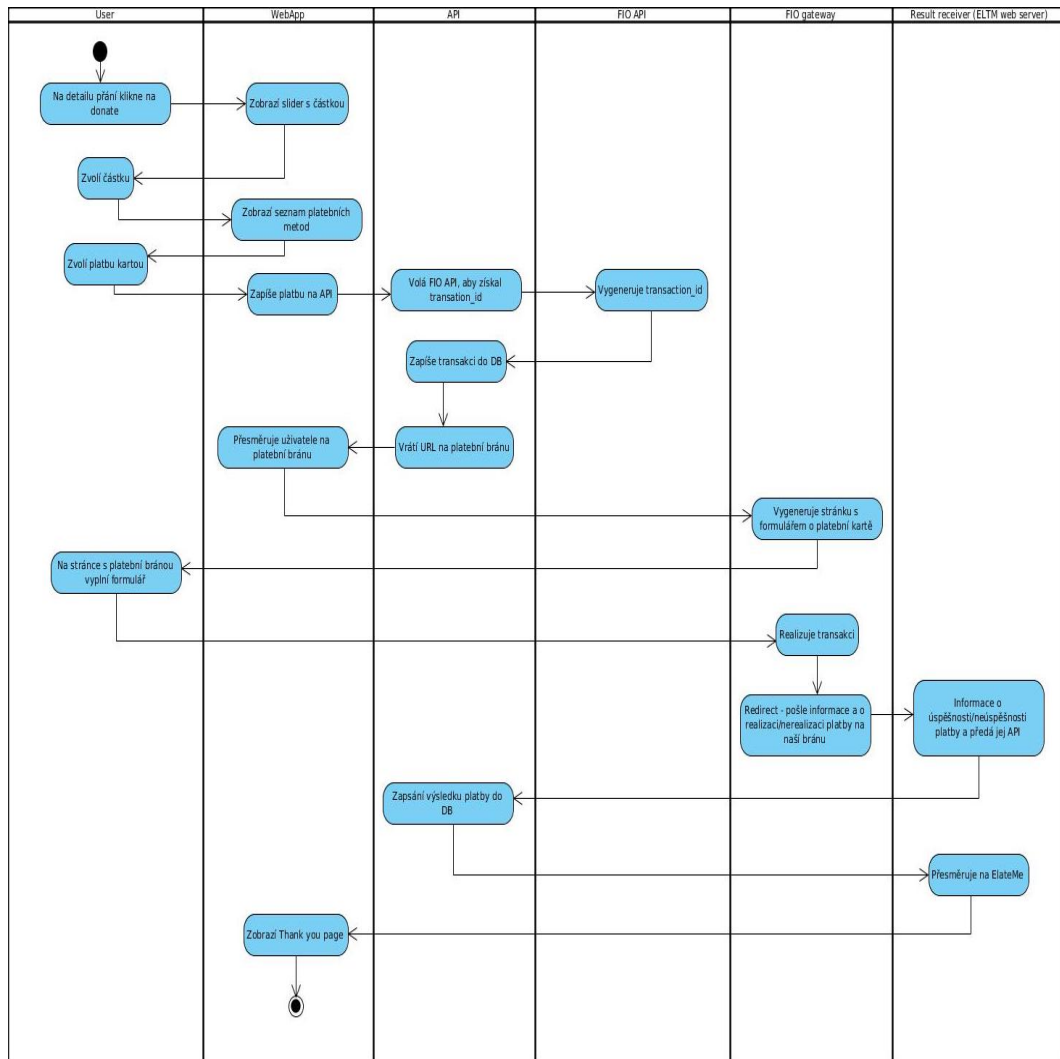
Stránka je bezpečná a je ve správě společnosti First Data International, partnera Fio banky, a.s.
Infolinka Fio banky: +420 224 346 800 (po - pá: 9-17 hod)

Obrázek 3.9: Detail platební brány dodávané společností First Data pro účely Fio banky.

po zpracování údajů je přeměrován zpět na stránku aplikace ElateMe, přesněji na obrazovku s poděkováním za příspěvek [3.3]. Pokud si zákazník platbu rozmyslí ve smyslu, že se rozhodne přispět menší nebo větší obnos a nebo z jakéhokoliv jiného důvodu, stačí aby stisknul tlačítko “Zpět na objednávku“ a dostane se zpátky na detail přání v aplikaci ElateMe. Přesměrování uživatele zpět na aplikaci ElateMe řeší opět serverová část aplikace.

3.3.1 Diagram aktivit

Následující UML diagram popisuje chování aplikace při procesu platby kreditní kartou. Popisuje jak část, která se odehrává na straně klienta ElateMe, tudíž část kterou budu implementovat, tak i serverovou část, na které pracují moji kolegové ve svých bakalářských pracích.



Obrázek 3.10: Diagram aktivit pro platbu kreditní kartou[8].

3.4 Návrh platby pomocí bitcoinu

Jak jsme se rozhodli v analytické části [2.4.1], tak jsme si jako poskytovatele služeb pro platbu pomocí bitcoinů vybrali společnost BitPay. Jeden z důvodů,

3. NÁVRH

proč jsme si tuto společnost vybrali, je výborná dokumentace a postup pro integraci platebního procesoru do aplikace. Tato společnost poskytuje knihovny, které už v sobě obsahují všechny potřebné metody a nám pouze stačí tyto metody volat a pracovat s jejich výsledky. Opět tedy použijeme balíček, který nainstalujeme pomocí npm (node package manager). Tento balíček nese název bitpay a obsahuje JavaScriptový soubor bitpay.js, který obsahuje všechny potřebné metody. Než začneme se samotnou integrací, musím nejdříve spárovat privátní klíč s obchodnickým bitpay účtem. Tento proces se nazývá párování.

Párování

Párování klíče s bitpay účtem se provádí z toho důvodu, že obchodník ve svém bitpay účtu má zřízenou bitcoin peněženku a číslo účtu, kam se bitcoiny z peněženky následně odesílají. Tím, že dojde k párování mezi klíčem a obchodnickým bitpay účtem, je při procesu platby, který se provádí na straně klienta aplikace ElateMe, automaticky vygenerována adresa obchodnickovy bitcoinové peněženky pro zákazníka. Tento proces se provádí pouze před prvním použitím BitPay API. Je vygenerován privátní klíč, který si následně obchodník pomocí příkazové řádky nebo pomocí svého bitpay účtu, spáruje s bitpay účtem. Privátní klíč, který byl použit pro párování, je uložen v souboru a je použit při každé další započaté transakci. Tento privátní klíč je vygenerován pro testovací prostředí. Pokud chce zákazník přejít do produkce, musí tento příznak nastavit v bitpay.js souboru a celý proces párování provést znova.

Poté, co dojde ke spárování klíče s bitpay účtem, je možno zahájit transakci. Proces transakce započne ve chvíli, kdy zákazník ve vyskakovacím okénku s platebními metodami [3.2] zvolí platbu bitcoiny kliknutím na tlačítko “Bitcoin“. Tuto akci odchytlí klíčová část aplikace ElateMe a jako první požádá ElateMe API o klíč. Klíč, který dostane jako výsledek API volání, je potřeba nejdříve zašifrovat, než dojde k samotnému použití a propojení s BitPay API. Po zašifrování klíče, dojde k vytvoření instance BitPay klienta. Tento klient slouží pro založení spojení s BitPay, vytvoření transakce a zobrazení uživatelského rozhraní. Při vytváření transakce je zapotřebí předat funkci objekt, jehož součástí je obnos, který bude zákazník přispívat, měna a notifikační URL, případně redirect URL.

- **Notifikační URL** – na toto URL bude BitPay API posílat formou POST metody změny v stavu transakce.
- **Redirect URL** – na toto URL bude zákazník přeměřován po dokončení procesu platby. Tento atribut je volitelný a používá se pouze pokud zákazník provádí dokončení transakce na odkazu vygenerovaném pomocí BitPay API.

Výsledkem vytvoření transakce je tak zvaný invoice. Invoice v překladu znamená faktura a obsahuje všechny důležité informace o transakci, kde nejdůležitějším údajem je ID transakce. Toto ID transakce je pak použito při odkázání uživatele na uživatelské rozhraní. Společnosti BitPay poskytuje tři způsoby, jak zákazníkovi zobrazit uživatelské rozhraní, kde se mu zobrazí informace o platbě.

- **Přesměrování na BitPay** – Tento způsob je implementačně nejjednodušší a jediným úkolem programátora je přesměrovat zákazníka na URL, které je obsaženo v invoice. Pokud ovšem bude využito tohoto postupu, je zapotřebí předat při vytváření transakce atribut `redirectURL`. Na tuto adresu bude totiž zákazník přesměrován po dokončení platby.
- **Modální okno** – Modální je okno je typ vyskakovacího okénka, které se zobrazí na aktuální stránce. Uživatel proto vůbec neopustí stránky aplikace ElateMe. Modální okno je vyvoláno voláním funkce obsažené v balíčku `bitpay` a jediným atributem funkce je ID transakce. Na základě tohoto atributu se vytvoří modální okno s informacemi o platbě. U toho způsobu můžeme kontrolovat stavy modálního okna, tj. můžeme provádět akce před tím než se otevře nebo potom co se zavře.
- **Vestavěné okno** – Tento poslední způsob dovoluje uživatelské rozhraní pro dokončení platby pomocí bitcoinů přímo vložit do webové aplikace ElateMe. Tato komponenta bude součástí HTML stránky stejně jako seznam platebních metod nebo seznam přání.

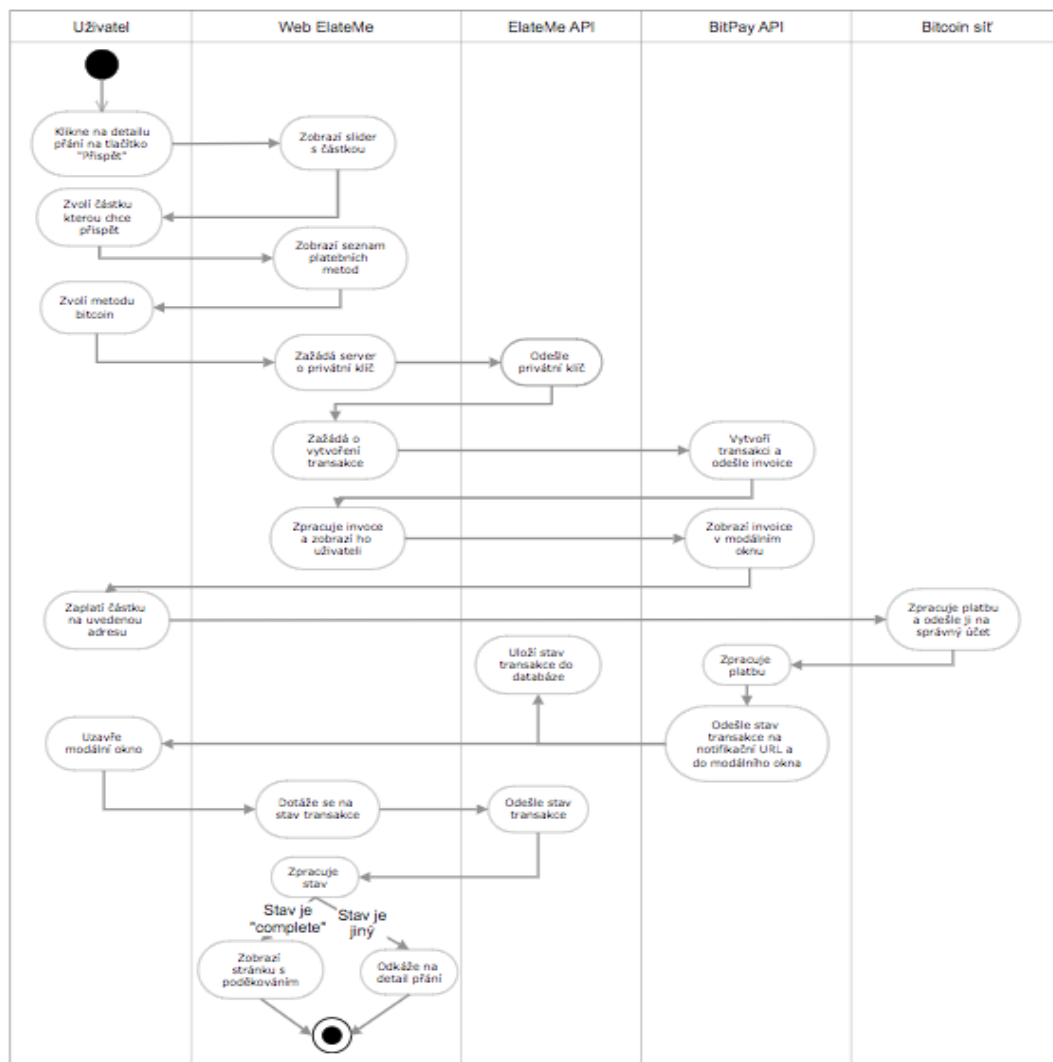
Jelikož chceme, aby zákazník po celou dobu transakce pokud možno zůstal na stránkách aplikace ElateMe a poslední možnost s vestavěným oknem se nám nehodí, využijeme proto zobrazení v modálním oknu.

Získáme z invoice ID transakce a pomocí funkce z `bitpay` balíčku vyvoláme modální okno pro uživatele. Uživatel projde obrazovky, které jsou k vidění v části 2.4.1, a na poslední obrazovce zvolí jednu ze tří možností jakým způsobem platbu, na kterou má 15 minut, provede. Pokud dojde ke změně stavu, bude o tom API ElateMe informováno díky notifikačnímu URL. Zároveň až uživatel pošle na uvedenou adresu bitcoiny, modální okno se automaticky zaktualizuje a informuje uživatele o správnosti provedení platby, ať už platba proběhla správně, bylo odesláno méně nebo více bitcoinů, než bylo potřeba, uplynulo 15 minut a platba nebyla detekována a nebo jestli nastal nějaký problém. Po zavření modálního okna zákazníkem, dojde k dotazu na ElateMe API na stav transakce. Pokud bude transakce ve stavu `“complete“`, což je jediný validní stav, je uživatel přesměrován na obrazovku s poděkováním [3.3]. Pokud transakce má jakýkoliv jiný stav, je o takové skutečnosti zákazník uvědoměn a přesměrován zpět na detail přání.

3. NÁVRH

3.4.1 Diagram aktivit

Následující UML diagram popisuje chování aplikace při procesu platby pomocí bitcoinů. Popisuje jak část, která se odehrává na straně klienta ElateMe, tudíž část kterou budu implementovat, tak i serverovou část, na které pracují moji kolegové ve svých bakalářských pracích.



Obrázek 3.11: Diagram aktivit pro platbu pomocí bitcoinů.

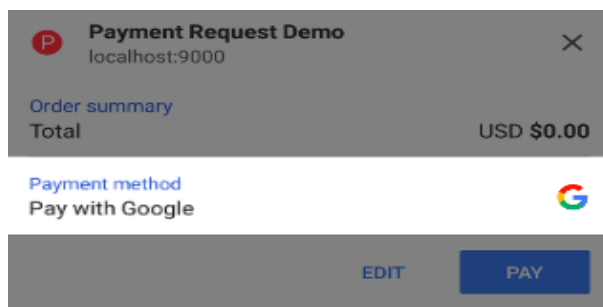
3.5 Návrh platby pomocí Google Pay

Jelikož společnost Braintree, kterou jsme využili už při procesu platby pomocí PayPal, dodává platební bránu i pro typ platby Google Pay, využijeme

opět služeb této společnosti. V balíčku dodávaném společností Braintree je obsažen způsob zprostředkování Google Pay platby uživateli pomocí Payment Requestu, tudíž nebude zapotřebí řešit uživatelské rozhraní nebo zabezpečování transakcí. O vše se zde postará Payment Request a Braintree. Naším úkolem bude pouze tyto služby v správném pořadí zavolat a odeslat na server payment method nonce [2.6.1].

Použití této metody je velmi podobné metodě platby pomocí PayPal, tudíž nebudu blíže znovu rozebírat generování klienta a podobné operace, které jsou popsány v kapitole s návrhem platební metody pomocí PayPal [3.2]. Při použití této metody je zapotřebí mít v uživatelském účtu Braintree povolené platby pomocí Google Pay. Co je zapotřebí zkontrolovat, je že daný prohlížeč podporuje Payment Request od Googlu. Payment Request je zabudován ve většině prohlížečů v různých verzích a pomocí JavaScriptové funkce lze velmi jednoduše zkontrolovat zda daný prohlížeč tuto funkci podporuje. Pokud prohlížeč Payment Request podporovat nebude, nebude tato možnost platby pro zákazníka viditelná.

Proces začne ve chvíli, kdy zákazník klikne na tlačítko “Google Pay“ na obrazovce s platebními metodami [3.2]. Dojde k vytvoření instance klienta na straně klientské části aplikace ElateMe a následně za použití této instance k vytvoření transakce pro Google Pay. Při vytváření komponenty Payment Requestu budou funkce předány pouze vybraná částka a měna. Payment Request zobrazí vyskakovací okénko s zobrazením dostupných platebních metod, kde v tomto případě bude zobrazeno tlačítko s platbou Google Pay. Poté, co uživatel klikne na tlačítko a dokončí platbu, bude výsledkem pay-



Obrázek 3.12: Detail zobrazení Google Pay metody pomocí payment requestu. [9]

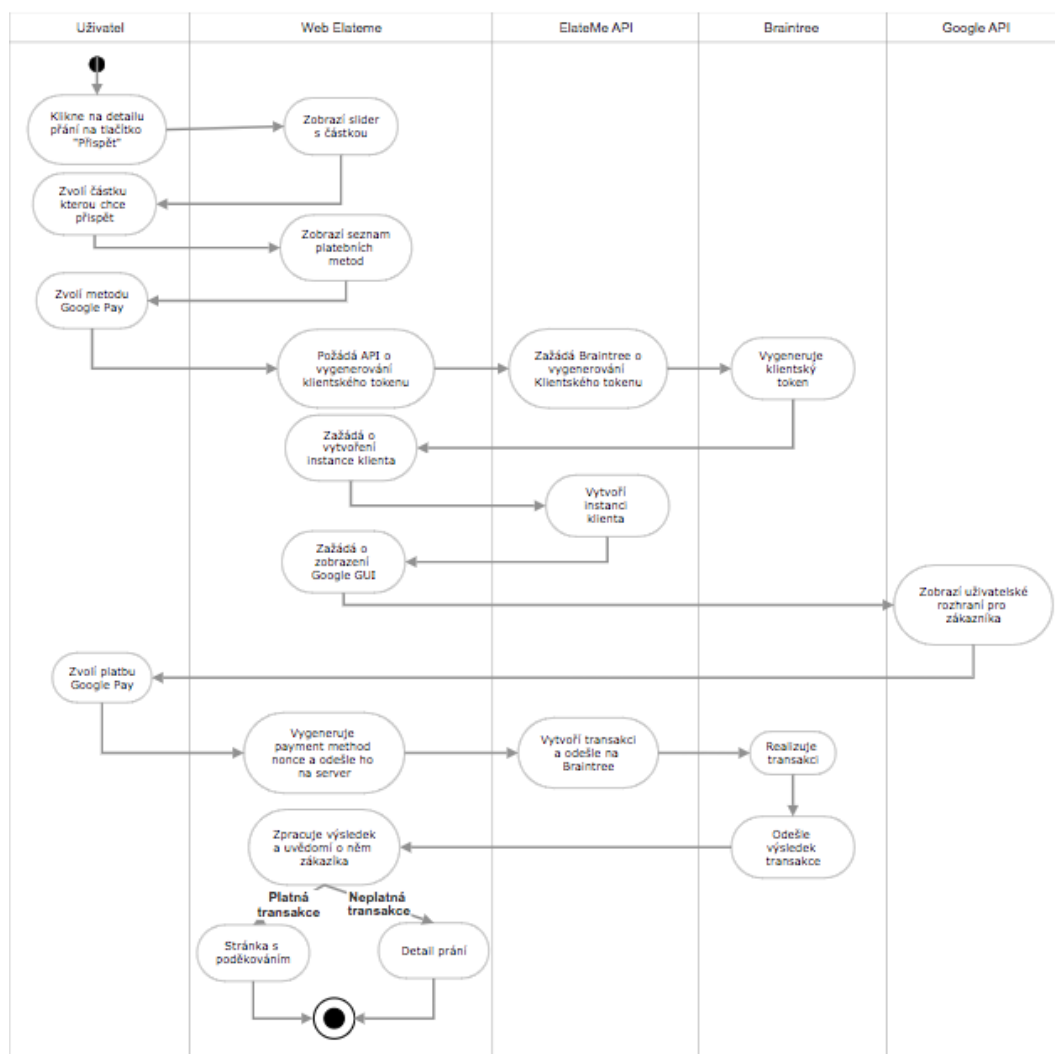
ment requestu payment method nonce [2.6.1], který zpracujeme a odešleme na serverovou část aplikace ElateMe pomocí API volání. Server odešle payment method nonce na Braintree API. Braintree transakci zvaliduje a po komunikaci s Googlem odešle výsledek transakce. Server odpověď zpracuje a pře pošle ji klientovi. Pokud bude odpověď kladná, tj. odpověď že transakce proběhla v pořádku, bude uživatel přeměrován na stránku s poděkováním

3. NÁVRH

[3.3]. V opačném případě bude zákazník uvědoměn o selhání transakce a bude odkázán na detail přání.

3.5.1 Diagram aktivit

Následující diagram znázorňuje aktivity, které probíhají při zvolení platební metody Google Pay.



Obrázek 3.13: Diagram aktivit pro platbu Google Pay za použití systému Braintree.

Implementace

V této části bakalářské práce využiji toho, co jsem rozebral a popisoval v kapitolách Analýza a Návrh a přiblížím čisté použití knihoven, funkcí a celkově jakým způsobem jsem vybrané platby implementoval do aplikace ElateMe.

Při použití Reactu, se snažíme co nejvíce vzhled stránky rozdělovat do jednotlivých komponent, kde každá komponenta zaobaluje určitou část uživatelského rozhraní. Každá platební metoda bude proto mít pro sebe samostatnou komponentu, kde veškeré operace týkající se určité platby, budou prováděny právě z dané komponenty. Co se týče ukládání a získávání dat z databáze, spadá tento úkol na server aplikace ElateMe. Z klientské části aplikace budeme posílat GET, POST nebo PUT metody s odpovídajícími endpointy pro získávání a odesílání dat. API aplikace ElateMe se nachází na adrese 'https://api.elateme.com' a na tuto adresu budou navázány odpovídající endpointy. API ElateMe využívá HTTPS protokolu, což je protokol spojující protokol HTTP a protokol SSL nebo TLS v jedno, a zajišťuje bezpečnost přenášených dat. Nemusíme se tudíž obávat odposlechnutí posílaných dat při volání API. Při volání API se budou z kódu používat odkazy na odpovídající endpointy a vytvoření celého URL má na starost část kódu, kterou jsem neimplementoval, tudíž zmíním pouze, že na konci je URL předána funkci `fetch()`, což je funkce dodávána opět správcem JavaScriptových balíčků neboli npm a má na starost získat odpověď z API podle zadaného URL. Tudíž pokud z kódu voláme API s endpointem `/donations/paypal`, bude mít URL předávané do funkce `fetch()` hodnotu `'https://api.elateme.com/donations/paypal'`.

4.1 Implementace platby pomocí QR kódu

Jak už bylo zmíněno v části Návrh, jediné, co je zde zapotřebí, je správně vygenerovat QR kód pro uživatele a informace o transakci odeslat na API ElateMe pomocí POST metody. Pro použití QR generátoru importujeme komponentu z balíčku `qrcode.react`.

```
import QRCode from 'qrcode.react';
```

Komponenta se poté použije uvozením do klasických HTML tagů jako `<QRCode />`. Tato komponenta přijímá jeden povinný atribut a tím je “value“, což je textový řetězec obsahující hodnotu, kterou chceme do QR kódu zakódovat. Komponenta má další volitelné atributy, jako jsou způsob zobrazení, velikost QR kódu, barva QR kódu a další. Plný výčet atributů zobrazuje následující tabulka. Textový řetězec, který předáme atributu “value“, sestavíme ve for-

atribut	typ	výchozí hodnota
value	string	
renderAs	string ('canvas' 'svg')	'canvas'
size	number	128
bgColor	string (CSS color)	"FFFFFF"
fgColor	string (CSS color)	"000000"
level	string ('L' 'M' 'Q' 'H')	'L'

Tabulka 4.1: Seznam atributů pro komponentu `<QRCode />`. [11]

mátu podle standardu pro kódování platebních informací z klíčů a atributů, které jsou popsány v tabulce 3.1. Do QR kódu budeme chtít zakódovat číslo účtu, částku, měnu, variabilní symbol a specifický symbol platby. Pro ukázkou jak by výsledný řetězec měl vypadat použijí následující data

- **Číslo účtu ve formátu IBAN:** CZ0320100000002500993327
- **Výše částky platby:** 50
- **Měna platby:** CZK
- **Variabilní symbol (ID přání):** 283
- **Specifický symbol (ID uživatele):** 11

Za použití odpovídajících klíčů a atributů, by měl řetězec s testovacími daty vypadat následovně

```
const QRvalue = 'SPD*1.0*ACC:CZ0320100000002500993327*AM:50*CC:CZK*X-VS:283*X-SS:11'
```

Dále pak máme tento řetězec uložený v proměnné `QRvalue` a stačí jí předat komponentě pro generování QR kódu.

```
<QRCode
  value={QRvalue}
/>
```

Výsledkem tohot kódu je pak použitelný QR kód, který uživatel nascanuje a provede platbu v mobilní aplikaci své banky. Zároveň při vygenerování kódu jsou informace o transakci vloženy do objektu a poslány do funkce, která tyto informace převede do formátu JSON, a na základě endpointu, který funkci bude předán jako argument, odešle na API. Informace jsou odesílány pomocí metody POST s endpointem ve tvaru `/donations/QRpayment/transaction` a odesílanými daty, což bude částka, ID přání a ID uživatele, který platbu provedl. Program, který běží na serveru aplikace ElateMe, jenž má na starost kontrolu výpisu z účtu zda nedošlo k zaplacení transakce, má na starost někdo jiný, a proto implementaci toho programu zde nebudu popisovat. Stejně je tomu tak s oznamováním uživatelů o přispění na přání či o nezdaru transakce.

4.2 Implementace platby kreditní kartou

Implementace platby kreditní kartou se ukázalo být jako implementačně nejjednodušší platební metoda z mého pohledu práce. Jelikož se značná část implementace odehrává na serveru aplikace ElateMe, implementace na klientské části znamenala zavolat pouze API pro získání URL a na toto URL zákazníka odkázat, aby zadal informace o kreditní kartě a dokončil platbu. Získávání informací o kreditní kartě opět nemá na starost klientská část aplikace ElateMe, je to platební brána od společnosti First Data, která poté dané údaje také zpracovává. Není dokonce třeba používat ani žádných JavaScriptových balíčků. Pro volání API se používá endpoint `/donations/cardpayment`. Výsledkem toho volání jsou pak data, ve kterých je obsaženo požadované URL, na které je zákazník odkázan, aby dokončil platbu. Veškeré volání vypadá nějak následovně.

```
const data = {
  user ,
  wish ,
  value
}

callCardPayment ( '/donations/cardpayment' , data )
. then ( ( data ) => {
  window . location . assign ( data . url );
} )
```

Funkce `window.location.assign()` má na starost přesměrovat uživatele na stránku s platební bránou. Dále data, uložená v objektu jsou

- **user** – instance uživatele, která má v sobě atribut `access_token`, který je použit při volání API jako přístupový údaj
- **wish** – instance přání, která má v sobě atribut `wish_id`, což je ID přání
- **value** – hodnota přispívané částky

4.3 Implementace platby pomocí bitcoinů

K implementaci platby pomocí bitcoinů budou zapotřebí tři JavaScriptové balíčky. Jeden pro vytvoření instance klienta pro spojení s bitpay API druhý pro zakódování API klíče při vytváření instance klienta a třetí pro zobrazení klientem vytvořeného invoice v modálním oknu.

```
import * as bitpayClient from 'bitpay-rest';
import * as bitAuth from 'bitauth';
import * as bitpay from 'bitpay';
```

Dále za pomoci těchto balíčků můžeme přejít přímo k vytvoření instance klienta. Nejdříve je tedy zapotřebí zakódovat API klíč, aby při případném útoku nedošlo k získání API klíče. K zakódování slouží funkce `decrypt()`, které je zapotřebí předložit 3 argumenty.

- **heslo** – Tento argument je volitelný a v případě, že pro zakódování je nastaveno heslo, je tento argument vyplněn. V opačném případě je vyplněn prázdný řetězec.
- **API klíč** – API klíč, který je vygenerován z příkazové řádky nebo na BitPay účtu zákazníka. Tento API klíč je v tomto případě pro testovací účely. Později až bude vše otestované se zamění za klíč určený pro produkci.
- **kódování** – Způsob jakým bude klíč zakódován.

Volání funkce vypadá poté následovně a rovnou tím pádem můžeme vytvořit i instanci klienta za pomoci funkce `createClient()`, jejímž jediným argumentem je privátní klíč.

```
var privateKey = bitAuth.decrypt(
  ' ',
  API_ACCESS_TOKEN,
  'utf8'
);
var bitpay_client = bitpayClient(privateKey);
```


Poté můžeme vytvořit invoice a díky tomuto invoicu zobrazit uživateli modální okno pro dokončení bitcoin platby. V následujícím kódu se provede celé řešení akce pro platbu pomocí bitcoinů. Funkce `on()` zařídí započatí akce. Funkce `as()` udá, zda je instance klienta typu `'pos'`, což je zkratka pro termín point of sale a nebo `'merchant'`. Toto jsou dva typy API tokenů, jejichž typ zvolí obchodník při vytváření API klíče v BitPay účtu. Funkce `post()` zažádá o vytvoření invoicu s přiloženými daty a výsledkem je buď chyba nebo validní invoice. Pokud bude invoice validní, získáme z něj ID transakce a zobrazíme modální okno pro uživatele. Dále definujeme funkci `onModalWillLeave()`, která se zavolá po zavření modálního okna uživatelem. V této funkci odešleme dotaz na API ElateMe o stav dané transakce v podobě GET metody na endpoint `'/donations/bitcoin/transaction'`. Výsledkem toho volání bude stav transakce, který když bude `'complete'`, pustíme zákazníka na stránku s poděkováním a v opačném případě na detail přání.

```

bitpay_client.on('ready', () => {
  var data = {
    price,
    currency,
    notification_url
  }

  bitpay_client.as('pos').post('invoices',
    data,
    (err, invoice) => {
      if(err) {
        moveToWishDetail(err);
      } else {
        bitpay.showInvoice(invoice.id);

        bitpay.onModalWillLeave(() => {
          getTransactionStatus(
            '/donations/bitcoin/transaction',
            invoice.id).then((response) => {
              if(response.status === 'complete') {
                showThankYouPage();
              } else {
                moveToWishDetail(response);
              }
            })
        })
      }
    })
});

```

Funkce `showThankYouPage()` přesměruje uživatele na stránku s poděkováním [3.3], v opačném případě bude stejně jako v případě, že došlo k chybě při vytváření invoice zavolána funkce `moveToWishDetail()`. Tato funkce zpracuje argument, zda-li jde o chybu či o nevalidní stav a uvědomí zákazníka o příčině chyby a přesměruje ho na detail přání. [30]

4.4 Implementace platby PayPal poháněná systémem Braintree

Pro implementaci této platby poskytuje společnost Braintree velmi dobrou dokumentaci i vývojové nástroje. Bude potřeba dvou JavaScriptových balíčků dodávaných společnostmi Braintree a společností PayPal. Jedná se o balíčky `braintree-web` a o balíček `paypal-checkout`. Jeden slouží k vytvoření instance klienta a druhý k zavolání PayPal okénka. Balíčky importujeme následujícím způsobem.

```
import * as braintree from 'braintree-web';
import * as paypal from 'paypal-checkout';
```

Další věc, co je třeba provést, je získat API klientský klíč, který je vytvořen při zakládání Braintree účtu a je generován na serverové straně ElateMe aplikace. Klientský API klíč získáme a uložíme voláním API ElateMe pomocí GET metody a endpointu `/donations/braintree/token`.

```
var API_KEY = null;

getBraintreeClientKey('/donations/braintree/token')
  .then((response) => {
    API_KEY = response.token;
  })
```

PayPal dodává společně s uživatelským rozhraním pro dokončení platby i svoje PayPal tlačítko, po jehož stisknutí se akce spustí. Je proto důležité mít v HTML části kódu `div` s ID, na který se poté bude odkazovat při vytváření instance tlačítka, což může vypadat nějak takto.

```
<div id="paypal-button"></div>
```

Poté můžeme už přímo vytvořit instanci klienta za pomocí funkce `client.create()` z balíčku `braintree-web`. Tato funkce jako první argument dostane API klíč a druhým argumentem je callback funkce, kde se vytvoří instance PayPalu a zavolá se komponenta pro zobrazení uživatelského rozhraní. Callback funkce je vlastně parametr funkce, jehož hodnota je funkce. Celkové zobrazení uživatelského rozhraní od PayPal má na starost funkce `Button.render()` z balíčku `paypal`. Tato funkce v sobě má pět povinných atributů, které se postupně volají pokud nastane určitá situace.[31]

- **env** – Tento atribut slouží k určení zda se jedná o ostrý provoz neboli 'production' a nebo zda se jedná pouze o testovací prostředí což je 'sandbox'. Více o tomto v kapitole Testování.
- **payment** – V tomto atributu je uložena funkce, která má na starost vytvoření instance platby.
- **onAuthorize** – Tento parametr se zavolá pokud se vytvoří instance platby správně. Dále se pomocí funkce tokenizePayment() instance platby zakóduje a výsledkem této funkce je datová struktura obsahující payment method nonce. Payment method nonce je pak odeslán na server ElateMe pomocí POST metody a endpointu /donations/braintree/purchase.
- **onCancel** – Obsahuje funkci, která provede akci pokud uživatel zruší komponentu platby.
- **onError** – Zavolá funkci, pokud nastane nějaká chyba v průběhu transakce.

```
braintree.client.create({
  API_KEY
}).then((clientInstance) =>{
  braintree.paypalCheckout.create({
    client: clientInstance
  });
}).then((paypalCheckoutInstance) => {
  paypal.Button.render({
    env: 'production', //nebo 'sandbox'

    payment: function() {
      return paypalCheckoutInstance.createPayment({
        flow: 'checkout',
        amount,
        currency,
        enableShippingAddress: false
      });
    },

    onAuthorize: function(){
      return paypalCheckoutInstance.tokenizePayment(data)
        .then((payload) => {
          sendNonceToServer(
            '/donations/braintree/purchase',
            payload.nonce,
            payload.amount
          );
        });
    }
  },
  {
    container: '#paypal-button-container',
    style: 'button',
    text: 'Pay with PayPal'
  });
});
```

```
        )
      });
    },

    onCancel: function(data){
      moveToWishDetail(data);
    }

    onError: function(err){
      moveToWishDetail(err);
    }
  }, '#paypal-button ');
}).then(() => {
  moveToWishDetail();
}).catch((err) => {
  moveToWishDetail(err);
});
})
```

4.5 Implementace platby Google Pay

Pro tuto metodu opět využijeme služeb společnosti Braintree, která funguje jako platební procesor i při platbě pomocí Google Pay. Společnost poskytuje dvě možnosti, jak platbu implementovat. První možností je ryze jako Google Pay bez využití Payment Requestu a druhá možnost je za využití právě Payment Requestu. Jelikož nám jde o zavedení Payment Requestu, tak využijeme přesně tuto možnost. Implementačně je tato metoda velmi podobná metodě PayPal [4.4], nebudu proto popisovat získání API klíče a vytvoření klienta což bylo popsáno v minulé kapitole. K vytvoření této platby použijeme balíček, který už jsme použili dříve, a to balíček braintree-web. Tento balíček obsahuje funkci `paymentRequest.create()`, která vrátí instanci vytvořené platby. Na tuto instanci použijeme opět funkci `tokenize()`, abychom zakódovali data o platbě, jako je výše přispívané částky a měna. Výsledkem této funkce je opět `payment method nonce`, který pomocí POST metody odešleme na server aplikace ElateMe na endpoint `/donations/braintree/purchase`.

```
import * as braintree from 'braintree-web';

if(window.PaymentRequest) {
  braintree.client.create({
    API_KEY
  }).then((clientInstance) => {
    braintree.paymentRequest.create({
```

```
        client: clientInstance
    }).then((instance) => {
        instance.tokenize({
            details: {
                total: {
                    label: 'Total',
                    amount: {
                        currency,
                        value
                    }
                }
            }
        })
    }).then((payload) => {
        sendNonceToServer(
            '/donations/braintree/purchase',
            payload.nonce,
            payload.amount
        );
    }).catch((err) => {
        moveToWishDetail(err);
    });
}).catch((err) => {
    moveToWishDetail(err);
});
}
} else {
    //dont show the payment method
}
```

Pokud zákazníkův prohlížeč nepodporuje PaymentRequest API, nebude zobrazena platební možnost pomocí Google Pay. Dále pokud nebude uživatelův prohlížeč podporovat Google Pay, zobrazí se v Payment Requestu výběr platby kreditní kartou jako přednastvená možnost, která je doplněna automaticky. [32]

Testování

Otestování platebních metod je pro aplikaci ElateMe velmi důležité, jelikož to poslední, co bychom chtěli, je, aby docházelo ke ztrátám peněz klientů. Platební metoda platební karta a platba pomocí QR kódu bezchybně pár měsíců na produkční verzi aplikace ElateMe funguje a testovat tyto metody bez použití opravdových peněz není možné. Byly zaplacené malé částky peněz pomocí platby platební kartou a platby QR kódem. Částky byly připsány na účet obchodníka a vše proběhlo tak, jak se předpokládalo. Platební metodu pomocí bitcoinu nebylo zatím možné otestovat i přes to, že společnost BitPay, kterou využíváme jako platebního procesora, dodává testovací prostředí pro testování transakcí. Není možné tuto metodu otestovat z toho důvodu, že při zakládání BitPay účtu je třeba vyčkat na schválení od společnosti, zda je zákazník doopravdy validním obchodníkem a zda se nejedná od podvrh. Toto schválení ElateMe zatím nedostalo a z toho důvodu zatím není možné provádět transakce jak reálné tak testovací. Zbývají nám tedy poslední dvě metody a to PayPal a Google Pay. Obě dvě platební metody využívají jako platebního procesora společnost Braintree a mají velmi podobnou implementaci. Není tudíž třeba testovat fungování obou dvou metod jelikož mají stejný základ a liší se pouze v třetích stranách, pro které je platební procesor určený. Třetími stranami mám na mysli PayPal a Google. Braintree pro účely testování vyvinulo vlastní testovací prostředí nazývané Sandbox, což je skoro identickou replikou klasického produkčního prostředí. Google Pay se používá na mobilních platformách android a z toho důvodu, že nevlastním telefon s operačním systémem android, nemohu danou metodu řádně otestovat.

Pro otestování jsme si tedy zvolili platební metodu PayPal. Zvolili jsme PayPal z toho důvodu, že Braintree je dceřinou společností společnosti PayPal a tyto dvě platformy spolu vzájemně úzce komunikují. Jelikož je pro provozování potřeba kromě Braintree účtu i PayPal účet, dodává společnost PayPal také vlastní sandbox účty za účelem testování, které se spojí s Braintree sandbox účtem a může zde dojít k otestování platby. PayPal testovací účet také poskytuje testovací kreditní karty s nastavitelným obnosem a díky

tomu můžeme proces platby pomocí PayPal reálně nasimulovat.

5.1 Testování platební metody PayPal

Jak už jsem výše zmínil, tak společnost Braintree dodává testovací prostředí pro potřeby zákazníka k otestování správnosti platebních metod a procesů. Toto testovací prostředí se nijak zvláště neliší od produkčního až na to, že nedojde doopravdy k použití reálných peněz. K tomu proto existují na Braintree dva typy účtů a to testovací a produkční účet. Tyto dva účty spolu nejsou nijak provázané. Nemůže se tedy stát, že by došlo k úniku peněz z produkčního účtu nebo případně k získání peněz z testovacího na produkční účet. Hlavní rozdíl způsobuje API klientský klíč, který je rozdílný pro testovací potřeby a pro reálný provoz. Na základě toho klíče pozná Braintree, o jaký typ účtu se jedná.

PayPal platbu je možno pomocí Braintree testovat dvěma způsoby.

- **Mockovací PayPal testování**

- Chová se podobně jako produkce, ale nedovoluje end-to-end testování. End-to-end testování je způsob testování aplikace od počátku do jejího konce.
- Používá Braintree testovací hodnoty pro nasimulování PayPal odpovědi v Braintree sandbox prostředí.
- Umožňuje potvrdit, že klientská i serverová strana konfigurace jsou korektní a že vracejí správné odpovědi na požadavky.
- Nevrací data do PayPal testovacího účtu. Účty PayPal a Braintree nejsou provázané.

- **Provázané PayPal testování**

- Je zapotřebí dodatečné nastavení.
- Přímo propojí Braintree testovací účet s PayPal testovacím účtem.
- Posílá data do obou dvou účtů.
- Umožňuje otestovat úplnou integraci PayPal platební metody do aplikace.

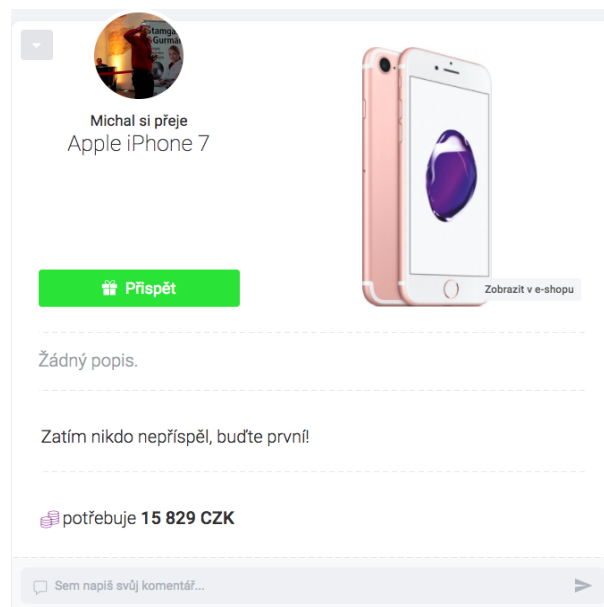
Jelikož chceme otestovat průběh platby od začátku do konce, což znamená, že chceme nasimulovat zvolení platby PayPal uživatele, jeho přihlášení do PayPal testovacího účtu a zaplacení obnosu, využijeme druhý způsob a to provázané PayPal testování. Obnos, který uživatel zaplatí, by se poté měl strhnout z PayPal testovacího účtu a připsat se na testovací účet obchodníkovi. Předtím, než se dostaneme k samotnému testování, je ale nejdříve zapotřebí propojit testovací účty PayPal a Braintree. Propojení probíhá tak, že si uživatel nejdříve vytvoří

jak testovací účet na Braintree tak testovací účet na PayPal účtu. Na Braintree účtu se pak uživatel dostane na úvodní stránku kde se dokliká na stránku PayPal způsobem Settings > Processing > Payment Methods > PayPal. Tam bude možnost propojit Braintree účet s PayPal účtem a to tak, že uživatel zadá emailovou adresu, obchodníkovu ID a klientský tajný klíč. Všechny tři informace uživatel získá na PayPal testovacím účtu. Pak stačí kliknout na tlačítko “Link PayPal sandbox“ a účty jsou připravené pro testování.[33]

Testování bude prováděno klasickým uživatelským klikáním, kdy jednotlivé kroky budou popsány v testovacím scénáři. Testovací scénář je rozdělen na jednotlivé kroky, kdy v každém kroku je napsáno, jaká akce by se měla provést a jaký by měl odpovídající výsledek na danou akci.

5.1.1 Testovací scénář

V tomto testovacím scénáři bude popsáno, jak by měl správně fungovat celý proces platby od rozhodnutí zákazníka pro příspěvek po připsání obnosu na testovací PayPal účet obchodníka. Pro testovací potřeby jsem vytvořil v aplikaci ElateMe přání, na které budu pomocí svého testovacího PayPal účtu chtít přispět. Vytvořením a přihlašovaním do aplikace se moje bakalářská práce nezabývá, proto zde postup testování těchto funkcionalit popisovat nebudu. Testování počíná detailem přání, které jsem vytvořil pro testování, kdy přání je Apple iPhone 7, ID přání je 2339, založil ho Michal (testovací jméno) a je potřeba na toto přání vybrat 15 829 českých korun. Na obrázku 5.1 můžeme



Obrázek 5.1: Počáteční krok testování, detail přání.

5. TESTOVÁNÍ

vidět počáteční stav testovacího scénáře, kdy jsme na obrazovce s detailem přání.

Krok 1

- **Akce** – Kliknu na tlačíto “Přispět“.
- **Výsledek** – Zobrazí se vyskakovací modální okno se sliderem pro výběr přispívané částky. Vyskakovací okno bude vypadat stejně jako je na obrázku 3.1, kde bude napsáno na jaké přání se přispívá a komu.

Krok 2

- **Akce** – Zvolím částku 500 Korun českých a kliknu na tlačítko “Dále“.
- **Výsledek** – Zobrazí se stránka s dostupnými platebními metodami stejně jako je tomu na obrázku 3.2.

Krok 3

- **Akce** – Zvolím platbu pomocí PayPalu a kliknu na tlačítko “Dále“.
- **Výsledek** – Objeví se vyskakovací okénko PayPalu.

Krok 4

- **Akce** – Jelikož aplikace ElateMe využívá testovací prostředí Braintree stejně jako testovací účet PayPal, použiji také testovací účet PayPal, ze kterého chci provést platbu. Přihlásím se proto do PayPal účtu a vyberu možnost zaplatit testovací kreditní kartou.
- **Výsledek** – Zobrazí se shrnutí platby, kde bude zobrazena částka, která se bude platit, v tomto případě 500 Korun českých. Dále zde bude účet, na který se částka posílá, účet aplikace ElateMe, který tam je předán v kódu a tato informace je neměnná. Poslední zobrazenou informací bude moje zvolená testovací kreditní karta.

Krok 5

- **Akce** – Odsouhlasím a potvrdím platbu tlačítkem “Zaplatit“.
- **Výsledek** – Zobrazí se stránka s poděkováním, kde mi bude zobrazeno, že jsem přispěl Michalovi 500 Korun na jeho přání.

Krok 6

- **Akce** - Přihlásím se do svého testovacího PayPal účtu, který jsem použil pro platbu a zkontroluju stav účtu.
- **Výsledek** - V PayPal testovacím účtu se mi odečetlo 500 korun českých.

Krok 7

- **Akce** - Zkontroluji testovací účet Braintree.
- **Výsledek** - Na testovacím účtu Braintree, je zobrazena ve výpisu provedených transakcí transakce, kterou jsem provedl. Je zde záznam o tom, kdo posílal peníze, jaký obnos, na jaký účet a jakého typu platba byla. V tomhle případě bude typ platby PayPal.

Krok 8

- **Akce** - Přihlásím se do testovacího bussiness PayPal účtu aplikace ElateMe a zkontroluji stav účtu.
- **Výsledek** - Na účtu přibylo 500 Korun českých z mého testovacího účtu, ze kterého jsem provedl platbu.

Otestoval jsem tímto funkčnost platby pomocí metody PayPal od samotného počátku příspěvku uživatele až po připsání hotovosti na účet obchodníka. Testování probíhalo na testovacím prostředí společnosti Braintree za využití testovacích účtů PayPal. Produkční běh se od toho testovacího odlišuje pouze záměnou API klíče a použitím atributu 'production' místo 'sandbox', co se kódu týče. Předpokládáme tedy, že po přechodu do ostrého provozu nedojde k chybám v transakci a tato platební metoda bude plně funkční a použitelná bez nežádoucího chování.

Závěr

V analytické části jsem podrobně popsal všechny vybrané platební metody a u těch, u kterých to bylo potřeba, jsem popsal i správné využití platebního procesoru. Jak jsem předpokládal, tak nejsou všechny platební procesory vhodné pro všechny typy aplikací. Co se týče aplikace ElateMe, jenž je začínající aplikací, která nemá prozatím obrat jako velké společnosti, nemůže využívat platebních procesorů jako společnosti s vyšším obratem. Je to dáno tím, že různé platební procesory si za své služby účtují různé poplatky a pro menší společnosti se služby některých platebních procesorů jednoduše nevyplatí. Dále jsem se setkal s problémem, že některé platební procesory neumožňují přijímat platby v cizí měně a musí docházet ke konverzím, které mohou vést k finančním ztrátám, které mohou postihnout jak obchodníka tak zákazníka. Konkrétně jsme se s tímto problémem setkali u platební brány poskytované Fio bankou pro platbu kreditní kartou. Jelikož má mít aplikace ElateMe globální rozsah, znamenalo by to určitě problém při mimorepublikové transakci. Toto řešení je proto dočasné a do budoucna máme v plánu najít řešení, které nám bude více vyhovovat. Dále kromě platební metody Google Pay přibude nová platební metoda Apple Pay, jenž funguje na stejném principu jako Google Pay, a bude využívat také Payment Requestu. Apple Pay ale není v České republice zatím podporováno.

V další části této aplikace bylo popsáno, jak jsem postupoval při návrhu implementace jednotlivých platebních metod. Díky správnému výběru platebních procesorů jsem mohl použít velmi dobrou dokumentaci, jenž společnosti zprostředkovávající platební procesory dodávají. Popsal jsem, jaké technologie jsem při vývoji použil a jakým způsobem jsem platební metody integroval do aplikace ElateMe. Popsané postupy jsem doplnil o diagramy aktivit, které slouží k lepší představě o tom, jaké pohyby se při procesu jednotlivých platebních metod provádějí na pozadí aplikace.

Nakonec jsem vybral jednu platební metodu k otestování, kde jsem také popsal, z jakého důvodu nebyly testovány ty ostatní. Popsal jsem jednotlivé kroky v testovacím scénáři, jenž počínal zvolením přání, na které se bude

ZÁVĚR

přispívat až po připsání obnosu na účet ElateMe.

Do budoucna máme v plánu sledovat nejnovější trendy v platebních metodách a držet s těmito trendy krok, abychom poskytli uživateli co nejpohodlnější a nejrychlejší průběh platby.

Literatura

- [1] Kitamura, E.; Gash, D.; Koch, Z.: Introducing the Payment Request API. February 2018. Dostupné z: <https://developers.google.com/web/fundamentals/payments/>
- [2] Chargebacks911: Ultimate Guide: Internet Processing for Credit Card Payments. Dostupné z: <https://chargebacks911.com/knowledge-base/guidelines-for-internet-processing/>
- [3] BitcoinPay: Bitcoin Pay. 2018. Dostupné z: <https://www.bitcoinpay.com/cs/>
- [4] BitPay: Displaying Invoices. Dostupné z: <https://bitpay.com/docs/display-invoice>
- [5] BitcoinPay: Bitcoin Payment Gateway API. Dostupné z: <https://bitpay.com/bitcoin-payment-gateway-api>
- [6] Braintree: How it works. 2017. Dostupné z: <https://developers.braintreepayments.com/start/overview>
- [7] Brain, D.: PayPal Checkout. 2018. Dostupné z: <https://github.com/paypal/paypal-checkout/blob/master/docs/button.md>
- [8] Maněna, M.: Fio banka diagram aktivit. 2017.
- [9] Gaunt, M.: Deep dive into Payment Request API. 2018. Dostupné z: <https://developers.google.com/web/fundamentals/payments/deep-dive-into-payment-request>
- [10] platba, Q.: Tabulka 1 – základní parametry. 2018. Dostupné z: <https://qr-platba.cz/pro-vyvojare/specifikace-formatu/>
- [11] O'Shannessy, P.: qrcode.react. 2017. Dostupné z: <https://www.npmjs.com/package/qrcode.react>

- [12] Maněna, M.: O aplikaci Elateme. February 2017. Dostupné z: <https://elateme.com/about>
- [13] Wikipedia contributors: Near-field communication — Wikipedia, The Free Encyclopedia. 2018, [Online; accessed 4-May-2018]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Near-field_communication&oldid=837846348
- [14] Wikipedia contributors: Google Pay — Wikipedia, The Free Encyclopedia. 2018, [Online; accessed 4-May-2018]. Dostupné z: https://en.wikipedia.org/w/index.php?title=Google_Pay&oldid=836917370
- [15] Jones, C.: How To Accept Credit Card Payments Online in 2018: What Are Your Best Options? Dostupné z: <https://www.bigcommerce.com/blog/how-to-accept-credit-card-payments-online/>
- [16] Crowdfunding. 2017, accessed Apr. 18, 2018. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=Crowdfunding&oldid=15203914>
- [17] Maněna, M.: private communication, April 2018.
- [18] Česká spořitelna: Nejvyšší zabezpečení internetových plateb - 3D Secure. Dostupné z: https://www.csas.cz/static_internet/cs/html/jakplatitkartou/internet.html#3d
- [19] Rouse, M.: Peer-to-peer. August 2014. Dostupné z: <http://searchnetworking.techtarget.com/definition/peer-to-peer>
- [20] BitPay: Bit Pay. Dostupné z: <https://www.bitpay.com>
- [21] Coufalová, D.: Jak dlouho trvá převod z banky do banky. July 2017. Dostupné z: <https://www.usetreno.cz/jak-dlouho-trva-prevod-penez-z-banky-do-banky/>
- [22] PayPal. 2018, [Online; accessed 20-April-2018]. Dostupné z: <https://en.wikipedia.org/w/index.php?title=PayPal&oldid=836705704>
- [23] Wikipedie: PayPal — Wikipedie: Otevřená encyklopedie. 2018, [Online; navštíveno 20. 04. 2018]. Dostupné z: <https://cs.wikipedia.org/w/index.php?title=PayPal&oldid=15719870>
- [24] contributors, W.: Braintree (company). 2018, [Online; accessed 20-April-2018]. Dostupné z: [https://en.wikipedia.org/w/index.php?title=Braintree_\(company\)&oldid=834284716](https://en.wikipedia.org/w/index.php?title=Braintree_(company)&oldid=834284716)
- [25] BigCommerce: Connecting with PayPal powered by Braintree. Dostupné z: <https://support.bigcommerce.com/articles/Public/Connecting-with-PayPal-Powered-by-Braintree#incontext>

-
- [26] Braintree: What you'll need. 2017. Dostupné z: <https://articles.braintreepayments.com/get-started/overview>
- [27] Braintree: Key concepts. 2017. Dostupné z: <https://developers.braintreepayments.com/start/overview>
- [28] Miksu, V.: React - Úvod. Dostupné z: <https://www.dzejes.cz/react-uvod.html>
- [29] Bittner, J.: Úvod do CSS preprocesoru Sass. Dostupné z: <https://www.itnetwork.cz/html-css/webove-portfolio/tutorial-moderni-webove-portfolio-sass>
- [30] BitPay: Creating invoices. Dostupné z: <https://bitpay.com/docs/create-invoice>
- [31] Braintree: Checkout with PayPal. Dostupné z: <https://developers.braintreepayments.com/guides/paypal/checkout-with-paypal/javascript/v3>
- [32] Braintree: Payment Request API - Setup and Integration. Dostupné z: <https://developers.braintreepayments.com/guides/payment-request/setup-and-integration/javascript/v3>
- [33] Braintree: Testing and Go live PayPal method. Dostupné z: <https://developers.braintreepayments.com/guides/paypal/testing-go-live/node#linked-paypal-testing>

Seznam použitých zkratk

API Application Programming Interface

npm node package manager

CSS cascade style sheets

Sass Syntacticaly awesome style sheets

HTTPS Hyper Text Transfer Protocol Secure

HTTP Hyper Text Transfer Protocol

SSL Secure Socker Layer

TLS Transport Layer Security

P2P Peer to Peer

Návod na vyzkoušení platebních metod v aplikaci ElateMe

Cílem této přílohy je popsat, jakým způsobem by měl čtenář této práce postupovat, aby si vyzkoušel implementované platby v chodu. Jelikož jsem se podílel pouze na části vývoje aplikace ElateMe, nemohu zveřejnit veškeré zdrojové kódy k aplikaci na přiloženém CD, a proto jsou platební metody v testovacím verzi k dispozici na stránce www.elateme.com.

Je zapotřebí, aby čtenář vlastnil profil na sociální síti Facebook, je to totiž jediná možnost, jak se do aplikace přihlásit. Dále je zapotřebí, aby si čtenář založil na sociální síti Facebook testovací profil. Čtenář si na svém reálném profilu musí přidat testovací profil do přátel, aby se mu následně v aplikaci ElateMe zobrazila přání založená pod testovacím profilem a mohl na ně přispívat. Čtenář se přihlásí do aplikace na stránce www.elateme.com pod svým testovacím profilem. Úvodní stránka aplikace, kde se zobrazují přání přátel, bude prozatím prázdná. V horní liště aplikace ElateMe klikne na tlačítko “Vytvořit přání“. Do vyhledávání napíše název přání, na které chce zahájit přispívání, například macbook air. Zobrazí se nabídka produktů s daným názvem. Čtenář zvolí jeden produkt ze seznamu nabízených a v pravém horním rohu detailu klikne na tlačítko “Potvrdit“. Tímto se vytvořilo na testovacím profilu nové přání, které se bude zobrazovat veškerým přátelům testovacího profilu, kteří budou využívat aplikaci ElateMe. Dále je zapotřebí se z testovacího profilu ElateMe odhlásit, v pravém horním rohu, a přihlásit se pod reálným profilem. Po přihlášení bude na úvodní stránce aplikace ElateMe zobrazeno přání, které vytvořil čtenář pomocí testovacího profilu. Čtenář klikne na nabízené přání a tím se dostane na detail přání, kde už může provést příspěvek, a vyzkoušet si implementované platební metody.

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe	adresář se spustitelnou formou implementace
	src	
	impl.....	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu L ^A T _E X
	text	text práce
	thesis.pdf	text práce ve formátu PDF
	thesis.ps	text práce ve formátu PS