



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	ElateMe - iOS klient I
Student:	Yegor Terokhin
Vedoucí:	Ing. Ji í Chludil
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

System ElateMe je webová aplikace s mobilními klienty poskytující funkcionalitu pro crowdfunding a pro pln ní p ání uživatel a jejich p átel (nap . Vánoce, narozeniny, atp.). ElateMe je týmový projekt. Analyzujte a implementujte klienta služby ElateMe na platform iOS.

1. Analyzujte:
 - diagram p ípadu užití v etn jejich diagram aktivity,
 - bitcoin platby,
 - diagram popisující funk nost aplikace jako celku.
2. Navrh te:
 - serverovou ást pro bitcoin platby,
 - jednotkové testy serverové ásti,
 - d ležitá sekven ní diagramy.
3. Implementujte:
 - zpracování a komentování p ání,
 - sledování novinek a jejich správu,
 - správu p ání,
 - grafické uživatelské rozhraní dle dodaného návrhu.
4. Vytvo te jednotkové testy dle návrhu z bodu 2.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 14. února 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

ElateMe - iOS klient I

Yegor Terokhin

Vedoucí práce: Ing. Jiří Chludil

15. května 2017

Poděkování

Já bych chtěl poděkovat svému vedoucímu práce, panu Jiřímu Chludilovi, za jeho pomoc v napsání této práci. Dál bych chtěl poděkovat Michalovi Maněnovi za řízení projektu a cenné rady.

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 15. května 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Yegor Terokhin. 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, je nezbytný souhlas autora.

Odkaz na tuto práci

Terokhin, Yegor. *ElateMe - iOS klient I*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Tato práce se zabývá vývojem mobilní iOS aplikace pro crowdfundingovou platformu s elementy sociální sítě ElateMe. Daný servis umožňuje uživatelům sbírat prostředky na vlastní přání a taky přispívat na přání svých kamarádů. Nejvíc užitečným to je v dobu svátků, například narozenin. Protože každý pozvaný na oslavu člověk chce zjistit co by opravdu chtěl dostat druhý za dárek. Právě v ElateMe tuto informace najde a navíc se může rovnou přidat k ostatním darujícím a pomoci kamarádovi ve splnění jeho přání. Celá práce nad projektem ElateMe se představuje kolektivním výsledkem spolupráci různých lidí, kteří dělali vývoj backend API, reklamního serveru, mobilní Android a iOS aplikací.

Součástí této práce je popis historie a problematiky projektu ElateMe, podrobná analýza a návrh funkčních a nefunkčních požadavků, případů užití, včetně jednotných diagram aktivit, kryptoměny Bitcoin a její způsob použití v rámci ElateMe a samotná implementace iOS aplikace. Nakonec aplikace je podrobena jednotkovým testům.

Klíčová slova ElateMe, crowdfundingová platforma, sociální síť, iOS, iPhone, Bitcoin platba, mobilní aplikace, dárek, přání

Abstract

The main aim of this thesis is the development of an iOS mobile application for the crowdfunding platform with ElateMe social network elements. This service allows users to gather finances for personal wishes and fund their friends' wishes. It will be very useful during holidays, like birthdays. Because every invited person would like to know exactly what gift someone wants. ElateMe becomes the perfect source for such information making friend's wish real by helping to donate to it. The whole ElateMe project is a collective work of different people, who were developing backend API, an advertisement server, Android, and iOS mobile applications.

This thesis contains history description, problems of the ElateMe project, detailed analysis, and design of functional and non-functional requirements, use cases, including activity diagrams, Bitcoin cryptocurrency and its role in ElateMe, and iOS implementation. In the end, the application was tested using unit tests.

Keywords ElateMe, crowdfunding platform, social network, iOS, iPhone, Bitcoin payment, mobile application, gift, wish

Obsah

Úvod	1
ElateMe	1
Hlavní cíl	1
Motivace	2
1 Analýza	3
1.1 Popis projektu jako funkčního celku	3
1.2 Funkční požadavky	8
1.3 Nefunkční požadavky	10
1.4 Případy užití	11
1.5 Donace	27
2 Návrh	31
2.1 Práce s digitální měnou Bitcoin	32
2.2 Testování síťové vrstvy	34
2.3 Návrh Bitcoin platby	35
2.4 Návrh komentování přání	37
2.5 Návrh notifikace	40
3 Implementace	45
3.1 Použité nástroje	45
3.2 Implementace zpracování a komentování přání	46
3.3 Správa novinek	49
3.4 Správa přání	50
4 Testování	53
4.1 Unit testování iOS aplikace	53
4.2 Testování Bitcoin platby	54
Závěr	55

Výsledek	55
Možnosti dalšího vývoje	55
Literatura	57
A Seznam použitých zkratk	59
B Obsah přiloženého CD	61
C Instalační příručka	63
C.1 Instalace z CD	63

Seznam obrázků

1.1	BPMN diagram popisující funkčnost aplikace jako celku	8
1.2	Use-case diagram, shrnuje všechny případy užití	12
1.3	Diagram aktivit, popisující přihlášení pomocí Facebook účtu . . .	14
1.4	Diagram aktivit, popisující odhlášení	15
1.5	Diagram aktivit, popisující přidání přání	17
1.6	Diagram aktivit, popisující splnění přání uživatele	19
1.7	Diagram aktivit, popisující zobrazení detailu přání	20
1.8	Diagram aktivit, popisující zobrazení seznamu komentářů ke kon- krétnímu přání	21
1.9	Diagram aktivit, popisující napsání komentáře ke konkrétnímu přání	22
1.10	Diagram aktivit, popisující zobrazení seznamu kamarádů uživatele	23
1.11	Diagram aktivit, popisující zobrazení seznamu přání kamarádů . .	24
1.12	Diagram aktivit, popisující nastavení bankovního účtu uživatele . .	25
1.13	Diagram aktivit, popisující donace	26
1.14	Ukázka fungování bitcoinů	29
2.1	Sekvenční diagram, popisující Bitcoin platbu	36
2.2	Sekvenční diagram, popisující zobrazení komentářů	39
2.3	Sekvenční diagram, popisující zobrazení seznamu notifikace	42
3.1	Grafický návrh vzhledu detailu přání s komentáři	47
3.2	Horní část detailu přání	48
3.3	Spodní část detailu přání s komentáři	48
3.4	Grafický návrh seznamu notifikací	50
3.5	Seznám notifikací	50
3.6	Grafický návrh správy přání	51
3.7	Správa přání	51

Seznam tabulek

Úvod

Za posledních 30 let se lidstvo velice rychle posunulo v technologickém oboru. Na začátku to bylo vidět nejvíce ve výrobním průmyslu, ale v poslední době se mnoho změnilo i v běžném lidském životě. Dnes si opravdu těžko představíme naše bytí bez internetu, chytrých telefonů a počítače. Ačkoliv jejich prvotním účelem bylo jen usnadnit život člověka při práci, staly se nedílnou součástí každodenního života každého z nás. Proč? Protože chytrá zařízení dokázala vyřešit problémy, kterých si spousta z nás vůbec nevšimla, a ti co ano, je v tu dobu nedokázali vyřešit, vyřešili je chybně, nebo je nebrali za vážné. Teď je však realita jiná. Pomocí svého telefonu, počítače, tabletu nebo jiného chytrého zařízení si můžete objednat pizzu, zavolat kamarádovi z jiné části planety, poslat fotografii babičce, domluvit si schůzku s vedoucím a usnadnit si i spoustu dalších běžných aktivit. Nejlepší je, že dokážete všechno z toho udělat během pár vteřin kdykoliv a odkudkoliv. Není však tajemstvím, že existuje ještě spousta dalších problémů, které zatím nikdo nevyřešil. Jedním z takových problémů je darování dárců.

ElateMe

Systém ElateMe je projektem, který reprezentuje crowdfunding platformu s elementy sociální sítě. Za svůj hlavní cíl bere vyřešit problém darování dárku svým blízkým a kamarádům na různé svátky (např. Vánoce, narozeniny, atp). Běžný uživatel služby ElateMe může sdílet své přání s dalšími uživateli a rovněž finančně přispívat ostatním uživatelům na jejich přání.

Hlavní cíl

Cílem této práce je vytvořit mobilního klienta služby ElateMe pro zařízení s operačním systémem iOS. Vývoj se bude skládat z analýzy a návrhu případů

užití, bitcoin platby a funkčnosti aplikace jako celku. Pak bude následovat samotná implementace aplikace a nakonec unit testování.

Motivace

Hlavní motivace je použít znalostí iOS programování v rámci vývoje velkého projektu, který má vyřešit podstatný problém pro běžného člověka. Navíc se mám velkou příležitost naučit pracovat v týmu, kde každý člen má právo na svůj názor a všichni to uvítají. To znamená, že po práci nad projektem ElateMe získám velmi důležité dovednosti nejen jako iOS vývojář, ale hlavně jako softwarový inženýr, který umí analyzovat, navrhovat a realizovat softwarový produkt a taky komunikovat s jinými členy týmu pro dosažení větších výsledků a nejlepší kvality.

Analýza

Následující kapitola se zabývá analýzou projektu před návrhem. Popíše zde hlavní cíle této analýzy.

- Historie projektu.
- Popis projektu jako funkčního celku
- Definovat funkční požadavky.
- Definovat nefunkční požadavky.
- Definovat případy užití aplikace.
- Seznámit čtenáře s Bitcoin platbou, kterou budu používat ve své práci.

Na konci analýzy budu mít všechny informace potřebné pro kompletní návrh iOS aplikace ElateMe.

1.1 Popis projektu jako funkčního celku

V této sekci popíšu problematiku a historie vzniku projektu ElateMe. Na konci uvedu diagram, který bude popisovat projekt jako funkční celek.

1.1.1 Problematika

Tady popíšu problémy, se kterými se setkávají lidé, když chtějí udělat někomu dárek. A taky problémy lidí, které čekají dárek na nějaký svátek.

1.1.1.1 Problém, který řeším

Určitě každý z nás měl narozeniny a dostával na nich dárky od svých rodičů a přátel. Každý z nás také vystupoval v roli darujícího. Rozeberme problémy a překážky, se kterými se setkáváte v těchto situacích.

1.1.1.2 Problémy, když darujete dárek

- Nevíte přesně, co by cílová osoba chtěla opravdu dostat.
- Nevíte jakou částkou přispět na dárek, abyste obdarovaného neurazili.
- Nevíte, jestli jim váš dárek k užitku.
- Nevíte, jak často bude váš dárek používán.
- Nechcete svým dárkem zhoršit své vztahy s obdarovaným.
- Nevíte, zda váš dárek bude unikátní, tj. nikdo další obdarovanému nedá stejný dárek jako vy.
- V případě, když nemáte postačující peněžní částku na vhodný dárek, snažíte se sehnat někoho, kdo by byl ochotný se k vám přidat a vymyslet spolu s vámi dobrý dárek. V nejhorším případě se můžete dostat do situace, kdy nikoho takového neseženete a/nebo se dostanete do sociálního nepohodlí, protože nejste tak bohatí jako ostatní.

1.1.1.3 Problémy, když očekáváte dárek

- Nevíte, co opravdu chcete dostat.
- Nechcete dostat něco, co nemáte rádi.
- Nechcete se přetvařovat, když dostanete něco, co nepotřebujete, ale zároveň nechcete naštvat darujícího.
- Nevíte, jak říct, co byste chtěli dostat, tak, aby lidé vás lidé nepovažovali za ziskuchtivého člověka.

1.1.2 Hlavní motivace projektu

Vyřešení dlouhodobého problému lidstva je největší motivací projektu ElateMe. Hlavní motivační slogan, který držíme v hlavě v průběhu práce nad ElateMe, zní takto:

„Tvoříme projekt, který vyřeší problémy lidstva s darováním dáreků. Nyní nikdo neztratí spoustu svého volného času a nervů přemýšlením o nejlepším dárku blízkému člověku. Nyní se nikdo nebude cítit jako pokrytec, když dostane věc, kterou nepotřebuje. Naše aplikace promění sen na realitu mnoha lidem. Změníme svět k lepšímu!“

A to není hloupá fantazie. Když vyprávím lidem o své bakalářské práci a projektu, dozvím se od nich, že to opravdu potřebují a chtějí, což mě motivuje ještě více.

1.1.3 Historie projektu

Historie projektu ElateMe začíná v únoru roku 2016, kdy 6 studentů, včetně mě, hledalo zajímavý projekt pro předmět BI-SP1 (Softwarový projekt I) a narazilo na inženýra Jiřího Chludila (dále jen vedoucí), který měl skvělý projekt od zadavatele Michala Maněny (dále jen zadavatel). V té době se náš tým skládal z teamleadera a backend specialisty Maksyma Balatska, backend a Android programátora Yevhena Kuzmovyche, architekta a Android programátora Georgia Solovyova, Android specialisty Borise Laskova, iOS programátora Gleba Arkhipova a mě, iOS specialisty Yegora Terokhina.

1.1.3.1 První setkání

Na úvodní schůzce vedoucí povídal o základech týmové práce na projektu a nejčastějších chybách, se kterými se setkávají lidé při týmovém vývoji narozdíl od samotného programování. O něco později se k nám přidal i zadavatel. Zeptal se nás na naši motivaci v předmětu BI-SP1 a hlavní výhody a nevýhody našeho týmu z našeho pohledu. Po naší odpovědi vypadal nadšeně, stejně jako my. O pár minut později nám pověděl, co nás čeká v projektu a co je základní myšlenkou projektu, sdělil nám svou vizi a cíl, kterého by projektem chtěl dosáhnout. Velmi se nám to líbilo a měli jsme chuť se do toho pustit a být součástí projektu, který později dostal název ElateMe.

1.1.3.2 Softwarový projekt I

Práci nad svým prvním velkým projektem náš tým začal ještě před začátkem semestru ve zkouškovém období. Začali jsme analýzou a programováním jednodušších součástí aplikace, např. autorizací pomocí Facebook účtu, dále pak nastavením různých programů, které jsme používali, např. IDE a GIT repositář. V té době bylo totiž jediným, co jsme měli, jen základní myšlenka aplikace, která zněla takto: „*Chceme udělat mobilní aplikaci, která usnadní lidem darování a dostávání dáreků*“. Byla to dobrá myšlenka, která však zatím neobsahovala nic konkrétního k implementaci. Tady jsme začali zlepšovat naše dovednosti v oblasti softwarové analýzy. Postupovali jsme od začátku – vytvořili jsme doménový model a ukázali jsme jej zadavateli. To mu dalo jasný přehled toho, jak jsme pochopili zadání projektu a možnost diskutovat možná vylepšení a problémové části, na které si máme dát pozor.

Půl semestru jsme se zabývali analýzou, sestavením různých diagramů, např. doménového modelu, class-diagramu, use-case diagramů, stavového diagramu. Inspirovali jsme se ukázkami, které nám poskytl zadavatel, a jeho poznámkami. Tak jsme se naučili převádět diskuze se zadavatelem do návrhu a pak i do následné implementace. Bylo to pro nás velmi zajímavé a užitečné, proto jsme se na každé schůzce snažili prodiskutovat co nejvíce věcí a odnášet si z ní více odpovědí než otázek.

Na konci semestru jsme měli funkční prototyp iOS a Android aplikace. Za celý semestr jsme dokázali implementovat následující případy užití:

- Přihlašování do aplikace pomocí Facebook účtu.
- Odhlašování z aplikace.
- Vytváření přání.
- Zobrazení přátel z Facebooku, kteří používají aplikaci ElateMe.
- Zobrazení přání uživatele.
- Zobrazení přání všech přátel.
- Zobrazení komentářů k samotnému přání.
- Psaní komentářů k samotnému přání.

Celkem se nám podařilo udělat hodně práce, včetně analýzy, návrhu univerzální architektury pro platformy Android a iOS a implementace funkčního prototypu. Co nám velmi chybělo, byl grafický návrh. Nikdo z nás neměl dostatečné dovednosti v návrhu kvalitního UI designu, který by bral v úvahu nejlepší UX uživatele s mobilními aplikacemi, zároveň dodržoval iOS Human Interface Guidelines [1] pro iOS a User Interface Guidelines pro Android a přitom docílil toho, že by mezi designem aplikací pro tyto dvě platformy nebyl podstatný rozdíl, aby nedošlo ke zmatení běžného uživatele, který by používal ElateMe na zařízeních s různým operačním systémem. Naštěstí byl tento problém vyřešen v následujícím semestru. Mohu natvrdo říct, že jsme udělali dobrý dojem na vedoucího a zadavatele, protože jsme byli aktivní a motivovaní k dlouhodobé a kvalitní spolupráci. Za sebe osobně můžu říct, že jsem velice chtěl udělat projekt od začátku do konce, tj. od analýzy do dodání projektu zadavateli.

1.1.3.3 Softwarový projekt II

V dalším semestru jsme pokračovali v práci na projektu ElateMe v rámci předmětu BI-SP2 (Softwarový projekt II). Zadavatel pro nás připravil grafické podklady a vymyslel spoustu dalších vylepšení a nových funkcí. Tady jsme narozdíl od předcházejícího semestru zlepšili dovednosti v oblasti projektového řízení a návrhu řešení, které dělá v práci běžný softwarový inženýr, například návrhu vhodné architektury a flexibilního řešení pro různé případy užití. Od té doby se projekt ElateMe přetvořil na startup z hlediska byznys modelu, protože požadavky se začaly rychle měnit a rozšiřovat oproti původnímu modelu. Museli jsme s tím počítat při návrhu.

Dále uvedu základní změny v projektu oproti tomu, co jsme dělali v rámci projektu BI-SP1.

- Probrali jsme grafické podklady, opravili a vylepšili různé věci s ohledem na platformu a mobilní UX.
- Probrali jsme možné byznys modely, které by se daly využít v ElateMe.
- Rozdělili jsme jeden velký grafický návrh na dva samostatné, zvlášť pro platformu iOS a zvlášť pro Android.
- Přidali jsme hezkou úvodní animaci při spuštění aplikace, což uvítá běžný uživatel, protože tímto způsobem děláme aplikaci živější.
- Sestavili jsme implementační plán s ohledem na změny.

Dopředu musím konstatovat, že se náš tým nedokázal držet implementačního plánu na 100%, protože uprostřed semestru jsme narazili na věci, které bylo potřeba upravit, abychom usnadnili implementaci a zlepšili celkovou kvalitu projektu. Myslím, že to není žádný problém, protože vývoj softwarového produktu je těžký proces, který skoro nikdy nemá pevně definované hranice a je mnohem lepší si všimnout případných chyb a nedostatků a zanalyzovat je uprostřed vývoje než na konci, protože je ještě nějaká šance to opravit relativně rychle s ohledem na potenciální výnos po opravě.

Při vývoji jsme také počítali s tím, že aplikace se v budoucnu bude neustále rozšiřovat a má tedy smysl psát kód tak, aby se další programátor, který se přidá k implementaci projektu, dokázal rychle zorientovat v projektu a být nápomocným co nejdříve. Proto bylo důležité sestavit sadu pravidel pro psaní kódu, pomocí kterých by bylo snadné udržovat samotný projekt konzistentní. Vybrali jsme rovněž moderní řešení každodenních problémů iOS programátora, které popíšu detailněji v kapitole Návrh.

Na konci semestru jsme měli hotové funkční alfa verze aplikací pro iOS a Android, které odpovídaly grafickým návrhům.

1.1.4 BPMN diagram popisující funkčnost aplikace jako celku

V této sekci ukážu a popíšu BPMN (Business Process Model and Notation) [2] diagram, který bude popisovat aplikaci jako funkční celek. BPMN byl zvolen z důvodu dobré srozumitelnosti z hlediska business logiky jak pro programátory, tak i pro lidi mimo obor IT. Na diagramu níže je popsána hlavní business logika platformy ElateMe – proces vytváření přání uživatelem a jeho evidence systémem. Potom následuje popis procesu přispívání na toto přání přáteli uživatele. V diagramu záměrně chybí konkrétní názvy entit, protože jejich změna by neměnila základní business logiku. Například entita "Online obchody" nemá žádný název, protože změna obchodu, který se bude používat v ElateMe, neovlivní proces vytváření přání.

F3: Udržování přihlášení Přihlášený uživatel bude po zavření a opětovném otevření aplikace opět automaticky přihlášen.

1.2.0.2 Přání

F4: Přidání přání Uživatel bude moci vytvořit své vlastní přání.

F5: Splnění přání Ke splnění přání může dojít v případě, když bude nasbírána požadovaná částka peněz.

F6: Nesplnění přání V případě, že dojde k situaci, kdy přání nebude splněno a nastane poslední den jeho platnosti, přání přejde do stavu nesplněného přání.

F7: Smazání přání Uživatel bude moci smazat své přání.

F8: Našeptávání přání Aplikace bude schopna ukázat seznam doporučených přání podle prvních písmen, která napsal uživatel při vyplňování názvu nového přání.

F9: Zobrazení přání uživatele Aplikace bude schopna zobrazit všechna přání uživatele.

F10: Zobrazení přání kamarádů Aplikace bude schopna zobrazit všechna přání kamarádů uživatele.

F11: Zobrazení detailu přání Aplikace bude schopna zobrazit detail přání.

1.2.0.3 Donace na přání

F12: Donace na přání Uživatel bude moci finančně přispět na přání kamaráda.

F13: Donace uživatele Aplikace bude schopna zobrazit všechny donace, které uživatel udělal v rámci ElateMe.

F14: Zobrazení donací na přání Aplikace bude schopna zobrazit všechny donace na konkrétní přání.

1.2.0.4 Kamarády

F15: Zobrazení kamarádů Aplikace bude schopna zobrazit kamarády z Facebooku, kteří mají ElateMe účet.

1.2.0.5 Komentáře

F16: Komentování přání Uživatel bude moci okomentovat přání kamarádů.

F17: Smazání komentáře Uživatel bude moci smazat komentář ke přání.

F18: Zobrazení komentáře k přání Aplikace bude schopna zobrazit komentáře k přání.

1.2.0.6 Nastavení

F19: Zobrazení uživatelských údajů Aplikace se bude moci zobrazit údaje uživatele, jako jméno, příjmení a foto v nastaveních.

F20: Přidání číslo bankovního účtu Uživatel se bude moci přidat platební kartu v nastaveních aplikace.

1.3 Nefunkční požadavky

V této sekci popíšu technické požadavky na aplikaci, které nebudou tvořit její funkcionalitu.

N1: Nativní iOS aplikace Výsledným produktem bude nativní aplikace pro operační systém iOS.

N2: Verze iOS Systém bude možné spustit na zařízeních s operačním systémem iOS verze 9.0 a vyšší.

N3: Standardní uživatelské rozhraní Uživatelské rozhraní systému bude v souladu s Apple guidelines.

N4: Uživatelská přívětivost Systém bude intuitivní pro uživatele a nebude potřebovat návod k použití.

N5: Odolnost proti nestandardním situacím Systém bude spolehlivě ošetřovat případné chyby serveru a nedostane se kvůli nim do nekonzistentního stavu.

N6: Jazyky rozhraní Systém bude s uživatelem komunikovat v českém jazyce s možností snadné rozšiřitelnosti na ostatní jazyky.

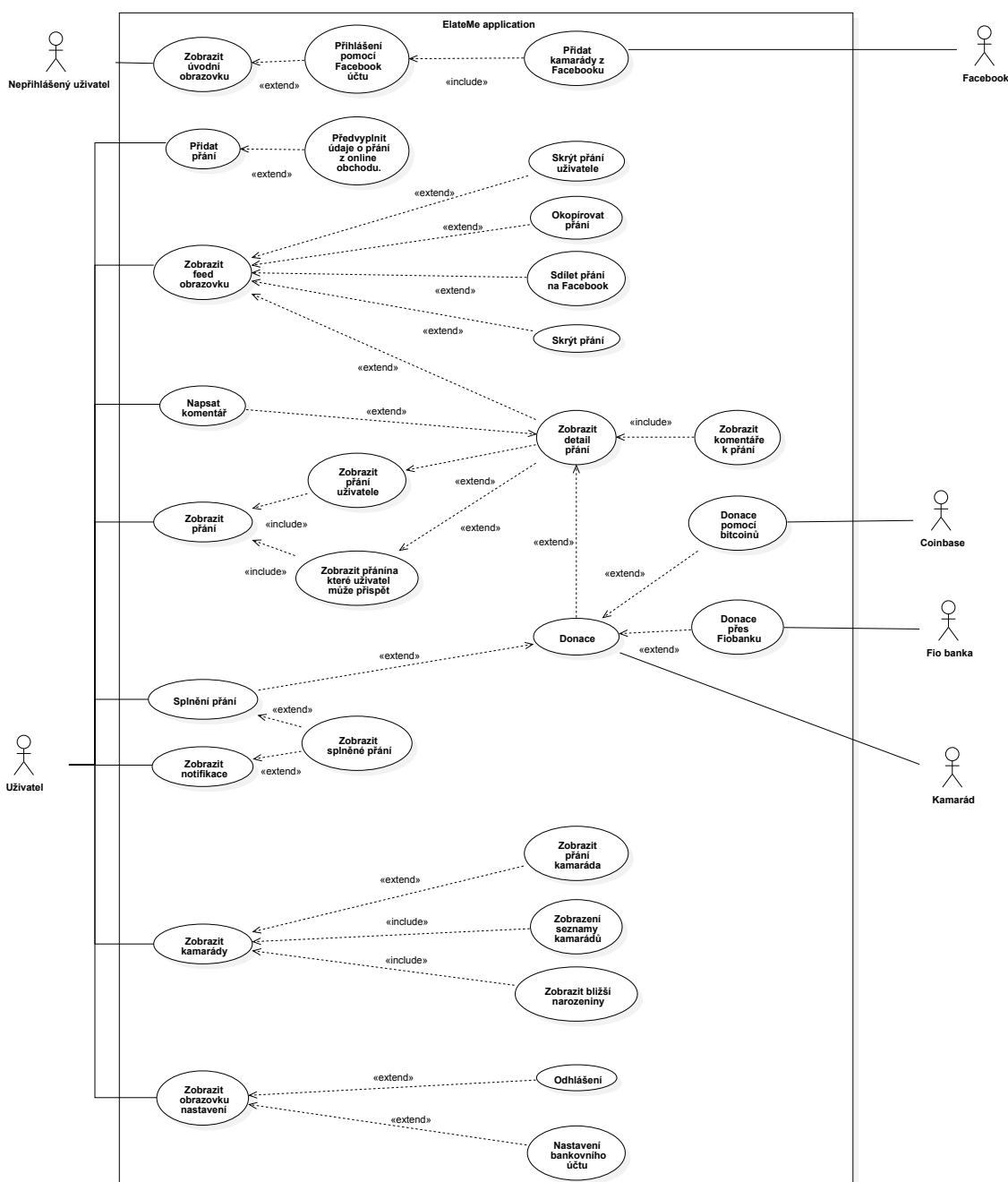
N7: Rozšiřitelnost Architektura bude navržena tak, aby aplikace byla snadno rozšiřitelná. Všechny statické zdroje budou dekomponované a malá změna uživatelského rozhraní nebude vyžadovat změny bussiness logiky.

1.4 Případy užití

Po analýze funkčních a nefunkčních požadavků jsem sepsal případy užití, tzv. *use cases*. Jejich hlavním cílem je detailně popsat všechny typické scénáře používání iOS aplikace ElateMe. Každý případ obsahuje popis, počáteční a konečné podmínky a diagram aktivit. Kvůli tomu, že vytvářím mobilního klienta, se kterým bude bezprostředně pracovat uživatel ElateMe, všechny případy užití budu prozkoumávat přímo z hlediska tohoto uživatele, tzn. nebudu popisovat, jak fungují různé části systému, například na serveru, ale zaměřím se přímo na průběh používání iOS aplikace ElateMe koncovým uživatelem.

Následující obrázek shrnuje všechny případy užití.

1. ANALÝZA



Obrázek 1.2: Use-case diagram, shrnuje všechny případy užití

1.4.1 UC1: Přihlášení pomocí Facebook účtu

1.4.1.1 Popis

Přihlášení přes Facebook účet. Bude využívat Facebook SDK [3].

1.4.1.2 Počáteční podmínky

Uživatel není přihlášen do aplikace.

1.4.1.3 Koncové podmínky

Uživatel je přihlášen do aplikace, pokud nenastala žádná chyba.

1.4.1.4 Aktér

Nepřihlášený uživatel.

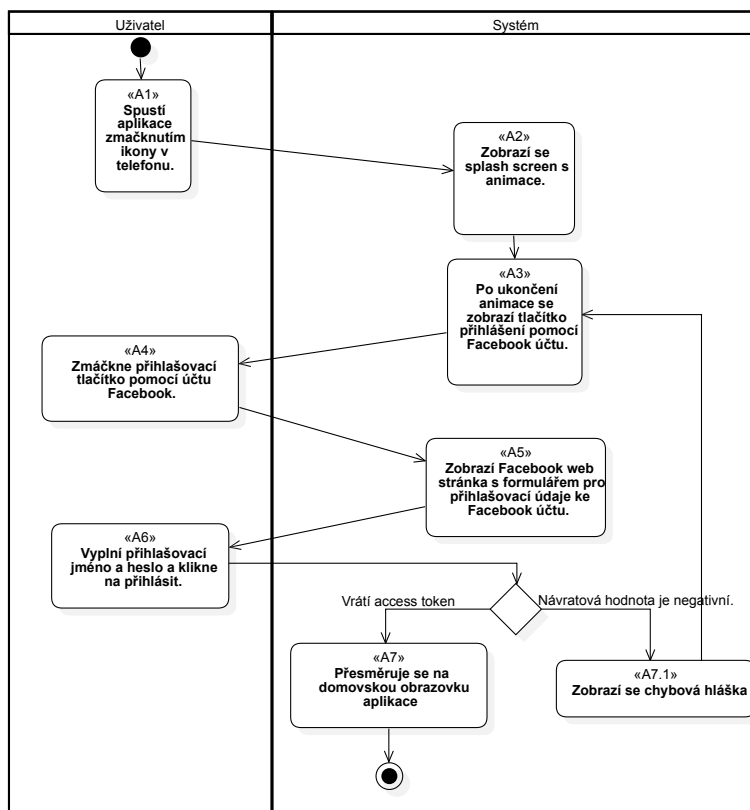
1.4.1.5 Popis aktivit

#	Uživatel	Systém
A1	Spustí aplikaci stisknutím ikony v telefonu.	
A2		Zobrazí splash screen s animací.
A3		Po ukončení animace zobrazí tlačítko přihlášení pomocí Facebook účtu.
A4	Stiskne tlačítko pro přihlášení pomocí účtu služby Facebook.	
A5		Zobrazí webový formulář pro zadání přihlašovacích údajů k Facebook účtu.
A6	Vyplní přihlašovací jméno a heslo a klikne na tlačítko přihlásit.	
A7		Vrátí access token a přesměruje uživatele na domovskou obrazovku aplikace.

1.4.1.6 Alternativní toky

#	Uživatel	Systém
A7.1		Návratová hodnota z Facebook serveru je negativní. Zobrazí se chybová hláška. Dále pokračuje do A2.

1. ANALÝZA



Obrázek 1.3: Diagram aktivit, popisující přihlášení pomocí Facebook účtu

1.4.2 UC2: Odhlášení

1.4.2.1 Popis

Odhlášení z aplikace.

1.4.2.2 Počáteční podmínky

Uživatel je přihlášen do aplikace.

1.4.2.3 Koncové podmínky

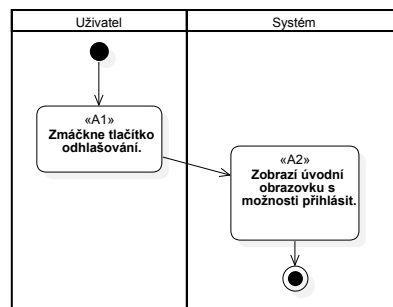
Uživatel není přihlášen do aplikace.

1.4.2.4 Aktér

Přihlášený uživatel.

1.4.2.5 Popis aktivit

#	Uživatel	Systém
A1	Stiskne tlačítko pro odhlášení.	
A2		Zobrazí úvodní obrazovku s možností přihlásit se.



Obrázek 1.4: Diagram aktivit, popisující odhlášení

1.4.3 UC3: Přidání přání

1.4.3.1 Popis

Uživatel má možnost vytvořit vlastní přání.

1.4.3.2 Počáteční podmínky

1.4.3.3 Koncové podmínky

Vytvořené přání, které bude viditelné pro vybrané kamarády uživatele.

1.4.3.4 Aktér

Přihlášený uživatel.

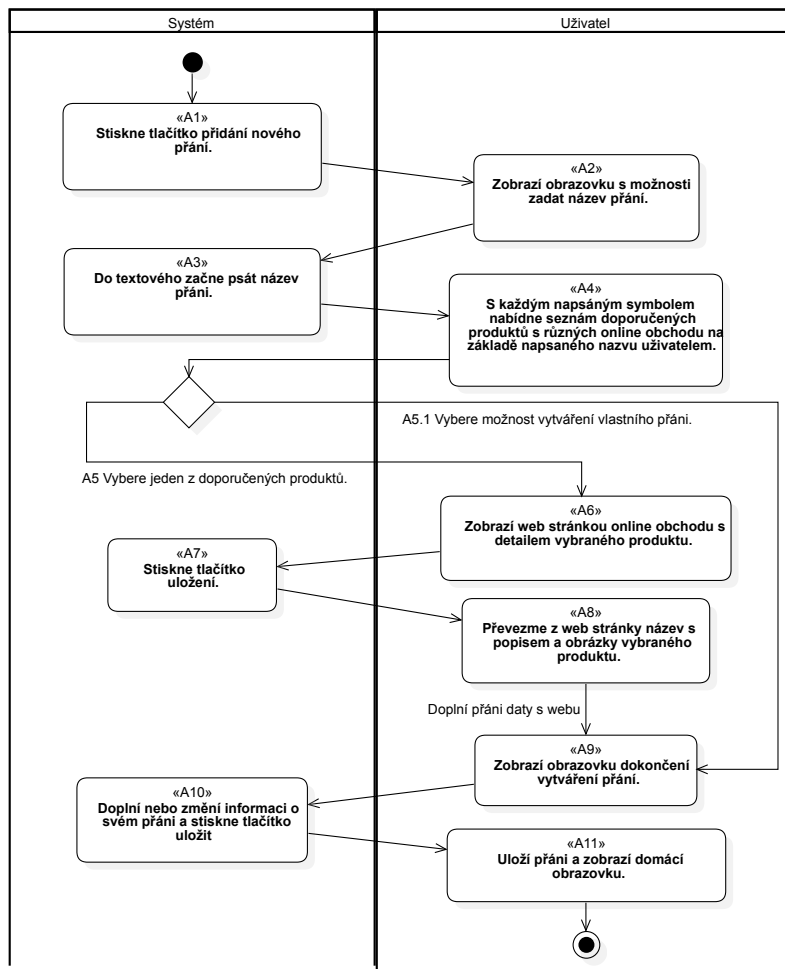
1. ANALÝZA

1.4.3.5 Popis aktivit

#	Uživatel	Systém
A1	Stiskne tlačítko pro přidání nového přání.	
A2		Zobrazí obrazovku s možností zadat název přání.
A3	Začne psát název přání do textového pole.	
A4		S každým napsaným symbolem nabídne seznam doporučených produktů z různých online obchodů na základě názvu zadaného uživatelem.
A5	Vybere jeden z doporučených produktů.	
A6		Zobrazí web stránku online obchodu s detailem vybraného produktu.
A7	Stiskne tlačítko uložení.	
A8		Převezme z web stránky název s popisem a obrázky vybraného produktu.
A8		Zobrazí obrazovku dokončení vytváření přání s předvyplněnými informacemi produktu s online obchodu.
A9	Doplní nebo změní informace o svém přání a stiskne tlačítko po uložení.	
A10		Uloží přání a zobrazí domovskou obrazovku.

1.4.3.6 Alternativní toky

#	Uživatel	Systém
A5.1	Vybere možnost vytvoření vlastního přání.	
A6.1		Zobrazí obrazovku pro dokončení vytváření přání s předvyplněným názvem.



Obrázek 1.5: Diagram aktivit, popisující přidání přání

1.4.4 UC4: Splnění přání uživatele

1.4.4.1 Popis

Systém upozorní uživatele, že částka peněz, kterou potřeboval na splnění přání, je nasbíraná a došlo ke splnění tohoto přání.

1.4.4.2 Počáteční podmínky

Uživatel přihlášený do aplikace, který má vytvořené přání.

1.4.4.3 Koncové podmínky

Uživatel přihlášený do aplikace, který má splněné přání.

1. ANALÝZA

1.4.4.4 Aktér

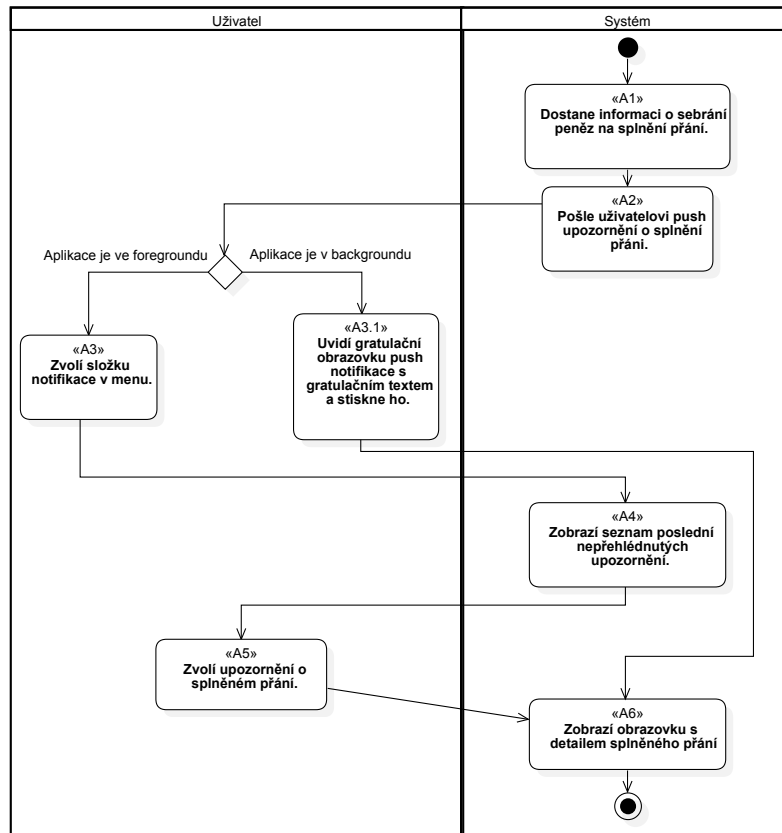
Přihlášený uživatel.

1.4.4.5 Popis aktivit

#	Uživatel	Systém
A1		Dostane informaci o nasbírání peněz na splnění přání.
A2		Pošle uživateli push upozornění o splnění přání.
A3	Zvolí položku notifikace v menu.	
A4		Zobrazí seznam posledních nezobrazených upozornění.
A5	Zvolí upozornění o splněném přání.	
A6		Zobrazí obrazovku s detailem splněného přání.

1.4.4.6 Alternativní toky

#	Uživatel	Systém
A3.1	Uvidí gratulační obrazovku push notifikace s gratulačním textem a stiskne ho.	



Obrázek 1.6: Diagram aktivit, popisující splnění přání uživatele

1.4.5 UC5: Zobrazení detailu přání

1.4.5.1 Popis

Zobrazení názvu, popisu, fotografie, jména vlastníka, počtu přátel, kteří přispěli penězi, sumy, která již byla vybrána, zbývajících času a posledních komentářů k přání uživatele.

1.4.5.2 Počáteční podmínky

Uživatel přihlášený do aplikace se zobrazeným seznamem buď svých přání, nebo přání svých přátel.

1.4.5.3 Koncové podmínky

Uživatel vidí detail přání.

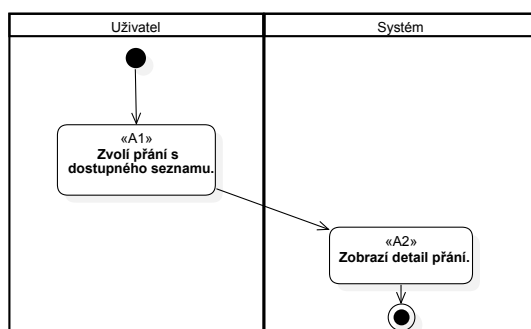
1.4.5.4 Aktér

Přihlášený uživatel.

1. ANALÝZA

1.4.5.5 Popis aktivit

#	Uživatel	Systém
A1	Zvolí přání ze seznamu dostupných přání.	
A2		Zobrazí detail přání.



Obrázek 1.7: Diagram aktivit, popisující zobrazení detailu přání

1.4.6 UC6: Zobrazení seznamu komentářů ke konkrétnímu přání

1.4.6.1 Popis

Zobrazení seznamu všech komentářů k přání, které zvolil uživatel.

1.4.6.2 Počáteční podmínky

Uživatel přihlášený do aplikace se zobrazeným detailem samotného přání.

1.4.6.3 Koncové podmínky

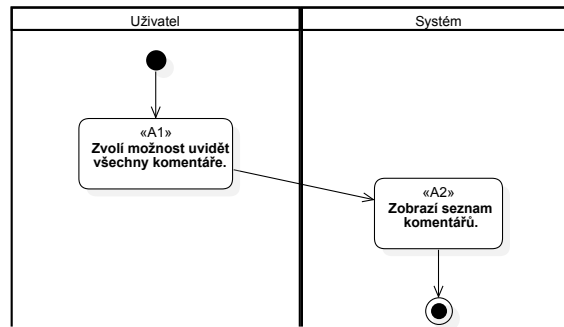
Uživatel vidí seznam všech komentářů k samotnému přání.

1.4.6.4 Aktér

Přihlášený uživatel.

1.4.6.5 Popis aktivit

#	Uživatel	Systém
A1	Zvolí možnost zobrazit všechny komentáře.	
A2		Zobrazí seznam komentářů.



Obrázek 1.8: Diagram aktivit, popisující zobrazení seznamu komentářů ke konkrétnímu přání

1.4.7 UC7: Napsání komentáře ke konkrétnímu přání

1.4.7.1 Popis

Uživatel má možnost napsat vlastní komentář k přání.

1.4.7.2 Počáteční podmínky

Uživatel přihlášený do aplikace se zobrazeným seznamem dalších komentářů.

1.4.7.3 Koncové podmínky

Systém přidává komentář uživatele do seznamu již napsaných komentářů.

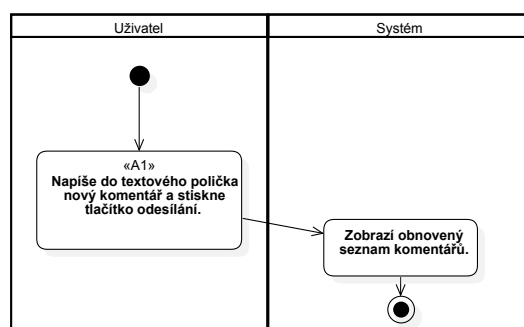
1.4.7.4 Aktér

Přihlášený uživatel.

1.4.7.5 Popis aktivit

#	Uživatel	Systém
A1	Napíše do textového políčka nový komentář a stiskne tlačítko pro odeslání.	
A2		Zobrazí obnovený seznam komentářů.

1. ANALÝZA



Obrázek 1.9: Diagram aktivit, popisující napsání komentáře ke konkrétnímu přání

1.4.8 UC8: Zobrazení seznamu kamarádů uživatele

1.4.8.1 Popis

Uživatel má možnost prohlédnout si seznam svých kamarády.

1.4.8.2 Počáteční podmínky

Uživatel přihlášený do aplikace.

1.4.8.3 Koncové podmínky

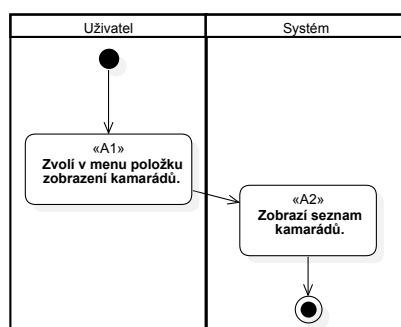
Uživatel vidí seznam svých přátel.

1.4.8.4 Aktér

Přihlášený uživatel.

1.4.8.5 Popis aktivit

#	Uživatel	Systém
A1	Zvolí v menu položku zobrazení kamarádů.	
A2		Zobrazí seznam kamarádů.



Obrázek 1.10: Diagram aktivit, popisující zobrazení seznamu kamarádů uživatele

1.4.9 UC9: Zobrazení seznamu přátel kamarádů

1.4.9.1 Popis

Uživatel má možnost zobrazit si přání kamaráda.

1.4.9.2 Počáteční podmínky

Uživatel přihlášený do aplikace.

1.4.9.3 Koncové podmínky

Uživatel vidí seznam přání zvoleného kamaráda.

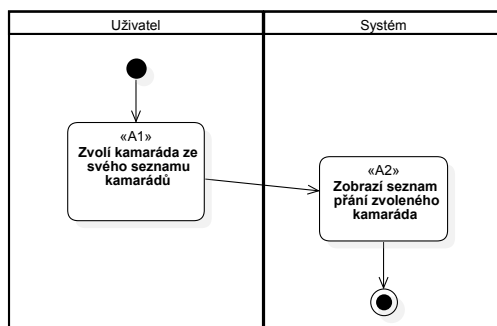
1.4.9.4 Aktér

Přihlášený uživatel.

1.4.9.5 Popis aktivit

#	Uživatel	System
A1	Zvolí kamaráda ze svého seznamu kamarádů.	
A2		Zobrazí seznam přání zvoleného kamaráda.

1. ANALÝZA



Obrázek 1.11: Diagram aktivit, popisující zobrazení seznamu přátel kamarádů

1.4.10 UC10: Nastavení bankovního účtu uživatele

1.4.10.1 Popis

Uživatel má možnost nastavit si číslo bankovního účtu, na který chce dostat nasbíranou částku peněz.

1.4.10.2 Počáteční podmínky

Uživatel přihlášený do aplikace.

1.4.10.3 Koncové podmínky

Uživatel má nastavený bankovní účet, na který může dostat peníze.

1.4.10.4 Aktér

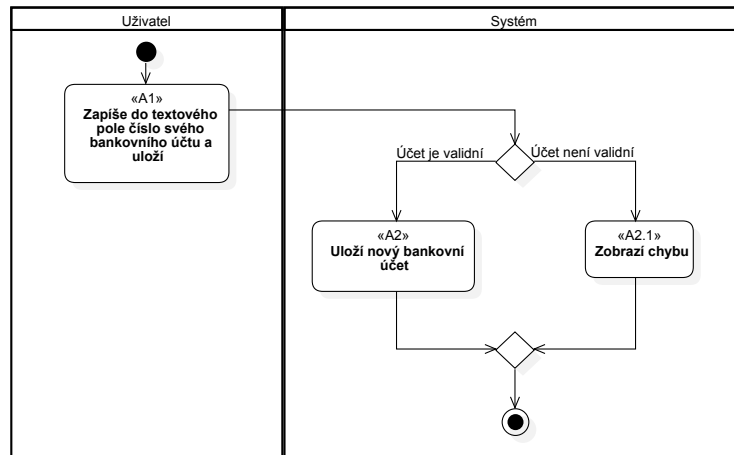
Přihlášený uživatel.

1.4.10.5 Popis aktivit

#	Uživatel	System
A1	Zapíše do textového pole číslo svého bankovního účtu a uloží.	
A2		Uloží nový bankovní účet.

1.4.10.6 Alternativní toky

#	Uživatel	System
A2.1		Zobrazí chybu.



Obrázek 1.12: Diagram aktivit, popisující nastavení bankovního účtu uživatele

1.4.11 UC11: Donace

1.4.11.1 Popis

Uživatel má možnost finančně přispět na přání.

1.4.11.2 Počáteční podmínky

Uživatel přihlášený do aplikace.

1.4.11.3 Koncové podmínky

Uživatel přispěl na přání.

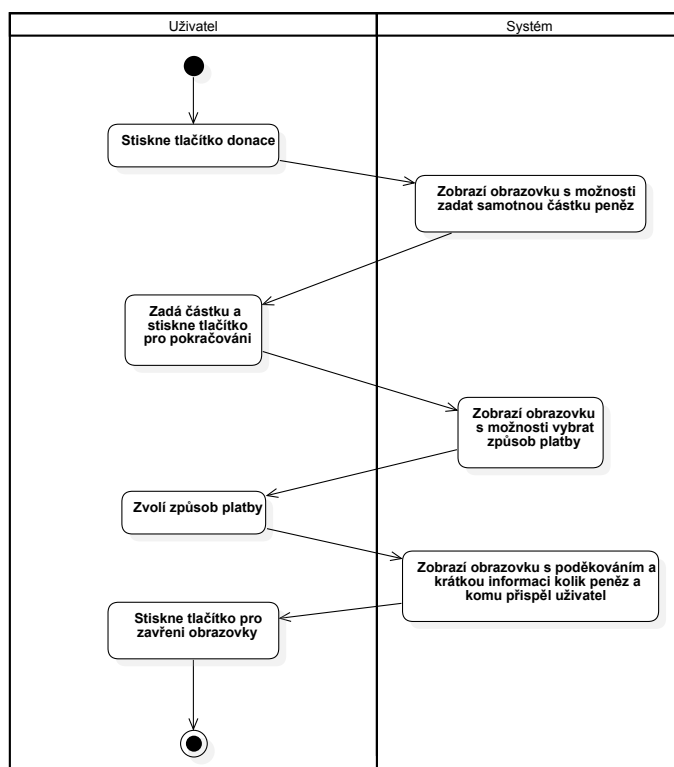
1.4.11.4 Aktér

Přihlášený uživatel.

1. ANALÝZA

1.4.11.5 Popis aktivit

#	Uživatel	Systém
A1	Stiskne tlačítko donace.	
A2		Zobrazí obrazovku s možností zadat samotnou částku peněz.
A3	Zadá částku a stiskne tlačítko pro pokračování.	
A4		Zobrazí obrazovku s možností vybrat způsob platby.
A5	Zvolí způsob platby.	
A6		Zobrazí obrazovku s poděkováním a krátkou informací kolik peněz a komu uživatel přispěl.
A7	Stiskne tlačítko pro zavření obrazovky.	
A8		Zobrazí domovskou obrazovku.



Obrázek 1.13: Diagram aktivit, popisující donace

1.5 Donace

ElateMe jako crowdfundingová platforma obsahuje možnost zaslání peněz na určité přání jiného uživatele. Se zadavatelem bylo prodiskutováno, jaké možnosti existují a které z nich jsou vhodné pro naše účely. Dále uvedu hlavní akceptační kritéria pro vhodný způsob platby v rámci našeho projektu.

- Nesmí obsahovat poplatky za transakci, protože chceme, aby částka, kterou poslal jeden uživatel druhému, došla celá.
- Musí být populární mezi lidmi, aby pro běžného uživatele nebyl problém najít svůj oblíbený způsob mezi poskytovanými.
- Musí poskytovat řešení pro mobilního klienta. Tímto řešením může být SDK pro platformy iOS a Android, nebo detailně zdokumentované a podporované REST API. To potřebujeme, abychom dokázali propojit platební službu s naším projektem.
- Musí být bezpečný.

V souladu s požadavkem zadavatele jsme vybrali platbu pomocí banky Fiobanka [4] a digitální měnu Bitcoin . Zkoumání platby přes banku má na starost kolega Gleb Arkhipov v rámci své bakalářské práce [5]. Mým úkolem je prozkoumat převod peněz pomocí digitální měny Bitcoin a všechno, co s tím souvisí.

1.5.1 Bitcoin

Před samotným popisem použití Bitcoin platby v aplikaci ElateMe, seznámím čtenáře s pojmem Bitcoin a fungováním této služby, jeho výhodami a nevýhodami oproti jiným způsobům platby a se zdroji informací, jelikož jsem před tím žádné zkušenosti s prací s Bitcoinem neměl.

1.5.2 Co je Bitcoin

V seznámení s Bitcoinem mi pomohla bakalářská práce Michala Rendla "Analýza digitální měny bitcoin z hlediska investiční příležitosti"[6]. Autor v ní popisuje Bitcoin takto:

„Bitcoin je P2P platební síť elektronické peněžní měny, která zároveň používá stejný název pro svou digitální měnu, které se často říká virtuální měna nebo kryptoměna. Tato síť je plně decentralizovaná, nemá žádnou centrální autoritu jako například administrátora, banku a podobně. Bitcoin mince je možné použít k úhradě zboží nebo služeb u prodejců, kteří jsou připraveni a ochotni je akceptovat. K dispozici jsou možnosti směny na obyčejné (papírové) měny pomocí burz, směnáren, tržišť bitcoinů a dalších specializovaných možností.“

K tomu bych ještě dodal, že bitcoin je nejznámější a nejrozšířenější kryptoměnou ve světě. Ohledně decentralizace chtěl bych doplnit, že hlavní výhodou plné decentralizace Bitcoinu je to, že nikdo, ani jednotlivec, skupina lidí nebo vláda nějakého státu nedokázali ovlivnit nebo ovládat tuto digitální měnu.

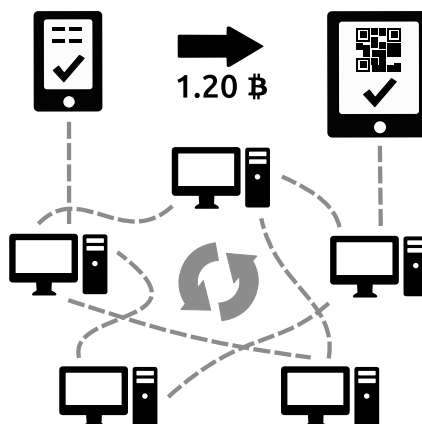
1.5.3 Základní princip fungování

Posílání transakcí funguje pomocí počítačového nebo mobilního klienta, který je propojen s dalšími klienty. Každý klient má peněženku, která slouží jako adresa pro platbu. Každá peněženka má veřejný a soukromý klíč, kterým podepisuje transakci, kterou chce poslat. Hash veřejného klíče se používá pro generování adresy.

V síti Bitcoin existují dvě různé uživatelské role: *koncoví uživatelé* a *těžaři*.

Ti první posílají a dostávají bitcoiny. Těžaři mají za úkol potvrzovat transakce v síti Bitcoin. Na webové stránce *bitcoin-bitcoiny.cz* [7] funkce těžaře popsána takto:

„Těžaři potvrzují transakce v síti. Těžař seskupí transakce čekající na potvrzení, přidá k nim odkaz na předchozí potvrzený blok transakcí a údaj zvaný kryptografická nonce. Snaží se najít takovou nonci, aby se hash nového bloku vešel pod síť stanovený limit. Limit je nastaven tak, aby se to v celé síti dařilo průměrně jednou za 10 minut, takže nalezení vhodné nonce je obtížné a to tím více, čím výkonnější celá síť je. Těžař, kterému se nalezení nonce a tím i vytvoření a potvrzení nového bloku transakcí podaří, si ponechá veškeré poplatky ze zahrnutých transakcí a odměnu za potvrzení bloku. Odměna za potvrzení bloku je momentálně 25 BTC a jedná se o jediný a předem stanovený způsob emise nových bitcoinů. Odměna se však každých 210 000 bloků (tj. každé 4 roky) snižuje na polovinu a růst množství peněz zpomaluje. Těžař si může vybrat, které transakce do nového bloku zahrne a které ne (podle výše poplatku), nyní je ale většina těžařů ochotna přijímat i transakce bez poplatku. Je důležité zmínit, že protože je síť anonymní, těžaři neví nic o odesílatelích ani příjemcích a jediným smysluplným kritériem pro zahrnutí nebo nezahrnutí transakce do bloku je právě zvolený poplatek. O tom, který těžař první nalezne vhodnou nonci (a rozhoduje o zařazení transakcí) rozhoduje náhoda. Transakční poplatky a odměna za potvrzení bloku jsou ekonomickou motivací činnosti těžařů. Snižováním odměny dojde ke stále většímu vyžadování transakčních poplatků.“



Obrázek 1.14: Ukázka fungování bitcoinů

Nicméně Bitcoin má mnoho výhod, ale nás zajímá v rámci implementaci aplikace ElateMe především možnost přeposílat bitcoiny na sdílený účet. Takže dále budu pokračovat v tom, že se budu snažit prozkoumat, jaké jsou možnosti pro řešení zaslání platby a jaké problémy při tom mohou nastat.

1.5.4 Výhody pro ElateMe

Když jsem od zadavatele dostal za úkol prozkoumat a integrovat Bitcoinů do ElateMe, nevěděl jsem, o co se jedná, a na otázku, proč to potřebujeme, jsem dostal odpověď, že pomocí Bitcoinů se můžeme vyhnout poplatkům za transakci, což dává smysl, protože nechceme uživatele ElateMe obtěžovat zbytečným přepočítáváním posílané částky.

Pro znázornění: Bob chce přispět Alici 500 Kč, ale Alice dostane jen 498,8 Kč kvůli poplatkům. Toto samozřejmě může vést ke zmatkům a nekonzistenci celkového konta, což já beru za velmi závažný problém. Absence poplatků je nejvýznamnějším důvodem, proč chceme použít Bitcoinů v ElateMe. Dále uvedu výhody použití Bitcoinů v našem projektu.

- Bitcoinů se převádějí přímo od osoby k osobě přes internet mimo jakoukoli banku a daně. To pro nás znamená, že poplatky jsou o mnoho nižší nebo žádné.
- Bitcoinů lze použít v jakékoli zemi.
- Vaše konto nemůže být zmrazené.
- Použití Bitcoinů nemá žádné podmínky nebo iracionální limity.
- Decentralizace. Žádná třetí strana neovlivní stav Bitcoinů.

Návrh

Následující kapitola se zabývá návrhem projektu. V analýze jsem zjistil funkční a nefunkční požadavky, vytvořil případy užití a seznámil čtenáře s Bitcoin platbou. S přihlédnutím na tyto věci mohu vytvořit návrh, který bude zobrazovat moji představu realizace iOS aplikaci ElateMe.

Níže uvádím cíle, kterých chci dosáhnout v návrhu:

- Prozkoumat SDK pro Bitcoin platbu.
- Popsat základní způsob testování síťové vrstvy.
- Navrhnout službu pro Bitcoin platbu pomocí Coinbase SDK.
- Navrhnout službu pro správu komentářů.
- Navrhnout službu pro správu notifikací o přáních.

Návrh zvláštních případů užití popíšu v následujícím formátu.

1. Popis. Popíšu případ užití z programátorského hlediska, tedy co to dělá a proč to potřebujeme.
2. Počáteční podmínky. Zadefinuji stav, ve kterém případ užití začíná a jaká data máme k dispozici pro použití při samotné implementaci.
3. Popis aktivit. Popíšu samotný návrh, který se bude skládat z jednotlivých aktivit (kroků), které vedou k žádanému výsledku, a sekvenčního diagramu, který bude znázorňovat jednotlivé kroky.
4. Návrh testování. Popíšu kritické sekce samotného návrhu a na základě toho navrhnu unit testy, které budou implementovány v kapitole **Testování**.

2.1 Práce s digitální měnou Bitcoin

V analýze jsem uvedl, co jsou Bitcoinů a jak je systém ElateMe bude používat. V této sekci popíši návrh přeposlání Bitcoinů z jednoho účtu na druhý. Pro práci s Bitcoin je nutně mít Bitcoin peněženku, pomocí které se budou vykonávat všechny platební úkony. Se zadavatelem jsme prodiskutovali existující Bitcoin peněženky a vybrali si Coinbase [8]. Hlavními přednostmi Coinbase je to, že zadavatel má účet, se kterým umí pracovat, a to, že Coinbase poskytuje SDK pro platformu iOS [9].

2.1.1 Coinbase

Coinbase je Bitcoin peněženka, která umožňuje nakupovat bitcoiny a pracovat s nimi. Dále uvedu hlavní výhody Coinbase:

- Coinbase funguje v 33 různých státech, včetně České republiky.
- Coinbase používá více než 6.7 milionu lidí.
- Má iOS a Android aplikaci [10] pro rychlou a snadnou práci s bitcoiny.
- Při odesílání a přijímání bitcoinů proběhne okamžitá konverze na místní měnu.
- Uživatel má plnou kontrolu nad soukromými klíči.
- Rychlá a bezplatná registrace.
- Poskytuje SDK pro různé platformy pro práci s jejich službami.

2.1.2 Coinbase SDK

Coinbase SDK je soubor nástrojů pro práci s Coinbase API [11], který umožní snadno integrovat práci s bitcoiny do iOS projektu. Vše, co je popsáno v API dokumentaci, je v něm implementováno. Jde o předdefinované funkce, které by programátor jinak musel napsat sám, avšak Coinbase je připravil za nás, což já osobně považuji za velmi pozitivní aspekt, protože nám to ušetřilo minimálně několik dnů práce.

Před začátkem práce s Coinbase SDK je potřeba založit vývojářský účet na webu Coinbase [12].

2.1.2.1 Registrace

Pro to, abych mohl použít Coinbase SDK ve své aplikaci, musím přidat ElateMe aplikaci do složky "My OAuth2 Applications" na portálu Coinbase. Dále popíšu celý postup po krocích:

1. Zvolit složku "Settings".

2. Zvolit podsložku "API Access".
3. Kliknout na tlačítko "+ New OAuth2 Application"
4. Doplnit povinné údaje o aplikaci, ve které chceme používat Coinbase SDK.
 - a) Napsat název aplikace.
 - b) Napsat popis aplikace.
 - c) Přidat URI pro přesměrování zpět do aplikace [13].
 - d) Napsat jméno autora nebo název firmy, která implementuje onu aplikaci.
 - e) Vybrat kategorii, do které aplikace patří.
5. Případně přidat další nepovinné údaje.
 - a) Přidat ikonu aplikace.
 - b) Přidat web stránku aplikace.
 - c) Přidat URL pro notifikace.
 - d) Přidat web stránku vývojářů, kteří za aplikací stojí.

Po registraci aplikace nám Coinbase portál vygeneruje hexadecimální řetězce *ClientID* a *ClientSecret*, které se používají pro přístup do Coinbase API.

2.1.2.2 Dokumentace Coinbase SDK

Pro orientaci při práci s Coinbase SDK využijeme oficiální dokumentaci Coinbase API na stránce vývojářský portál a návod na Github stránce Coinbase SDK.

2.1.3 Možné problémy

Níže uvádím věci, které se mohou stát problémem při implementaci.

- Na Githubu SDK je popsán problém [14], že nefunguje metoda `sendAmount:to:completion:`. Problém byl evidovaný 10. května 2016 a stále není označený jako vyřešený.
- Poslední verze SDK byla vydaná v září roku 2016, což může znamenat nepříliš aktivní podporu.
- SDK pro iOS verze používá Coinbase API v1, ale na svých webových stránkách Coinbase uvádí, že můžeme používat již Coinbase API v2 [15].

Na závěr shrnu, že Coinbase SDK pro iOS verzi nevypadá jako projekt, který by jeho zakladatelé aktivně podporovali. Tento faktor může ovlivnit implementaci Bitcoin platby v projektu ElateMe.

2.2 Testování síťové vrstvy

Aplikace ElateMe v průběhu používání neustále komunikuje s backendem přes HTTP protokol. Tuto komunikace používá pro získávání, posílání a obnovování dat, například pro poslání nového přání nebo získání nových notifikací. Po každé, když se aplikace pokusí komunikovat s backendem, mohou nastat následující scénáře:

- Komunikace proběhla úspěšně.
- Klient poslal chybná data a backend vrátí chybu.
- Klient nemá připojení k internetu a komunikace nemůže proběhnout.
- Backend není dostupný a vrátí chybu na jakýkoliv poslaný dotaz.
- Backend změnil formát odpovědi a neupozornil o tom klienta. Kvůli tomu klient nemůže zpracovat aktuální odpověď.

Na všechny tyto scénáře musí klient umět reagovat, a to správným způsobem. Ruční testování každého z těchto scénářů by bylo časově náročné, bereme-li v úvahu, že chceme testovat každou síťovou komunikaci. Proto by bylo vhodné použít unit-testy.

2.2.1 Unit-testy

Unit-testy jsou programové testy, které zahrnují rozdělení programu na části (jednotky) a jejich následovné otestování. Pro velké projekty se rozklad programu na části může skládat z několika kroků, než je dosaženo toho, že je každá testovaná část nezávislá a odpovídá pouze jednomu případu užití. Příkladem pro ElateMe může být získání přání uživatele z backendu nebo zobrazení kamarádů. Naopak vytvoření přání je příliš velký use-case, který není triviální otestovat pomocí unit-testů. Proto je lepší rozdělit jej na menší části, například na doporučení produktu podle názvu přání, doplnění údajů přání a odeslání nového přání na server. Pomocí unit-testů se v projektu obecně testují následující věci:

- Klíčová funkcionalita. To mohou být třídy modelu, jejich metody a interakce s dalšími třídami v programu.
- Hraniční podmínky. Například horní hranice pro datum vypršení přání při jeho vytvoření.
- Oprava chyb.

2.3 Návrh Bitcoin platby

2.3.1 Popis

Jakožto crowdfundingová platforma umí aplikace ElateMe posílat bitcoiny na účet.

Níže uvádím seznam zkratk, které dále používám v textu pro jednodušší čtení:

- BitcoinPaymentService objekt - Služba.
- Coinbase object - Bitcoin systém.
- CoinbaseAccount objekt - Účet.
- Coinbase Transaction - Transakce.
- Unikátní řetězec s délkou 32 znaků - Token.

Transakce může mít několik stavů:

- Failed - chybový.
- Pending - čekající na potvrzení.
- Completed - ověřený a vyplněný.

2.3.2 Počáteční podmínky

Pro posílání bitcoinů použijeme funkci z Coinbase SDK

`CoinbaseAccount::sendAmount:to:completion:`. Pro inicializaci tohoto objektu potřebujeme token. Token aplikace obdrží, když se uživatel přihlásí do svého Coinbase účtu.

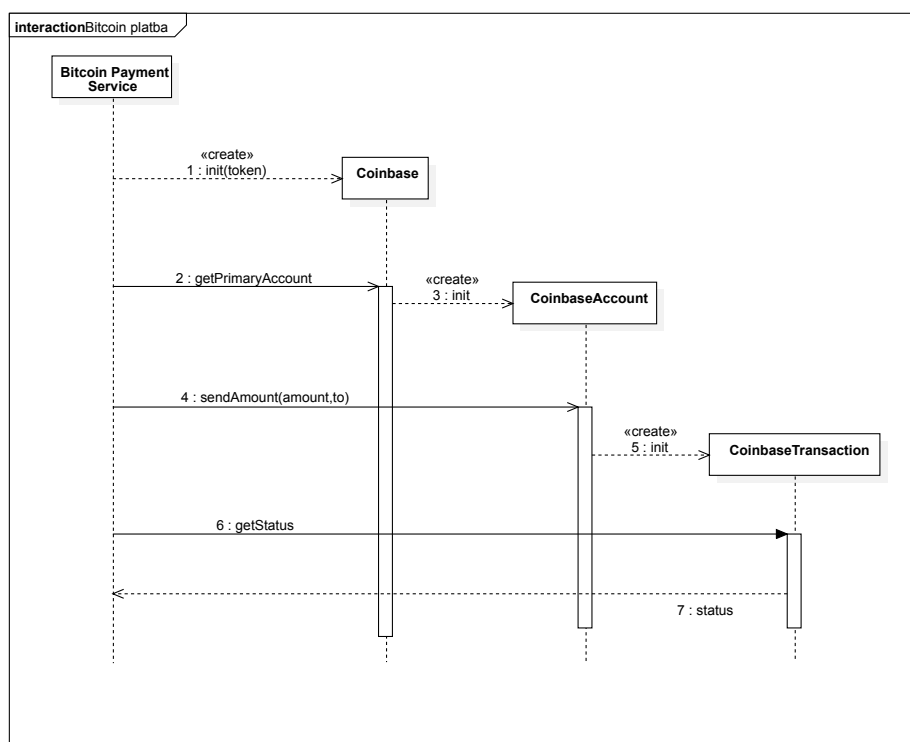
2.3.3 Popis aktivit

1. Služba inicializuje Bitcoin systém pomocí tokenu a po úspěšné inicializaci jej uloží.
2. Zavolá se metoda Bitcoin systému `getPrimaryAccount`.
3. Z Bitcoin systému se asynchronně vrátí objekt Účet a Služba jej uloží.
4. Servis zavolá metodu Účtu `sendAmount(amount:to:)`.
5. Účet asynchronně vrátí objekt Transakce a Služba jej uloží.
6. Služba zavolá metodu `getStatus` na Transakci.
7. Transakce vrátí svůj stav.

2. NÁVRH

Alternativní toky

- V 1. kroku Služba nedokáže inicializovat Bitcoin systém a vrátí chybu.
- V 3. kroku Bitcoin systém nedokáže inicializovat Účet a vrátí chybu.
- V 5. kroku Účet nedokáže inicializovat Transakci a vrátí chybu.



Obrázek 2.1: Sekvenční diagram, popisující Bitcoin platbu

2.3.4 Testování

V této části aplikace se budou testovat následující scénáře:

- Služba použije správné údaje a na konci dostane status Pending nebo Completed.
- Služba použije chybný Token při inicializaci Bitcoin systému a dostane chybu.
- Služba použije neexistující Token, který bude mít správný formát. Inicializace Bitcoin systému by v takovém případě měla proběhnout úspěšně, ale volání funkce `getPrimaryAccount` vrátí chybu.

- Služba použije chybnou adresu a dostane chybu při inicializaci Transakce.

2.4 Návrh komentování přání

2.4.1 Popis

Jakožto crowdfundingová platforma umožňuje aplikace ElateMe uživateli reagovat na přání, a to jak své, tak kamarádů, pomocí napsání komentářů.

Níže uvádím seznam zkratk, které dále používám v textu pro jednodušší čtení:

- Comment Service objekt - Služba.
- Comment object - Comment.
- Network Layer object - Network.
- ElateMe API - API.
- Identifikační číslo - id

2.4.2 Počáteční podmínky

Pro zobrazení komentáře k určitému přání použijeme JSON odpověď, kterou nám vrací server. JSON má následující formát.

```
1  [
2    {
3      "count": 2,
4      "next": null,
5      "previous": null,
6      "results": [
7        {
8          "id": 1,
9          "author": 1,
10         "wish": 1,
11         "text": "Me too",
12         "date_created": "2017-02-10T15:46:33.854478Z"
13       },
14       {
15         "id": 2,
16         "author": 3,
17         "wish": 1,
18         "text": "But why?",
19         "date_created": "2016-12-22T15:46:33.854478Z"
20       }
21     ]
22   }
23 ]
```

2. NÁVRH

Tady lze vidět, že server posílá jen id autora komentáře. To znamená, že budeme potřebovat stáhnout ze serveru jednotlivé autory podle jejich id. Jakýkoliv autor komentáře je běžným uživatelem systému, takže pro to, abychom dostali ze serveru informaci o určitém autorovi, musíme poslat požadavek na server s id autorem. Odpověď ze serveru dostaneme v následujícím formátu JSON:

```
1 [
2   {
3     "id": 1,
4     "image": "https://img.url",
5     "first_name": "Name",
6     "last_name": "Surname",
7     "date_of_birth": "1996-01-01",
8     "email": "name.surname@test.com",
9     "gender": "M",
10    "place_of_birth": "City",
11    "bank_account": "1234567890\\0800"
12  }
13 ]
```

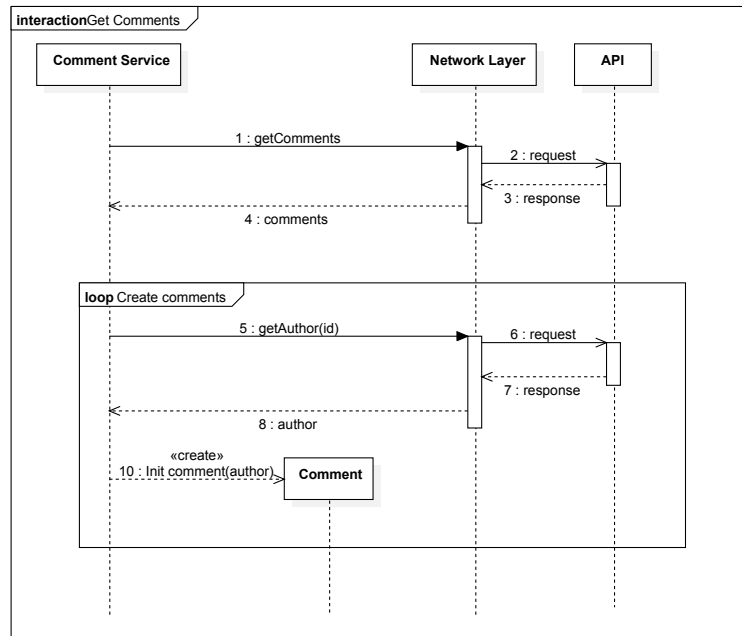
2.4.3 Popis aktivit

1. Služba zavolá metodu Networku `getComments:`.
2. Network požádá API o komentáře.
3. API vrátí Networku odpověď ve formátu JSON.
4. Network zpracuje odpověď a pošle komentáře Službě.
5. Služba uloží komentáře a začne volat u Networku metodu `getAuthor(id)` pro každého autora v uložených komentářích.
6. Network požádá API o autora s určitým id.
7. API vrátí Networku odpověď ve formátu JSON.
8. Network zpracuje odpověď a pošle autora Službě.
9. Služba vytvoří Comment pomocí autora.

Alternativní toky

- V 3. kroku API vrátí chybu.
- V 4. kroku Network nedokáže zpracovat komentáře z API a vrátí chybu.
- V 8. kroku Network nedokáže zpracovat autora z API a vrátí chybu.

- V 7. kroku API vrátí chybu.



Obrázek 2.2: Sekvenční diagram, popisující zobrazení komentářů

2.4.4 Návrh testování

V této části aplikace se budou testovat následující scénáře:

- Služba použije správné údaje a na konci dostane seznam komentářů k určitému přání.
- API vrátí komentáře ve špatném formátu a Network na základě toho vrátí Službě chybu.
- API vrátí autora ve špatném formátu a Network na základě toho vrátí Službě chybu.
- Služba dostane chybu místo autora a zahodí komentář náležící k danému autorovi.
- API vrátí chybu po odeslaném požadavku na něj.

2.5 Návrh notifikace

2.5.1 Popis

Jakožto crowdfundingová platforma umožňuje aplikace ElateMe uživateli sledovat novinky, které jsou pro něj v rámci systému ElateMe důležité. Tyto informace bude ElateMe poskytovat ve formě notifikací. Celkově existuje 5 druhů notifikací:

- Notifikace o splněním přání uživatele.
- Notifikace o novém příspěvku na přání uživatele.
- Notifikace o novém komentáři k přání uživatele.
- Notifikace o vypršení času, který uživatel stanovil pro své přání.
- Notifikace o splněním přání, na které uživatel přispěl.

Níže uvádím seznam zkratk, které dále používám v textu pro jednodušší čtení:

- Notifacation Service objekt - Služba.
- Notifacation object - Notifikace.
- Network Layer object - Network.
- ElateMe API - API.
- Identifikační číslo - id

2.5.2 Počáteční podmínky

Pro zobrazení notifikace použijeme JSON odpověď, který nám vrací server. JSON má následující formát.

```
1  [  
2  "results": [  
3    {  
4      "notification_type": "my_wish_completed",  
5      "wish": {  
6        "id": 1,  
7        "author": 1,  
8        "images": [  
9          {  
10         "id": 1,  
11         "image": "http://image1.url",  
12         "date_created": "2017-02-09T17:38:19.446691Z"  
13       }  
14     ]  
15   },  
16 ]
```



```
15         "amount_gathered": 0,
16         "surprise_wish": null,
17         "title": "iPhone 7",
18         "description": "I want dis, I need dis.",
19         "amount_needed": 18888,
20         "date_created": "2017-02-09T17:27:27.097937Z",
21         "date_of_expiration": null,
22     "date_completed": null,
23     "is_public": false,
24     "donators_count": 5
25     }
26 }
27 ]
28 ]
```

Podobně jako v návrhu zobrazování komentářů nám backend u notifikací posílá pouze id jejich autora. Řešení bude rovněž stejné jako u komentářů. Budeme pro každou notifikaci stahovat každého autora zvlášť.

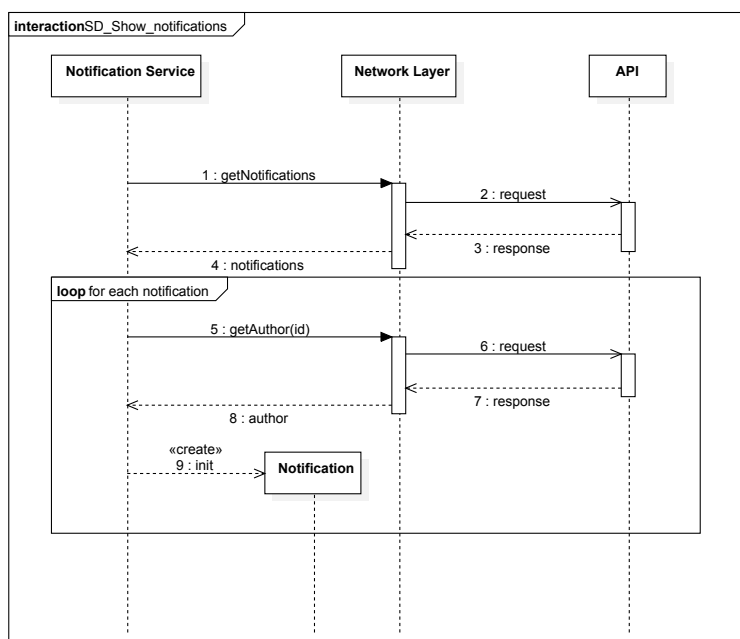
2.5.3 Popis aktivit

Návrh získání notifikací bude vypadat podobně jako získání komentářů, které jsem popisoval v předchozí sekci. Hlavní rozdíl bude ve zpracování druhu notifikace, protože komentáře nemají narozdíl od notifikací různé druhy. Až na tento rozdíl je získání notifikací téměř stejné.

1. Služba zavolá metodu Networku `getNotifications:`.
2. Network požádá API o notifikace.
3. API vrátí Networku odpověď ve formátu JSON.
4. Network zpracuje odpověď a pošle notifikace Službě.
5. Služba uloží notifikace a začne volat metodu Networku `getAuthor(id)` pro každého autora v uložených notifikacích.
6. Network požádá API o autora s určitým id.
7. API vrátí Networku odpověď ve formátu JSON.
8. Network zpracuje odpověď a pošle autora Službě.
9. Služba vytvoří Notifikace pomocí autora.

Alternativní toky

- V 3. kroku API vrátí chybu.
- V 4. kroku Network nedokáže zpracovat notifikace z API a vrátí chybu.
- V 8. kroku Network nedokáže zpracovat autora z API a vrátí chybu.
- V 7. kroku API vrátí chybu.



Obrázek 2.3: Sekvenční diagram, popisující zobrazení seznamu notifikace

2.5.4 Návrh testování

V této části aplikace se budou testovat následující scénáře:

- Služba použije správné údaje a na konci dostane seznam notifikací.
- API vrátí notifikace ve špatném formátu a Network na základě toho vrátí Službě chybu.
- API vrátí autora ve špatném formátu a Network na základě toho vrátí Službě chybu.
- Služba dostane chybu místo autora a zahodí notifikaci náležící k danému autorovi.
- Služba nedokáže rozpoznat druh notifikace a zahodí ji.

- API vrátí chybu po odeslání požadavku na něj.

Implementace

3.1 Použité nástroje

V této kapitole popíši nástroje a programy, které jsem používal v průběhu samotného vývoje.

3.1.1 Xcode

Xcode [16] je vývojové prostředí, které obsahuje balíček vzájemně propojených softwarových programů. On umožňuje vývojáři programovat aplikace pro iPhone, iPad, Mac, Apple Watch a Apple TV zařízení. Samotný Xcode obsahuje textový editor pro napsání kódu, kompilátor, simulátor a mnoho dalších nástrojů pro testování, ladění a zkoumání aplikací. Do toho patří i nahrání programu na zařízení, podepsání certifikátem a zaslání aplikace na App Store, obchod s aplikacemi. V průběhu celého vývoje iOS aplikaci ElateMe jsem používal Xcode jako klíčový nástroj pro programování.

3.1.2 Git

Na implementaci iOS aplikace jsem pracoval spolu s dalším členem ElateMe týmu Glebem Arkhipovem. Spolupráce na jednom projektu v několika lidech zahrnuje spojení výsledků práce každého z vývojářů do jednoho velkého výsledku. Rovněž je vhodné mít uložený každý z těchto výsledků zvlášť pro archivaci, čili je verzovat. Proto bylo zvoleno řešení v průběhu vývoje používat verzovací nástroj Git [17].

V rámci vývoje iOS aplikaci ElateMe byl zvolená následující strategie řízení Git repozitáře. V repozitáři jsou 2 hlavní větve – master a develop. Master větev používáme pro verzování verzí aplikace zaslaných zadavateli. To znamená, že pokaždé, když vytvoříme distribuční verzi, sloučíme všechny změny do master větve. Develop větev používáme pro verzování vývojářských verzí. To znamená, že po každé implementované funkcionalitě v programu,

pošle vývojář své změny na sdílenou develop větev, aby i ostatní vývojáři měli přístup k nově implementovaným částem. Každý vývojář, který pracuje nad projektem, zavádí vlastní větev, na které pracuje, a pak sloučí všechny změny do develop větve. Takovým způsobem se dodržuje dekomponovanost repositáře – dva vývojáři nikdy nepracují na stejné větvi a konflikty mezi změnami se tak stávají málokdy a historie změn je mnohem přehlednější.

3.1.3 Apiary

Pro práci s backendem jsme potřebovali jednoznačný zdroj informací, kde bychom zjistili vše potřebné o komunikaci mezi mobilním klientem a serverem. Na základě doporučení zadavatele jsme použili službu Apiary [18], která umožňuje navrhnout komunikaci klientů s API backendu, důkladně ji zdokumentovat a přidat příklady komunikace, které lze použít jako testovací data. V průběhu celého vývoje jsem byl s touto službou naprosto spokojen a s radostí budu pokračovat v jejím používání.

3.1.4 Knihovny a CocoaPods

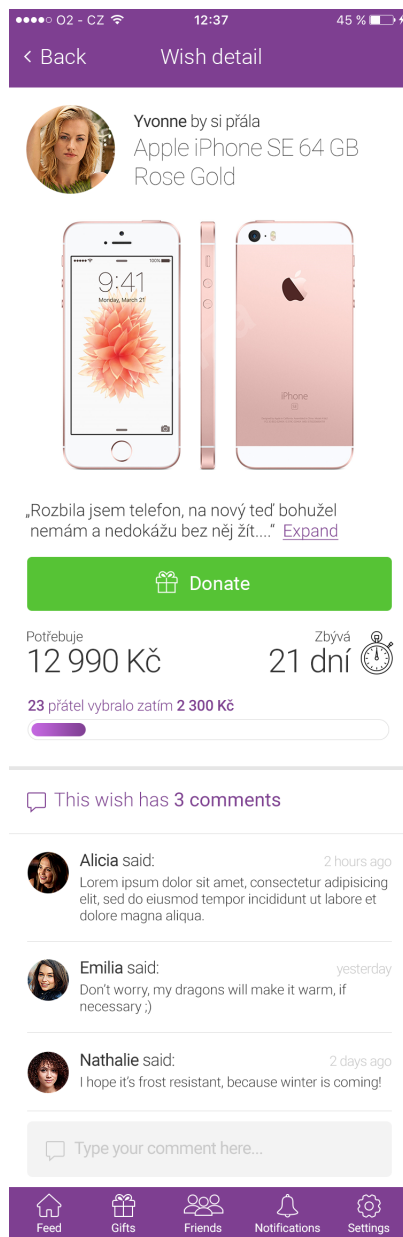
V každém IT oboru existují funkcionality, jejichž používání se často opakuje. Například práce s databází nebo práce z internetovou sítí. Je vhodné tuto funkcionalitu v různých projektech implementovat stejně, aby se ušetřil čas samotného vývoje. Na druhou stranu, když by některý programátor chtěl něco v tomto kódu zlepšit, musel by přidávat takto vylepšenou funkcionalitu do každého projektu. V případě, že existuje víc než 2 projektů se sdíleným kódem, mohou vzniknout problémy. Proto je v projektech vhodné používat knihovny, které v sobě danou funkcionalitu zahrnují. iOS vývoj není výjimkou, také existuje nepřeberné množství různých knihoven a frameworků pro práci se sítí, databázemi, grafickým rozhraním a spoustou dalšího. V ElateMe používáme na řízení knihoven systém CocoaPods [19], který vystupuje jako manažer závislostí pro iOS projekty. CocoaPods drží kolekce knihoven, jejichž počet se pohybuje v jednotkách tisíců. CocoaPods řídí linkování knihoven s hlavním projektem způsobem, že vytvoří centralizovanější ekosystém projektu. Knihovny, které chceme používat, zapíšeme do textového souboru s názvem Podfile. Poté stačí v terminálu spustit příkaz `pod install` a CocoaPods vloží zdrojový kód vypsáných knihoven do projektu.

3.2 Implementace zpracování a komentování přání

3.2.1 Popis

Uživatel zvolí přání. Ukáže se obrazovka s detailem přání. Dolu se objeví komentáře k samotnému přání. Pod komentáři se bude nacházet textové pole

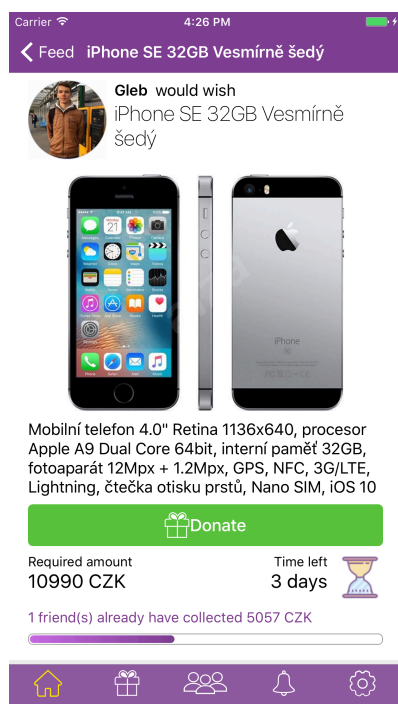
3.2. Implementace zpracování a komentování přání



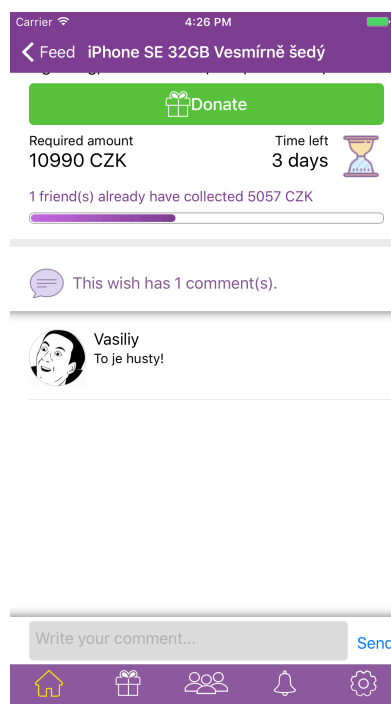
Obrázek 3.1: Grafický návrh vzhledu detailu přání s komentáři

pro napsání nového komentáře. Uživatel napíše do textového pole nový komentář a stiskne tlačítko odeslání. V případě, že se to nepodaří, aplikace ukáže uživateli hlášku s popisem problému.

3. IMPLEMENTACE



Obrázek 3.2: Horní část detailu přání



Obrázek 3.3: Spodní část detailu přání s komentáři

3.2.2 Grafický vzhled

Oproti grafickému návrhu, implementovaná grafika má drobné rozdíly, které jsou schválené zadavatelem.

3.2.3 Výsledek

Podle zadání bylo potřeba implementovat možnost uživateli komentovat jakékoliv přání a ukazovat samotné přání. Dál uvedu akceptační kritéria, které byly splněny a které nebyly.

Následující akceptační kritéria jsou splněné:

- Uživatel po otevření detailu přání uvidí seznam komentářů pod samotným přáním.
- Uživatel může napsat komentář k vybranému přání.
- Po stisknutí tlačítka odesílání komentáře aplikace pošle komentář na server.
- Po úspěšném odeslání komentáře na server, uživatel uvidí v seznamu svůj komentář.

- V případě, že v průběhu odesílání se stane nějaká chyba, uživatel uvidí hlášku s chybou.

Dál uvedu funkcionalitu, pomocí které se dá vylepšit daný případ užití.

- Uživatel může smazat napsaný komentář.
- Komentáře se načítají dynamické. Po přečtení 30 komentářů se načítají další 30 komentářů a tak až nebudou načtené všechny.
- Komentáře nebudou součástí obrazovky s detailem přání, ale budou na zvláštní obrazovce. Na ni uživatel bude moct přejít s detailu.

3.3 Správa novinek

3.3.1 Popis

Uživatel zvolí složku notifikace, která bude obsahovat seznam novinek. Novinky budou různých druhů dle grafického návrhu. Po stisknutí tlačítka na samotné přání, uživatel přejde do obrazovky s detailem přání. Po přetáčení obrazovky dolů se obnoví seznam notifikace.

3.3.2 Grafický vzhled

Oproti grafickému návrhu, implementovaná grafika má drobné rozdíly, které jsou schválené zadavatelem. Na místě obrázku v implementační verzi jsou obrázky samotných přání, ke kterým patří novinky. Požadavek na implementaci novinek byl splněn.

3.3.3 Výsledek

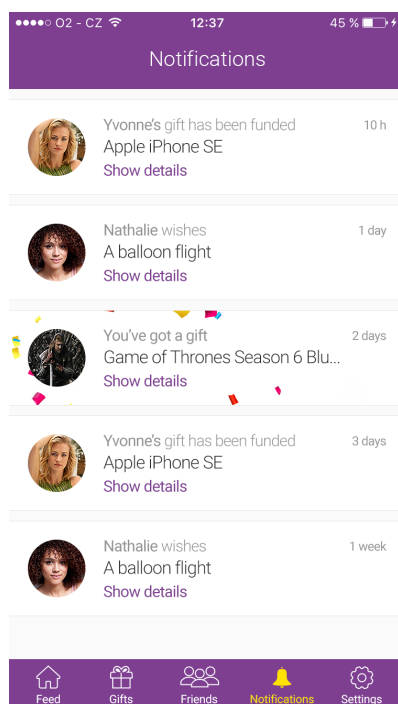
Následující akceptační kritéria jsou splněné:

- Uživatel zvolením složky notifikace uvidí seznam novinek.
- Po stisknutí tlačítka ukázání detailu, otevře se obrazovka se samotným přáním.
- Po přetáčení obrazovky dolů se obnoví seznam notifikace.
- V případě, že při načítání se stane nějaká chyba, uživatel uvidí hlášku s popisem problému.

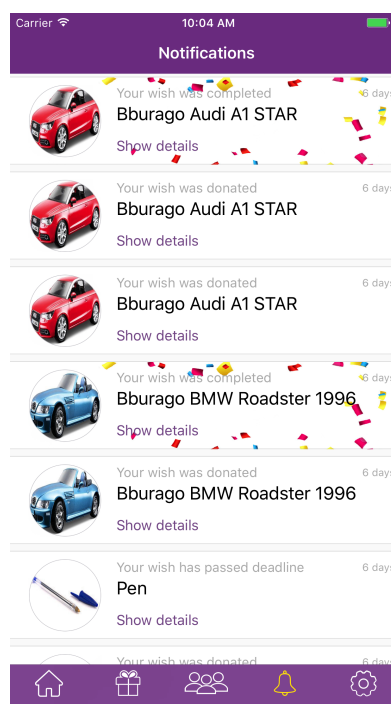
Dál uvedu funkcionalitu, pomocí které se dá vylepšit daný případ užití.

- Uživatel může skrýt novinku.
- Novinky se načítají dynamické. Po přečtení 30 novinek se načítají další 30 novinek a tak až nebudou načtené všechny.

3. IMPLEMENTACE



Obrázek 3.4: Grafický návrh seznamu notifikací



Obrázek 3.5: Seznám notifikací

- Systém bude upozorňovat uživatele o novince pomocí push notifikací.

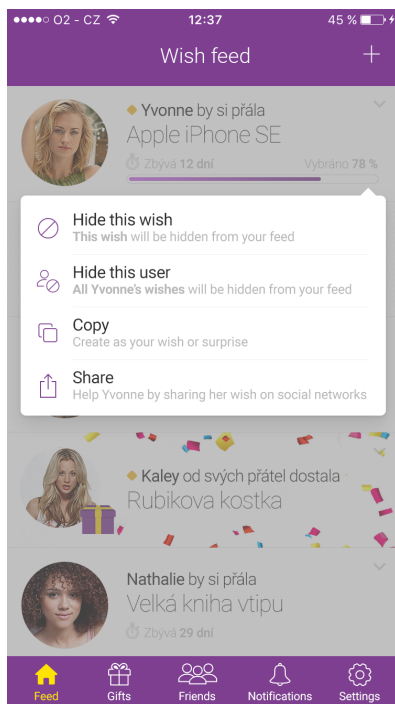
3.4 Správa přání

3.4.1 Popis

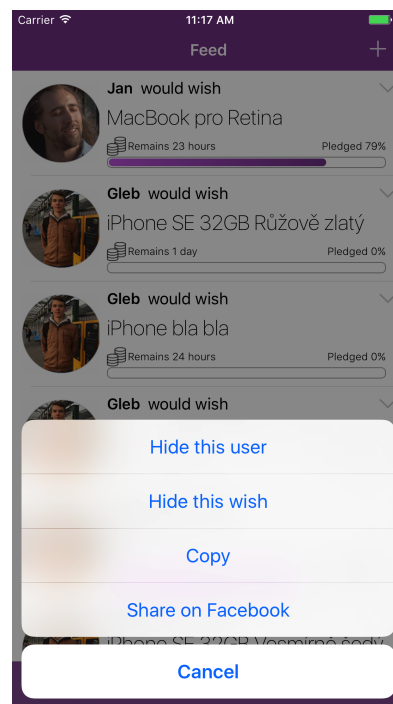
V aplikaci existuje možnost správy přání ve složce "Feed", která obsahuje seznam přání kamarádů. Správa zahrnuje v sobě následující funkce:

- Možnost skrýt určité přání.
- Možnost skrýt všechny přání od určitého uživatele.
- Možnost zkopírovat přání a pak na základě něho vytvořit vlastní.
- Možnost sdělit přání na sociální síť Facebook.

Uživatel přijde na to stisknutím tlačítka v pávem horním rohu buňky s samotným přáním. Ukáže se modální okno s seznamem možností zprávy přání.



Obrázek 3.6: Grafický návrh správy přání



Obrázek 3.7: Správa přání

3.4.2 Grafický vzhled

Oproti grafickému návrhu, implementovaná grafika má určitý rozdíl. S zadavatelem bylo prodiskutováno grafický návrh, který byl udělán na základě grafických pravidel pro zařízení s operačním systémem Android. Proto byl zvolena nativní iOS komponenta **UIAlertController**, kterou poskytuje grafická knihovna UIKit [20]. Na UX testech bylo ověřené, že uživatel, který používá chytré zařízení s operačním iOS, je zvyklý na tuto komponentu a žádný problem s tím nebyl nalezen.

3.4.3 Výsledek

Následující akceptační kritéria jsou splněné:

- Po stisknutí tlačítka v pravém horní rohu buňky s přáním, otevře se obrazovka nativní komponenta se seznamem možností pro správu přání.
- Po zvolení skrytí určitého přání, provede se dotaz na server, a přání se skryje.

3. IMPLEMENTACE

- Po zvolení skrytí všech přání od konkrétního uživatele, provede se dotaz na server, a obnoví se seznam, ve kterém nebudou žádná přání od tohoto uživatele.
- Po zvolení kopírování určitého přání, uživatel bude automatické přesměrovaný do obrazovky vytvoření přání, kde budou předvyplněné položky s informací na základě vybraného přání.
- Po zvolení sdílení určitého přání na Facebook, otevře se okno s možností doplnit zveřejnění nějakým komentářem. Potom uživatel zvolí odeslat a na stránce jeho Facebook účtu se objeví vyhlášení s vybraným přáním.
- V případě, že při dotazech na server se stane nějaká chyba, uživatel uvidí hlášku s popisem problému.

Dál uvedu funkcionalitu, pomocí které se dá vylepšit daný případ užití.

- Na UX testech se ukázalo, že uživateli dělá problém malé rozměry tlačítka pro správu přání. Zvětšení tlačítka mohlo by ovlivnit celkový design. Bylo by vhodné upravit návrh designu buňky přání pro jednodušší ovládání.

Testování

V této kapitole popíšu, jak probíhá testování iOS aplikace, s čím jsem se setkal v průběhu testování a popíšu výsledky unit testování [21].

4.1 Unit testování iOS aplikace

Vývojové prostředí Xcode, ve kterém probíhala celá implementace iOS aplikace, poskytuje kvalitní podporu pro napsání unit testů. Pro napsání jednotlivých testů v rámci jednoho případu užití se vytvoří třída, zděděná od třídy **XCTest**. Tato třída obsahuje funkce **setUp** a **tearDown**. V těle první funkce se píše kód, který inicializuje všechny objekty, které se budou testovat. V těle **tearDown** se píše kód, který uvolní inicializované objekty v **setUp** funkci. Pak se napíší samotné funkce, které budou zahrnovat v kódu logickou část. Tato část bude testovat jeden scénář. Takové scénáře jsem popsal v kapitole návrhu. Dál ukážu ukázkou kódu, jak vypadá základ pro unit test.

```
class NotificationsTests: XCTestCase {  
  
    override func setUp() {  
        super.setUp()  
        // Put setup code here. This method is called  
        // before the invocation of each test method  
        // in the class.  
    }  
  
    override func tearDown() {  
        // Put teardown code here. This method is called  
        // after the invocation of each test method  
        // in the class.  
        super.tearDown()  
    }  
}
```

```
func testExample() {
    // This is an example of a functional test case.
    // Use XCTAssert and related functions to verify
    // your tests produce the correct results.
}

func testPerformanceExample() {
    // This is an example of a performance test case.
    self.measure {
        // Put the code you want to measure the time
        // of here.
    }
}
}
```

4.2 Testování Bitcoin platby

V této sekci já bych chtěl uvést, že v průběhu testování Bitcoin plateb pomocí Coinbase SDK jsem našel zásadní problém ohledně funkčnosti knihovny. Po odesílání platby, funkce objektu **CoinbaseAccount** `sendAmount` nevrátí nic, ani chybu, ani objekt **Transaction** a po 5 sekundách čekání testovací funkce `waitForExpectations` vrátí chybu. V návrhu jsem popisoval možné problémy, které mohou ovlivnit implementaci. Jedno s toho bylo, že na github stránce knihovny je popsán problém od uživatele, že funkce `sendAmount` nefunguje. Přesně toto stálo v našem případě. V testech bylo otestováno následující případy:

- Servis použije chybný Token při inicializaci Bitcoin systému a dostane chybu.
- Servis použije neexistující Token, který bude mít správný formát. Inicializaci Bitcoin systému má proběhnout úspěšně, ale dostane chybu při zavolání funkce `getPrimaryAccount`.

Následující testy se nepodařilo otestovat vůči problémem za strany Coinbase SDK

- Servis použije správné údaje a na konci dostane status Pending nebo Completed.
- Servis použije chybovou adresu a dostane chybu při inicializaci Transakce.

Závěr

Hlavním cílem této práce bylo vytvořit iOS aplikaci pro crowdfundingovou platformu ElateMe, která by sloužila klientem pro API backend. V průběhu vývoje byla udělána detailní analýza případů užití, funkčních a nefunkčních požadavku, Bitcoin platby a její použití v projektu ElateMe a taky byl udělán BPMN diagram, který popisuje funkčnost aplikace jako celku. Na základě analýzy byl proveden návrh Bitcoin platby pomocí Coinbase SDK, komentování přání a řízení notifikace včetně návrhů vhodných jednotkových testů. Po návrhu byla provedená samotná implementace, která v sobě zahrnovala realizaci všech známých případů užití podle dodaných grafických návrhů. Nakonec byly provedené jednotkové testy, které pomohli ošetřit chyby ve fázi implementaci.

Výsledek

Na konci vývoju za výsledek jsme dostali plně funkční prototyp, který umí dělat skoro všechno dle existujících případů užití. Dál uvedu případy užití, které chybí v aktuální iOS aplikaci.

- Bitcoin platba - framework, který poskytuje Coinbase platforma se aktivně nepodporuje a problém odesílání bitcoinu na účet, který zveřejnil jeden z jejich uživatelů před rokem, nikdo pořad neopravil.
- Stav po splnění přání. Zatím nebyl detailně domyšlený způsob předání peněz uživateli, u kterého se splnilo přání.

Možnosti dalšího vývoje

Jako možnosti dalšího vývoje uvedu následující zlepšení:

- Prozkoumat a zvolit jinou Bitcoin peněženku nebo framework, který by poskytoval možnost přeposílání bitcoinu na účet.

- Přidat možnost platby kartou.
- Napsat víc jednotkových testů.
- Přidat push notifikace.
- Zařídít developer certifikát iOS vývojáře pro možnost posílání enterprise verzí.
- Přidat průběžnou integrace.
- Poslat funkční verze na App Store.

Literatura

- [1] iOS Human Interface Guidelines [online]. [Cit. 2017-04-03]. Dostupné z: <https://developer.apple.com/ios/human-interface-guidelines/overview/design-principles/>
- [2] BPMN Specification [online]. [Cit. 2017-04-05]. Dostupné z: <http://www.bpmn.org>
- [3] Facebook SDK for iOS [online]. [Cit. 2017-04-06]. Dostupné z: <https://developers.facebook.com/docs/ios/>
- [4] Fio banka [online]. [Cit. 2017-04-06]. Dostupné z: <https://www.fio.cz>
- [5] Arkhipov, G.: *ElateMe - iOS client II: bachelor's thesis*. Czech Technical University in Prague, Faculty of information technology, 2017.
- [6] Rendla, M.: *Analýza digitální měny Bitcoin z hlediska investiční příležitosti: Bakalářská práce*. České vysoké učení technické v Praze, Fakulta informačních technologií, 2014.
- [7] Jak těžit Bitcoin? [online]. [Cit. 2017-04-09]. Dostupné z: <http://www.bitcoin-bitcoiny.cz/jak-tezit-bitcoiny/>
- [8] Coinbase [online]. [Cit. 2017-04-11]. Dostupné z: <https://www.coinbase.com>
- [9] Coinbase iOS SDK [online]. [Cit. 2017-04-12]. Dostupné z: <https://github.com/coinbase/coinbase-ios-sdk>
- [10] Coinbase mobile applications [online]. [Cit. 2017-04-12]. Dostupné z: <https://www.coinbase.com/mobile>
- [11] Coinbase API documentation [online]. [Cit. 2017-04-14]. Dostupné z: <https://developers.coinbase.com/api/v2>

- [12] Coinbase API Access [online]. [Cit. 2017-04-14]. Dostupné z: <https://www.coinbase.com/settings/api>
- [13] Redirect URIs for Native Apps [online]. [Cit. 2017-04-16]. Dostupné z: <https://www.oauth.com/oauth2-servers/redirect-uris/redirect-uris-native-apps/>
- [14] Issue: CoinbaseAccount sendAmount not working [online]. [Cit. 2017-04-20]. Dostupné z: <https://github.com/coinbase/coinbase-ios-sdk/issues/15>
- [15] Introducing Coinbase API v2 [online]. [Cit. 2017-04-21]. Dostupné z: <https://developers.coinbase.com/blog/2015/07/01/coinbase-api-v2>
- [16] Xcode 8 [online]. [Cit. 2017-04-22]. Dostupné z: <https://developer.apple.com/xcode/>
- [17] What is Git [online]. [Cit. 2017-04-23]. Dostupné z: <https://www.atlassian.com/git/tutorials/what-is-git>
- [18] Apiary [online]. [Cit. 2017-04-24]. Dostupné z: <https://apiary.io>
- [19] What is CocoaPods [online]. [Cit. 2017-04-27]. Dostupné z: <https://cocoapods.org>
- [20] UIKit [online]. [Cit. 2017-04-28]. Dostupné z: <https://developer.apple.com/reference/uikit>
- [21] iOS Unit Testing and UI Testing Tutorial [online]. [Cit. 2017-04-30]. Dostupné z: <https://www.raywenderlich.com/150073/ios-unit-testing-and-ui-testing-tutorial>

Seznam použitých zkratek

- GUI** Graphical user interface
- XML** Extensible markup language
- UX** User experience design
- IDE** Integrated development environment
- URI** Uniform Resource Identifier
- SDK** Software Development Kit
- JSON** JavaScript Object Notation
- API** Application programming interface
- REST** Representational State Transfer

Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	src	
	client-ios	zdrojové kódy implementace
	thesis	zdrojová forma práce ve formátu \LaTeX
	text	text práce
	thesis.pdf	text práce ve formátu PDF

Instalační příručka

ElateMe aplikace je stále ve fázi vývoje, takže jediná možnost, jak spustit iOS aplikaci ElateMe, je použít na to prostředí Xcode. Aby to dopadlo úspěšně, musí být splněné následující podmínky:

- Verze operačního systému počítače macOS 10.12 nebo vyšší.
- Verze integračního prostředí Xcode 8.3.2 nebo vyšší.
- Verze CocoaPods 1.2.0 nebo vyšší.

C.1 Instalace z CD

1. Najděte adresář `client-ios`.
2. Zkopírujte obsah tohoto adresáře na vhodné místo a otevřete jeho.
3. Spustíte příkaz `pod install` z příkazové řádky v zkopírovaném adresáře. CocoaPods stáhne a spojí všechny knihovny a frameworky.
4. Spustíte příkaz `open ElateMe.xcworkspace`. Otevře se Xcode s projektem.
5. V levém horním rohu zvolte simulátor a stisknete tlačítko `Run`. Projekt se zkompile a otevře se simulátor s aplikací ElateMe.