

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SOCIÁLNÍ SÍŤ PRO DOTAZNÍKOVÉ PRŮZKUMY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ HUVAR

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

SOCIÁLNÍ SÍŤ PRO DOTAZNÍKOVÉ PRŮZKUMY

SOCIAL NETWORKING FOR QUESTIONNAIRE SURVEYS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ HUVAR

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN HRUBÝ, Ph.D.

BRNO 2015

Abstrakt

Cílem této bakalářské práce je navrhnout a implementovat aplikaci pomocí experimentální knihovny CloudKit. Tato aplikace bude sloužit jako sociální síť pro komunitu zadavatelů a respondentů dotazníkových průzkumů. Implementovaná sociální síť je otestována na skupině uživatelů, kteří se podílejí na chodu této sítě. Přínosem této práce je testování knihovny CloudKit a její využitelnosti pro sociální sítě.

Abstract

The aim of this work is to design and implement an application using an experimental library CloudKit. This application will serve as a social network for community submitters and questionnaire surveys respondents. Implemented social network is tested on a group of users who are involved in running the network. The benefit of this work is to test the Library CloudKit and its relevance for social networks.

Klíčová slova

iOS, Xcode, CoreData, CloudKit, Průzkumy, Sociální síť, socialQ

Keywords

iOS, Xcode, CoreData, CloudKit, Surveys, Social network, socialQ

Citace

Lukáš Huvar: Sociální síť pro dotazníkové průzkumy, bakalářská práce, Brno, FIT VUT v Brně, 2015

Sociální síť pro dotazníkové průzkumy

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Martina Hrubého Ph.D. Uvedl jsem veškeré literární prameny a publikace, ze kterých jsem čerpal.

.....

Lukáš Huvar
20. května 2015

Poděkování

Tímto bych chtěl poděkovat svému vedoucímu Ing. Martinu Hrubému Ph.D. za poskytnutí potřebných informací, rad a připomínek při řešení bakalářské práce a tvorbě technické zprávy.

© Lukáš Huvar, 2015.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	2
2	Rozbor tématu	4
2.1	Struktura typizované aplikace v iOS	4
2.2	Archivace dat	5
2.3	Knihovna Core Data	5
2.4	Knihovna CloudKit	5
3	Návrh dotazníkového systému	10
3.1	Koncept sociální sítě socialQ	10
3.2	Koncept uložení dat	11
3.3	Koncept aplikace	14
4	Implementace systému	16
4.1	Filtrování aktuálních dotazníků	16
4.2	Stáhnutí a vyhodnocení výsledků	17
5	Testování	18
5.1	Testování sociální sítě socialQ	18
5.2	Výkonnostní testy služby CloudKit	19
5.3	Výsledky testů	20
5.4	Ukázky z aplikace	20
6	Závěr	23
A	Obsah CD	25

Kapitola 1

Úvod

V dnešním světě je získávání informací jedna z nejdůležitějších činností. Informace slouží k získání přehledu, nabytí vědomostí, vytvoření názorů a postojů. Pro sběr informací z velkých skupin lidí slouží různé typy dotazníků, testů nebo pozorování[8]. Dotazník je určen hlavně na hromadné získávání informací od velkého počtu osob, proto se řadí ke kvantitativním metodám výzkumu. Dotazník je výzkum prováděný formou kladení otázek a získávání odpovědí. Na začátku by měly být kladeny jednodušší nebo zajímavé otázky a další mohou být méně zajímavé. Na konci dotazníku by mělo být poděkování respondentovi. Výhodou elektronických dotazníků je jejich časová i ekonomická nenáročnost proti papírovým nebo telefonickým dotazníkům. Papírové dotazníky jsou materiálově a finančně náročnější. Nutné zajistit poštovní rozesílání nebo jinou formu dopravy k respondentům. U telefonických dotazníků je nutné mít proškolený personál k vedení telefonických dotazníků. Naproti tomu u elektronických dotazníků je distribuce lehčí a menší ekonomická náročnost.

Do základní terminologie řadíme respondenta a zadavatele[7]. Respondent je osoba, která vyplňuje dotazník. Zadavatel je osoba, která vytvoří, provádí, zpracovává a hodnotí výzkum. Ve většině případů osoba zadavatele nebývá pouze jedna osoba, ale je to více osob. Pokud se jedná o jednu osobu, výsledky prezentuje spolupracovníkům, kolegům nebo široké veřejnosti. Problém nastává hlavně s motivací respondentů, kteří nejsou vždy ochotni dotazníky vyplnit. Motivaci respondentů můžeme rozdělit na hmotnou a nehmotnou. Do nehmotné motivace můžeme zahrnout dobrý pocit z vyplnění dotazníku. Toho se využívá u vyplnění dotazníku blízké osobě (rodina, kamarád) nebo při vyplňování dotazníku pro neziskovou organizaci. Důležité u nehmotné motivace je zdůraznit důležitost názoru a hlavně poděkování na závěr.

Do hmotné motivace patří nějaká forma odměny. Může jít o finanční prostředky, dárek nebo různé benefity. Tato forma motivace je využívána hlavně u komerčních dotazníků. Do hmotné motivace můžeme zahrnout i protislužbu. Mezi systémy využívající hmotné motivace patří například systém *CASI*¹ od agentury *IPOS*. Tato agentura se zabývá výzkumem veřejného mínění v České Republice více než 20 let. Mezi další systémy můžeme počítat *vypnto.cz*², které využívá modelu zpoplatněné propagace dotazníků. Tyto dotazníky jsou rozesílány uživatelům jejich databáze a také jsou dotazníky propagovány na internetu.

Cílem bakalářské práce je vytvořit systém na sdílení dotazníku mezi uživateli, který bude využívat hmotnou motivaci formou protislužby. Po vyplnění několika jiných dotazníků dostává respondent protislužbu (satisfakci). Satisfakce může být ve formě možnosti umístění

¹<https://casicz.ipsos.cz/>

²<https://www.vypnto.cz/>

vlastního dotazníku k vyplnění. Z respondenta se stává zadavatel. Zisk je na obou stranách. Tato služba bude využívat novou experimentální knihovnu *CloudKit* [4]. *CloudKit* knihovnu představila firma *Apple Inc.* na *WDDC 2014*. Hlavní princip knihovny slouží k tomu, aby se vývojáři soustředili na vývoj aplikací a neřešili serverovou část aplikace. Tento systém by měl ušetřit finanční prostředky na provoz vlastní serverové struktury.

Kapitola 2

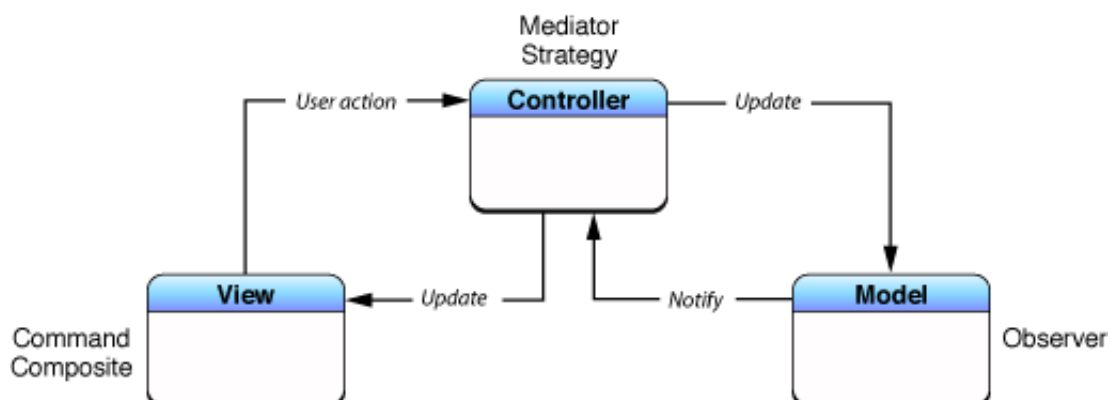
Rozbor tématu

Tato bakalářská práce zkoumá využití nové knihovny *CloudKit*. Tuto knihovnu vyvíjí společnost *Apple Inc.* pro platformu *iOS* a *OS X*. Platforma *iOS* je určena pro mobilní zařízení a využívá knihovnu *Cocoa Touch* pro grafické prostředí. Výsledná práce bude implementována v jazyce Objective-c, ve vývojovém prostředí *Xcode*. Vývojové prostředí obsahuje také emulátor zařízení s operačním systémem *iOS*. Tento emulátor slouží pro jednoduché testování aplikací bez použití zařízení od společnosti *Apple Inc.*.

2.1 Struktura typizované aplikace v iOS

Podle dokumentu user experience[2] na platformě *iOS*, by se většina aplikací měla řídit podle návrhového vzoru *Model View Controller*[3]. Návrhový vzor je rozšířen u všech komponent, frameworků a je zobrazen na Obrázku 2.1. Objekty navržené podle *Model View Controller* bývají více použitelné a jejich rozhraní bývá lépe definováno. Koncept se skládá ze tří základních rolí:

- Role *Model* – zapouzdřuje data a stará se o jejich spravování.
- Role *View* – prezentuje data uživateli.
- Role *Controller* – tvoří bránu mezi *Model* a *View*. Zajišťuje jejich komunikaci.



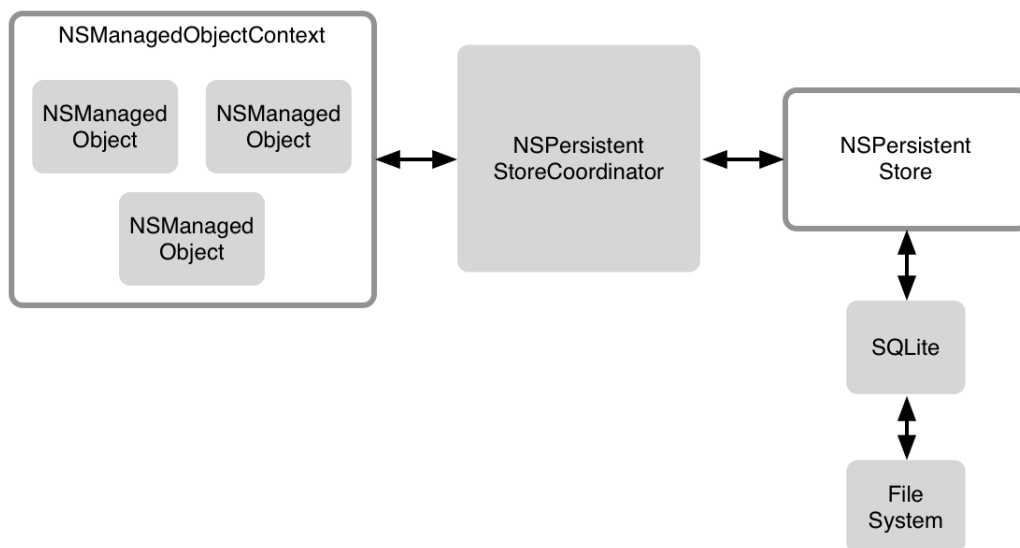
Obrázek 2.1: Model View Controller¹

2.2 Archivace dat

Slouží pro převedení objektů a jejich hodnot do proudu bytů. Tento proud zachovává vztahy mezi objekty a jejichmi hodnotami. Pro vytváření a čtení těchto proudů se využívají kodéry implementované třídou **NSCoder**. Tyto kodéry zasílají dvě zprávy objektu, aby byl zakódován nebo rozkódován. Kodér zasílá zprávu **encodeWithCoder:** pro vytvoření archivu a **initWithCoder:** pro čtení archivu. Tyto zprávy definují **NSCoding** protokol. Pouze objekty, které implementují **NSCoding** protokol, můžou být archivovány.

2.3 Knihovna Core Data

CoreData[9] knihovna slouží k vytváření objektového grafu. Na pozadí používá *SQLite*, ale není to relační databáze ani objektově relační mapování. Je to vyšší úroveň abstrakce, která je zobrazena na Obrázku 2.2.



Obrázek 2.2: Abstrakce Core Data²

NSManagedObject reprezentuje entitu v datovém modelu, která se nachází v **NSManagedObjectContext**. Tento kontext obsahuje objektový graf modelu a stará se o jeho zpravování. **NSPersistentStore** se využívá k prezistentnímu ukládání informací v *OS X* a *iOS*. Mezi těmito dvěma částmi je **NSPersistentStoreCoordinator**, který slouží jako prostředník pro ukládání a načítání modelu.

2.4 Knihovna CloudKit

Knihovna *CloudKit* je postavena nad datovým centrem společnosti *Apple Inc.* nazývaném *iCloud*. Tato knihovna umožňuje strukturované ukládání dat v podobě záznamů. Záznamy jsou tvořeny slovníkem klíč-hodnota. Klíč představuje atribut záznamu a hodnota většínou představuje jednoduchý datový typ například **NSString**, **NSNumber** nebo **NSDate**.

¹Převzato z internetové stránky <https://developer.apple.com/library/ios/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html>

²Převzato z internetové stránky <http://www.objc.io/issue-4/core-data-overview.html>

Hodnota může být nějaký soubor nebo blok dat, ale také vztahy mezi záznamy. Tímto se dá vytvořit komplexní model databáze. CloudKit knihovna zajišťuje kompletní přehled nad přenosem dat a také nad změnami, které proběhly.

V dalších podkapitolách budou uvedeny potřebné části knihovny *CloudKit*, které jsou nutné k implementaci databáze. Mezi tyto části patří například vytváření a stahování záznamů nebo sledování změn na *iCloud* uložišti.

2.4.1 Kontejnér a databáze

Každá aplikace má vlastní kontejnér, který slouží jako lokální uložště iCloud této aplikace. Kontejnér je reprezentován **CKContainer** objektem, který je rozdělen na veřejnou a privátní databázi. Tento kontejnér je reprezentován na Obrázku 2.3. Veřejná databáze je dostupná všem uživatelům a velikost se škáluje s rostoucím počtem uživatelů³. Toto zajišťuje problém s rostoucím počtem uživatelů, problémy s latencí a dostupností služby. Služba je přístupná všem uživatelům, i když nemají aktivní iCloud účet, ale nejsou oprávněni ukládat a modifikovat data. Privátní databáze je přístupná pouze jednomu uživateli a zabírá místo na jeho iCloud účtu. Databázi reprezentuje objekt **CKDatabase**, který zabezpečuje operace nad celou databází s využitím **addOperation:** nebo také operace s jedním záznamem pomocí metod:

- **fetchRecordWithID:completionHandler:** Stáhne požadovaný záznam pomocí jeho **recordID**.
- **saveRecord:completionHandler:** Uloží záznam do databáze.
- **deleteRecordWithID:completionHandler:** Smaže záznam s odpovídajícím **recordID**.

Kontejnér také obsahuje záznam o uživateli, který v základu neobsahuje žádné informace o uživateli a slouží k jedinečné identifikaci. Tento záznam se může využít k uchování informací o uživateli, ale nikdy by neměl obsahovat citlivé informace. Aby tento záznam mohl být nalezitelný dalšími uživateli aplikace, je nutné požádat uživatele o svolení. Pokud je svolení uděleno, aplikace může také získat křestní a rodné jméno.

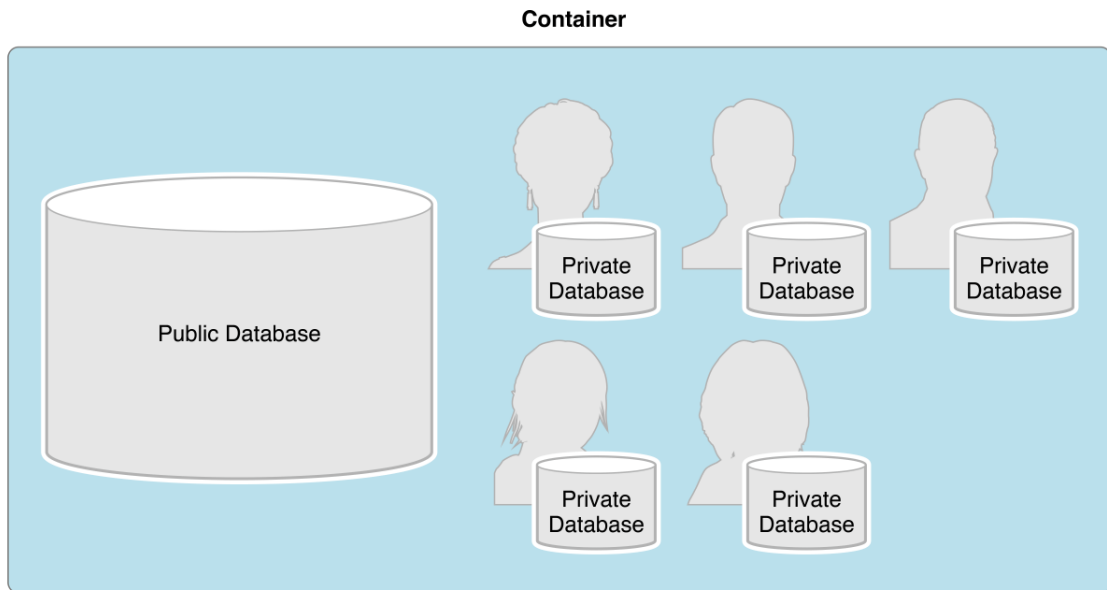
2.4.2 Třída CKRecord

Základní třída, která se využívá pro spravování dat v CloudKitu. Tato třída reprezentuje záznam v datovém modelu. Je to slovník s definicí klíč-hodnota, hodnota v tomto případě je objekt. Pro získávání hodnot objektů slouží metoda **objectForKey:** a pro nastavení hodnoty **setObject:forKey:**. Tato třída má taky vlastnost **recordID**, která je jedinečný klíč, podle kterého je možné záznam vyhledat a upravovat. CKRecord podporuje tyto typy objektů:

- **NSString** – Slouží pro krátké textové řetězce, dlouhé se ukládají jako **CKAsset**.
- **NSNumber** – Ukládání číselných informací s plovoucí čárkou i bez.
- **NSData** – Ukládání bytových informací, pro velké také **CKAsset**.

³<https://developer.apple.com/icloud/documentation/cloudkit-storage/>

⁴Převzato z internetové stránky <https://developer.apple.com/library/ios/documentation/DataManagement/Conceptual/CloudKitQuickStart/EnablingiCloudandConfiguringCloudKit/EnablingiCloudandConfiguringCloudKit.html>



Obrázek 2.3: Kontejnér aplikace⁴

- **NSDate** – Informace o čase a datu.
- **NSArray** – Ukládání polí ostatních objektů ve výčtu.
- **CLLocation** – Obsahuje informace o geografické poloze.
- **CKAsset** – Využívá se pro ukládání diskových dat například: hudba, obrázky, video, dlouhý text.
- **CKReference** – Vytváří vztahy mezi **CKRecord**, obsahuje **recordID** cílového záznamu.

CloudKit může indexovat zvolené typy u záznamů, což urychluje následné vyhledávání pomocí predikátu dotazu. Indexování zabírá místo v databázi a vyžadují nějaký čas pro vytvoření a udržování. Proto je dobré vypnout indexování u atributů, podle kterých se nevyhledává.

2.4.3 Třída CKAsset

Objekt, který reprezentuje velký soubor přiřazený záznamu. Tento objekt se používá k uchovávání externích souborů například: obrázek, video, hudba, text a binární data, které mají více než pár kilobajtů. Soubory nesmějí překročit hranici 250 MB, nefunguje to jako uložisko souborů. Tento záznam je uchován mimo **CKRecord**, ale je s ním svázaný. Pokud chceme odstranit CKAsset odkaz na něj nastavíme na **nil** a CloudKit server je poté sám líně smaže v budoucnu. To znamená, že není možné smazat záznam přímo na serveru.

2.4.4 Třída CKReference

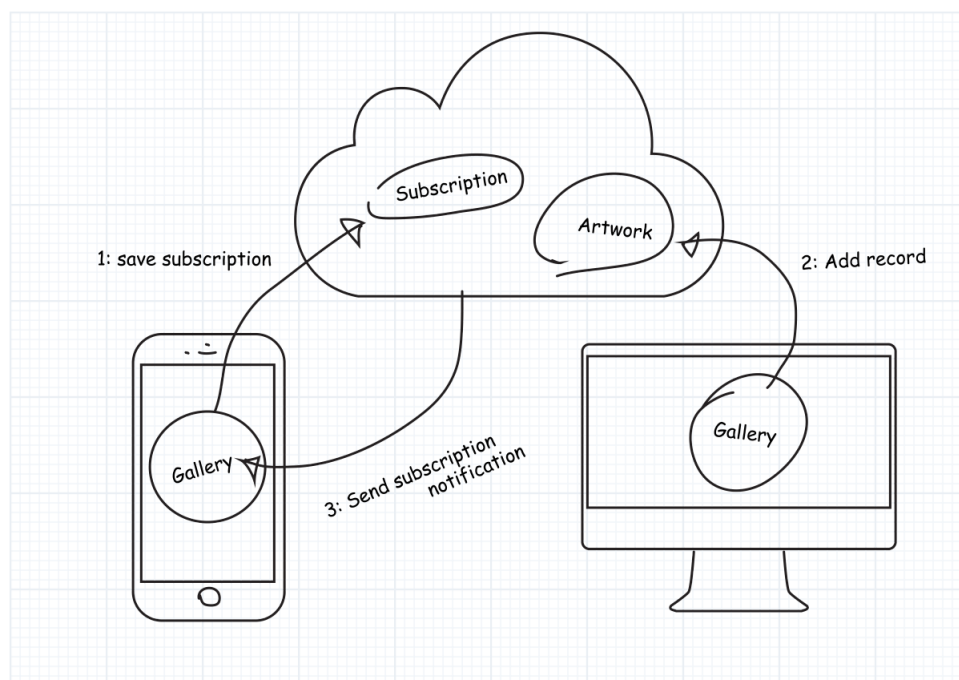
Vytváří vazby 1:N mezi záznamy v databázi. Ukládá informaci **recordID** o jednom cílovém záznamu, se kterým je spojena. Tento vztah má silnější vazbu, která má přiřazenou akci.

CKReference se inicializuje pomocí metody **initWithRecordID:action:**, kde **RecordID** je cílový záznam a **action** je jedna z následujících:

- **CKReferenceActionNone** – neprovádí se žádná akce. Pokud je smazán vlastník, referencované záznamy neprovádí žádnou akci a zůstávají beze změny.
- **CKReferenceActionDeleteSelf** – smazání závislých záznamů. Pokud je smazán vlastník, smažou se všechny referencované záznamy. Smazáním vlastníka se smažou i záznamy, které jsou s ním spojeny. Může nastat také kaskádové mazání všech závislých záznamů.

2.4.5 Třída CKSubscription

Využívá se pro sledování změn na serveru, aby aplikace dostala upozornění. Funguje jako perzistentní dotaz na serveru, který sleduje vytvoření, smazání nebo upravení záznamu. Pokud nastane změna aplikace obdrží oznámení o změně, aby aplikace na něj mohla adekvátně reagovat. Po uložení tohoto objektu odebrání do databáze, kdykoliv nastane změna server rozešle notifikace všem zařízením, které mají tento objekt uložen. Metoda **application:didReceiveRemoteNotification:fetchCompletionHandler:** v delegátu aplikace se stará o zpracování vzdálených notifikací.



Obrázek 2.4: Systém odebrání notifikací ⁵

⁵Převzato z internetové stránky <https://developer.apple.com/library/ios/documentation/DataManagement/Conceptual/CloudKitQuickStart/SubscribingtoRecordChanges/SubscribingtoRecordChanges.html>

2.4.6 Třída CKQuery

Obstarává vyhledávání záznamu v databázi pomocí zvolených kritérií. Objekt **CKQuery** obsahuje vyhledávací parametry, typ entity na prohledání, vyhledávací kritéria (predikát) a řadící parametry. Tento objekt se poté využívá k inicializaci **CKQueryOperation**, která poté vygeneruje naše výsledky. Vyhledávací predikát je omezen pouze na typy: **NSString**, **NSDate**, **NSNumber**, **NSArray**, **CKReference** a **CLLocation**.

2.4.7 Třída CKOperation

Rodičovská třída pro všechny operace, které pracují s databází *CloudKit*. Tyto operace zahrnují komunikaci s *iCloud* serverem pro zasílání a přijímání dat. Operace pracují v asynchronním režimu a proto je nutné respektovat dobu zpracování. Tato doba zpracování se může lišit v důsledku kvality připojení a zátěže služby *CloudKit*. Průběh operací se zpracovává pomocí bloků, které musejí být definovány před začátkem vykonávání operace. Operace jsou poté zaslány do fronty na zpracování.

2.4.8 Třída CKFetchRecordsOperation

Operace, která slouží k načtení **CKRecord** objektů, od kterých známe už jejich **recordID**. V průběhu operace objekt oznamuje stav operace pomocí několika rozdílných bloků, které se můžou používat pro zpracování operací. Mezi tyto bloky patří:

- **perRecordProgressBlock** - Tento blok slouží pro poskytování zpětné vazby uživateli a udává průběh zpracování od 0.0 do 1.0, kde 1.0 znamená, že jsou všechny záznamy staženy.
- **perRecordCompletionBlock** - Tento blok se vykoná pro každý **recordID** v naší operaci a slouží ke zpracování výsledků.
- **fetchRecordsCompletionBlock** - Vykoná se pouze jednou a to na konci zpracování, obsahuje informace o zpracování operace a oznamuje chyby, pokud nějaké nastaly.

2.4.9 Třída CKQueryOperation

Pro vyhledávání záznamů s neznámým **recordID** slouží **CKQueryOperation**. Operace se provádí proti databázi, pro vyhledávání využívá objekt **CKQuery**, který určuje vyhledávací kritéria. Volitelně je možné nastavit **resultsLimit**, tento limit určuje počet stažených záznamů. V základním nastavení server **CloudKit** podle odezvy klienta a zatíženosti vrátí počet záznamů, který je pro něho nejvíce optimální. Také je možné v této operaci použít **CKQueryCursor**, který označuje ukončující místo operace. Toto místo se může použít jako startující bod další operace, pro stažení další dávky záznamů. Bloky pro zpracování jsou v téhle operaci pouze dva, jelikož předem nevíme kolik bude výsledků a nemůžeme tedy sledovat průběh.

Kapitola 3

Návrh dotazníkového systému

Kapitola se zabývá analýzou požadavků a popisuje návrh aplikace a sociální sítě **socialQ**. Tato síť bude založena na nové experimentální knihovně *CloudKit*. V síti budou dvě role uživatelů, jmenovitě zadavatelé a respondenti. Zadavatelé musejí mít možnost vytvářet dotazníky cílené na určitou skupinu respondentů. Cílová skupina bude vybírána podle kategorie dotazníku. Zadavatelé by měli mít možnost exportovat nebo sdílet výsledky dotazníků. Systém by měl obsahovat ochranu, která by zamezovala vytváření velkého počtu dotazníků jedním zadavatelem. Systém by měl být plně automatizovaný a schopný provozu bez dalších zásahů.

Požadavky, které jsem stanovil pro svůj dotazníkový systém:

- Vytvářet nové dotazníky.
- Odpovídat na dotazníky vytvořené ostatními uživateli sítě.
- Informovat respondenty o dostupnosti nových dotazníků.
- Publikovat výsledky dotazníků mimo sociální síť **socialQ**.
- Upravovat profily uživatelů v systému.
- Sledovat aktivitu sociální sítě.
- Chránit systém proti zahlcení dotazníky.

Dále popisovaný návrh systému reflektuje ho požadavky. Je tvořen:

- Koncept pravidel a fungování sociální sítě **socialQ**.
- Koncept implementace dotazníků a databázového uložště.
- Koncept aplikace a jejího uživatelského rozhraní.

3.1 Koncept sociální sítě socialQ

Sociální síť by měla sloužit jako místo uživatelů pro odpovídání a publikaci dotazníků. Tato síť by měla motivovat uživatele pro vyplnění dotazníků. Motivace bude založena na odměně ve formě možnosti vytvořit si vlastní dotazník. Tyto vytvořené dotazníky jsou poté možné publikovat přímo na sociální síti. Výsledná součinnost uživatelů by měla zaručit dostatečný počet respondentů.

3.1.1 Princip tokenů v systému

Tokeny v systému slouží jako virtuální měna, která má zabránit zahlcování systému dotazníky. Aby uživatel směl publikovat dotazník, bude potřeba zaplatit určitou částku tokenů za tento dotazník. Poplatek za publikaci dotazníku (3.1) se bude odvíjet od počtu aktivních dotazníků v systému.

$$cenaPublikace = (pocetDostupnychDotazniku + pocetDotaznikuUzivatele) * 1.5 \quad (3.1)$$

Po zaplacení poplatku se dotazník nahraje na sociální síť a respondenti jej můžou začít vyplňovat. Za vyplnění dotazníku obdrží respondent právě jeden token. Získané tokeny se přičítají respondentovi na účet. Po získání potřebného počtu tokenů je umožněno také respondentovi publikovat vlastní dotazník. Tento systém by měl umožnit měnit cenu dotazníku v reálném čase.

Pokud dojdou všem uživatelům tokeny a nebude žádný aktivní dotazník, dochází k takzvané **nulové** situaci. Tato situace nastává podle vzorce (3.1), pokud nejsou aktivní žádné dotazníky. Při **nulové** situaci je možné publikovat dotazníky zdarma. Pokud někteří uživatelé objeví tuhle situaci, je možné jí využít i v budoucnu a předejít drahému publikování dotazníku.

Princip tokenů může také sloužit jako druh motivace. Uživatelé mají přehled o počtu tokenů v systému a také o počtu získaných tokenů. Z tohoto přehledu pak může uživatel usoudit, že patří mezi nejvíce aktivní uživatele.

3.1.2 Fungování uživatelů v systému

Každý uživatel sítě **socialQ** může vystupovat v každém okamžiku v jedné z následujících rolí:

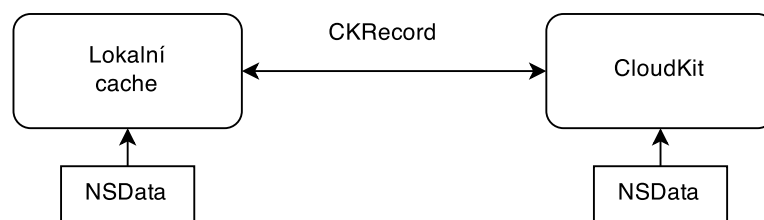
- Zadavatel - Vytváří dotazníky a publikuje jejich výsledky.
- Respondent - Vyplňuje dotazníkové průzkumy a může se stát zadavatelem.

Při prvním vstupu do systému se každý uživatel stává respondentem. Vstupem do systému získá také svůj první token. Tento token slouží jako základ jeho účtu. Při každém zisku nového tokenu se zvedá celkový počet tokenů uživatele. Respondent má také právo upravovat svůj uživatelský profil. Tento profil určuje, které dotazníky se mu zobrazují. Mezi jeho další možnosti patří informování o nových dotaznících v systému. Respondent si může také vytvářet dotazníky, ale nemůže je publikovat dokud nemá potřebný počet tokenů.

Po zisku potřebných tokenů může respondent publikovat dotazník a stává se zadavatelem. Při procesu publikace jsou odebrány získané tokeny a dotazník je nahrán na sociální síť *socialQ*. Zadavatel má všechny možnosti jako respondent a také navíc může prohlížet a publikovat výsledky dotazníků.

3.2 Koncept uložení dat

Data v demonstrační aplikaci jsou uloženy třemi způsoby. První způsob je uložení dotazníku ve formě *NSData*, který je popsán v kapitole 5.1.2. Druhý způsob uložení dat dotazníků a jejich výsledků v databázi *SQLite*, nad kterou je vytvořena abstrakce v podobě *CoreData*. Poslední způsob využívá knihovnu *CloudKit* pro komunikaci s datovým uložištěm *iCloud*. Uložiště *iCloud* bude obsahovat všechny informace o sociální síti a publikovaných dotaznících. Přenos dat je zobrazen na Obrázku 3.1.



Obrázek 3.1: Přenos dat v systému

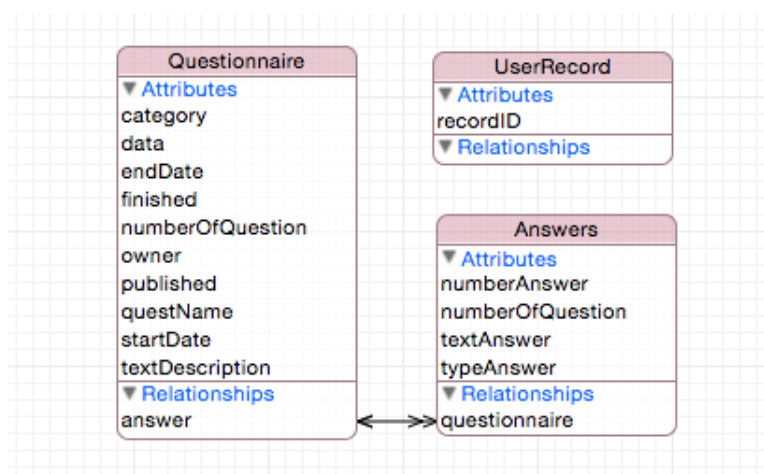
3.2.1 Dotazník

Dotazník bude tvořen třídou **QuestionnaireModel**, tato třída bude obsahovat pole otázek. Pole otázek se bude skládat ze třídy **Question** a jejich podtříd. Obě třídy budou implementovat *NSCoding* protokol, zajišťující archivaci. Dotazníky mohou být modulární, u kterých předem neznáme jejich strukturu. Tento návrh nám umožní ukládat dotazníky jako objektovou paměť. Uložení v *CloudKit* databázi bude zajištěno jako *NSData*. Mezi další výhody tohoto návrhu patří:

- Nezávislost na datovém modelu.
- Možná modifikace struktury dotazníku.
- Vylepšení otázek pomocí obrázků, zvuků nebo videí.

3.2.2 Databáze systému

Dotazníkový systém bude využívat abstrakci *Core Data*, pro ukládání dat a výsledků dotazníků. *Core Data* bude také navíc ukládat informace o uživateli **recordID**. Tato informace slouží pro stahování informací o uživateli. Datový model je zobrazen na Obrázku 3.2. Lokální databáze bude sloužit výhradně jako uložisko pro uživatele, pokud nemá dostupné připojení k internetu. Umožní vytvářet a upravovat dotazníky na zařízení.

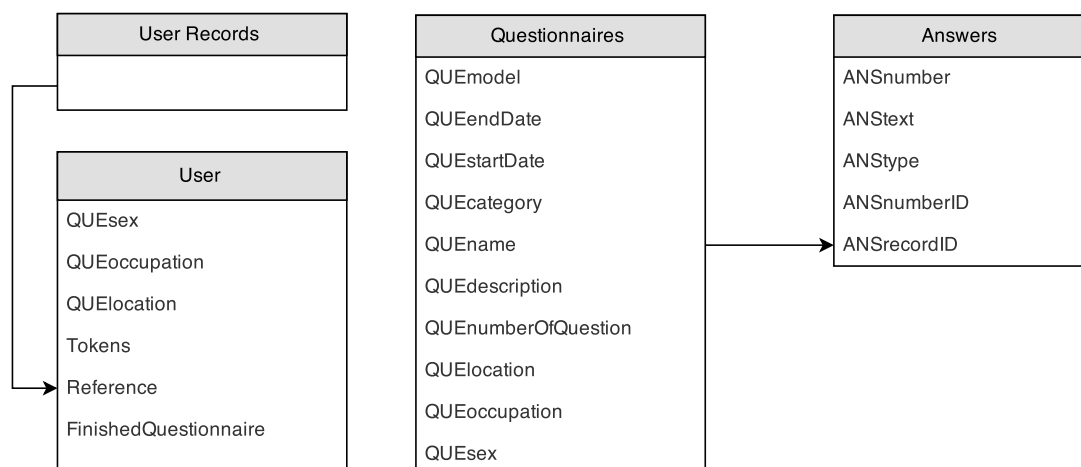


Obrázek 3.2: Datový model na lokální uložení.

Datový model *CoreData* je složen z entit **Questionnaire** a **Answers**. Entita **Questionnaire** slouží k uložení meta informací dotazníku a také obsahuje datový model dotazníku z kapitoly 5.1.2. Tento model je zde uložen jako *NSData* v atributu **data**.

Výsledky dotazníků jsou uloženy v entitě **Answers**. Pro každý výsledek je vytvořen samostatný záznam. Tyto záznamy jsou spojeny s určitým dotazníkem pomocí vztahu **questionnaire**. Tento vztah také zaručuje smazání všech odpovědí po smazání svázaného dotazníku.

Služba *CloudKit* bude zajišťovat vzdálenou databázi. Tato databáze bude sloužit jako uložisko všech publikovaných dotazníků a odpovědí. Bude také obsahovat všechny informace o uživateli. Datový model vzdálené databáze je zobrazen na Obrázku 3.3.



Obrázek 3.3: Datový model na *iCloud* uložišti.

User Record nepodporuje *CKQueryOperation* a proto bylo nutné vytvořit entitu **User**. Tato entita je spojena pomocí *CKReference* s **User Record**. Entita **User** obsahuje informace o uživateli:

- **QUEsex** - Udává pohlaví uživatele.
- **QUEoccupation** - Určuje zařazení ve společnosti například student, pracující nebo důchodce.
- **QUElocation** - Obsahuje informaci o poloze uživatele.
- **Tokens** - Udává aktuální počet tokenů získaných uživatelem.
- **FinishedQuestionnaire** - Obsahuje celkový počet vyplněných dotazníků.

Záznam dotazníku se liší oproti *CoreData*. Tato entita obsahuje navíc meta informace dotazníku. Tyto informace určují kategorii dotazníku a slouží pro vyhledávání dostupných dotazníků pro uživatele. Pro stahování dotazníků slouží metoda **fetchQuestionnairesWithCompletionHandler**. Implementace této metody je popsána v kapitole 4.1. Datový model dotazníku je zde uložen jako *NSData*. Entita **Answers** slouží také jako uložení výsledků k dotazníku jako v modelu *Core Data*.

3.3 Koncept aplikace

Výsledná aplikace bude mimo knihovny *CloudKit* využívat také knihovnu *TwitterKit*[10] od společnosti *Twitter Inc.*. Mezi další knihovny bude patřit *DropboxSDK*[6] od společnosti *Dropbox Inc.*. Obě knihovny budou používány k publikaci výsledků mimo dotazníkový systém popsáný v kapitole 3.3.2.

3.3.1 Uživatelské rozhraní

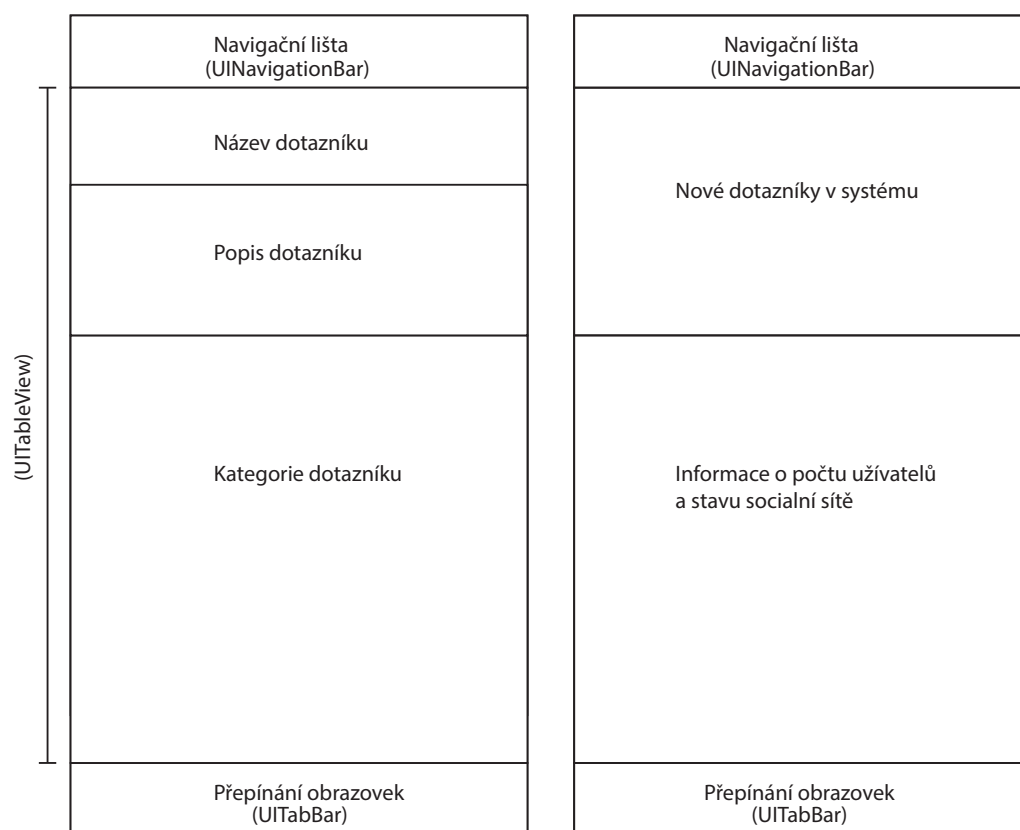
Při návrhu uživatelského rozhraní jsem se snažil dodržovat pokyny od společnosti **Apple Inc.**[5]. Vytvořit jednoduché a intuitivní rozhraní skládající se ze 4 obrazovek, které budou sloužit pro rozdílné účely.

- Obrazovka nástěnky - Bude zobrazovat základní informace o systému, počet uživatelů, počet vyplněných dotazníků, počet tokenů v oběhu a také nejnovější dotazníky.
- Obrazovka dotazníků - Umožňuje uživateli prohlížet dotazníky dostupné v systému a také dotazníky vyplňovat.
- Obrazovka vytváření dotazníků - Využívá se pro vytváření nových dotazníků a prohlížení výsledků publikovaných dotazníků od uživatele.
- Obrazovka profilu - Slouží k nastavení informací o uživateli a také jestli si uživatel přeje dostávat notifikace o nových dotaznících.

Na Obrázku 3.4 máme návrh obrazovky vytváření dotazníků a nástěnky. Obě obrazovky jsou tvořené ze záhlaví *UINavigationController*. Tělo bude tvořené *UITableView*. U vytváření dotazníku toto tělo bude obsahovat název dotazníku, popis dotazníků a zařazení dotazníku pomocí kategorií. Obrazovka nástěnky bude obsahovat seznam nejnovějších dotazníků v systému a také údaje o stavu tokenů v systému. Zápatí obou obrazovek tvoří *UITabBar*, který bude sloužit pro přepínání mezi okny obrazovek.

3.3.2 Publikace výsledků

Publikace výsledků mimo dotazníkový systém bude využívat uložiště *Dropbox*, které bude sloužit pouze jako uložiště mimo ekosystém *Apple*. Dotazník je exportován jako PDF dokument. Tento dokument vytváří funkce **generatePdf** z **TableView** výsledků dotazníku. Exportované dotazníky jsou ukládány do uložiště *Dropbox*. Výsledky je možné prohlížet přímo na službě *Dropbox*. Další možnost je publikovat výsledky na sociální síť *Twitter* přímo z aplikace. Tato metoda bude využívat *Dropbox* odkazu, který bude obsažen v novém příspěvku.



Obrázek 3.4: Vytváření dotazníků a nástěnka sociální sítě

Kapitola 4

Implementace systému

Následující kapitola se zaměřuje na implementaci sociální sítě **socialQ**. Komunikaci se službou *CloudKit* obstarává třída *CloudKitHandler*. Pro zpracování výsledků je definován typ **CloudKitCompletionHandler**. Tento typ obsahuje *NSArray* pro uložení výsledků operace a *NSError* pro zvládání chybových situací.

4.1 Filtrování aktuálních dotazníků

Pro filtrování aktuálně dostupných dotazníků k vyplnění slouží metoda **fetchQuestionnairesWithCompletionHandler:handler**. Argument metody je **CloudKitCompletionHandler**, který zpracovává výsledky. Volání metody se vyvolá další metoda **userIdFromCloudWithCompletionHandler:handler**, která stáhne meta informace o uživateli.

Meta informace jsou ve formě *CKRecord*, ze kterého získáme **QUEsex**, **QUEoccupation** a **QUElocation**. Poté jsou získány všechny **recordID** vyplněných dotazníků z *Core Data* pomocí **activeRecordID**. Ze všech potřebných informací je vytvořen *NSPredicate* ve formátu:

```
[NSPredicate predicateWithFormat:@"NOT(QUEname IN %@) AND
    ((QUEstartDate <= %@) AND (QUEendDate >= %@)) AND
    (QUEsex IN %@) AND (QUEoccupation IN %@) AND
    (QUElocation IN %@)",
    questionnaireName, currentDate, currentDate,
    sexArray, ocuArray, locArray];
```

Pomocí predikátu a typu **Questionnaires** entity se vytvoří *CKQuery*, které tvoří základ *CKQueryOperation*. Blok pro zpracování *recordFetchedBlock* přidá model do výsledného pole výsledků. Po zpracování všech záznamů jsou všechny výsledky přidány do fronty na zpracování:

```
dispatch_async(dispatch_get_main_queue(), ^{
    handler([NSArray arrayWithArray:tmp], error);
});
```

Pro začátek vykonávání operace slouží metoda:

```
[[self publicCloudDatabase] addOperation:queryOperation];
```

4.2 Stáhnutí a vyhodnocení výsledků

Pro stahování výsledku slouží metoda **fetchAnswersWithRecordId:recordId completionHandler:handler**. Vyhledávání může začít dvěma způsoby od definovaného *CKQueryCursor*, který byl koncem minulé operace nebo pomocí nového predikátu. Predikát pro vyhledávání výsledků dotazníků je velice jednoduchý a závisí pouze na **recordID** dotazníku. Predikát má formu:

```
[NSPredicate predicateWithFormat:@"ANSrecordID == %@", recordId];
```

Výsledky, které vyhovují vyhledání, jsou přidány v bloku **recordFetchedBlock** do pole výsledků. Po ukončení operace se provádí kontrola na pozici kurzoru vyhledávání. Kontrola konce kurzoru se provádí v bloku **queryCompletionBlock** a její forma je následující:

```
if(cursor != nil) {
    self.cursor = cursor;
    self.array = array;
    dispatch_async(dispatch_get_main_queue(), ^{
        [self fetchAnswersWithRecordId:nil completionHandler:handler];
    });
} else {
    self.cursor = nil;
    dispatch_async(dispatch_get_main_queue(), ^{
        handler(array, error);
    });
}
```

Po ukončení všech operací jsou výsledky ve formě *CKRecord*, a proto je nutné je přemapovat na výsledky dotazníku. Za použití metody **mapResultArray:** se převádí výsledky z *CKRecord* na potomky třídy **Question**. Každý potomek této třídy má vlastní modifikovanou metodu, která určuje zpracování *CKRecord*. Mapování výsledků:

```
for (CKRecord *record in array) {
    id tmp = [self.questionArray questionAtIndex:index];
    NSInteger index = [[record valueForKey:ANSnumberID] intValue];
    [tmp addQuestionAnswerWithRecord:record];
}
```

Kapitola 5

Testování

Následující kapitola se zabývá testováním aplikace **socialQ**, která slouží pro publikování a odpovídání dotazníkových průzkumů. Cílem testování je zjištění fungování:

- Konceptu sociální sítě **socialQ** a fungování tokenového systému.
- Výkonnostní testy služby *CloudKit*.

Tato kapitola obsahuje také ukázky aplikace s krátkým komentářem.

5.1 Testování sociální sítě socialQ

Testování má za úkol ověřit fungování sociální sítě **socialQ**. Toto testování probíhalo po dobu 7 dní a zúčastnili se ho 5 uživatelů. Uživatelé měli za úkol vyzkoušet:

- Vytvořit alespoň jeden dotazník, který obsahuje minimálně 10 otázek.
- Vyplnit co největší počet dotazníků.
- Publikovat dotazník na sociální síti **socialQ**.
- Exportovat výsledky dotazníku na cloudové uložisko *Dropbox*.
- Sdílet dotazník na sociální síti *Twitter*.

Po ukončení testování byly všechny výsledky analyzovány a shrnutí je v kapitole 5.1.1. Mezi výsledky testování také patří dotazník, na který odpovědělo nejvíce uživatelů. Tento dotazník je analyzován v kapitole 5.1.2.

5.1.1 Výsledky testování

Po prozkoumání výsledků v tabulce 5.1, můžeme konstatovat, že všichni uživatelé vytvořili a publikovali alespoň jeden dotazník. Nejaktivnější byl uživatel D, který vyplnil 5 dotazníků a publikoval 2 dotazníky. Tento uživatel nejspíše využil takzvané **nulové** situace zmíněné v kapitole 3.1.1. Můžeme to usoudit z počtu tokenů, které mu zůstaly a také z počtu publikovaných dotazníků. Z tabulky můžeme vyčíst, že uživatelé celkově vyplnili 18 dotazníků a publikovali 7 dotazníků.

Publikace výsledků mimo sociální síť je zobrazena v tabulce 5.2. Všichni uživatelé až na jednoho publikovali své výsledky na *Dropbox* a *Twitter*. Tento uživatel uvedl, že nebyl ochotný se registrovat do služeb *Dropbox* a *Twitter*.

Tester	Vyplněné dotazníky	Publikovaných dotazníků	Tokeny na konci testu
Uživatel A	4	2	0
Uživatel B	2	1	0
Uživatel C	4	1	1
Uživatel D	5	2	2
Uživatel E	3	1	1
Celkem	18	7	4

Tabulka 5.1: Stav systému po ukončení testování

Tester	Export <i>Dropbox</i>	Sdílení na sociální síti <i>Twitter</i>
Uživatel A	Ano	Ano
Uživatel B	Ne	Ne
Uživatel C	Ano	Ano
Uživatel D	Ano	Ano
Uživatel E	Ano	Ano

Tabulka 5.2: Publikace mimo sociální síť **socialQ**

5.1.2 Analýza dotazníku

Nejvíce odpovědi obdržel dotazník **Dovolena**, na který odpověděli 4 uživatelé. Tento dotazník se zajímal o styl a průběh trávení dovolené. Nejčastěji uživatelé jezdí na dovolenou relaxovat. Dovolenu tráví buď s přáteli nebo rodinou. Dopravu volí leteckou nebo automobilem. Délka dovolené se u respondentů pohybuje mezi 7-14 dny. Většina respondentů preferuje pobyt v hotelovém zařízení se stravováním all inclusive. Nejčastěji vyhledávají dovolenou v katalogu cestovních kanceláří. Výsledky dotazníku je možné si prohlédnout v příloze A na CD.

5.2 Výkonnostní testy služby CloudKit

Služba *CloudKit* patří mezi nové technologie. Využívá ji malá skupina uživatelů, proto informací o zatížení není mnoho. Jediné informace jsou o množství přenesených dat, které závisí na počtu uživatelů aplikace. Tato informace byla již zmíněna v kapitole 2.4.1. Informace o počtu přenesených záznamů, limitů a omezení nejsou zmíněny ani v dokumentaci knihovny[1]. Proto bylo potřeba tyto informace zjistit za pomoci experimentu. Provedení experimentů je uvedeno v následující kapitole.

5.2.1 Příprava testů

Pro simulování velkého počtu odpovědních dotazníků je potřeba zjistit, kolik jde uložit záznamů v jedné operaci. Poté je nutné zjistit, kolik může být vytvořeno operací pro nahrání testovacích odpovědí. Tyto odpovědi je nutné stáhnout, zkontrolovat počet a jestli sedí nebo ne.

Při testování maximalního počtu vytvořených *CKRecord* v jedné operaci *CKModifyRecordsOperation* jsem začal od jednoho záznamu a zvyšoval počet záznamu po stovce. Po dosažení 401 záznamů v jedné operaci nastala chyba:

"Invalid Arguments" (12/1020);

```
"Your request contains more than the maximum number of items  
in a single request (400)"
```

Proto maximální počet záznamů v jedné operaci nesmí překročit 400.

U maximálního počtu operací jsem postupoval obdobně. Začínal jsem pouze u jedné operace a zvyšoval počet pouze o jednu operaci. Po dosažení 11 operací nastala chyba:

```
"Request Rate Limited" (7/2008);  
"This operation has been rate limited"; Retry after 32.0 seconds>
```

Vyvodil jsme, že počet operací musí mít mezi sebou nějaký timeout nebo nepřidat do fronty pro zpracování více než 10 operací současně. Hodnota **Retry after** se měnila v závislosti na lokálním čase. Při jednom testu dosáhla i hodnoty 1302 sekund, což je více než 17 minut pro možnost vytvoření nové operace. V průběhu experimentování jsem vyzoroval, že odezvy služby *CloudKit* se mění i podle aktuálního času. Vyšší odezvy jsou nejspíše způsobeny větším využitím uložiště *iCloud* v určitou dobu.

Z analyzovaných informací jsem vytvořil sadu testů. Tato sada testů bude pracovat s jedním testovacím dotazníkem, který bude mít 15 otázek. Pro počet *CKRecord* v jedné operaci jsem zvolil 300, jelikož je dostatečně velký počet odpovědí a nepřesahuje hodnotu 400. Rozhodl jsem se udělat 3 skupiny testů:

- Test A, který obsahuje pouze jednu operaci s 300 záznamy.
- Test B, který obsahuje pět operací s 1500 záznamy.
- Test C, který obsahuje deset operací s 3000 záznamy.

Tyto testy jsem třikrát provedl za sebou a výsledné časy délky operací nahrávání jsem zprůměroval. Testy jsem prováděl každé dvě hodiny od 8:00 do 24:00. Poté jsme testoval i délku načítání všech výsledků testovaného dotazníku.

5.3 Výsledky testů

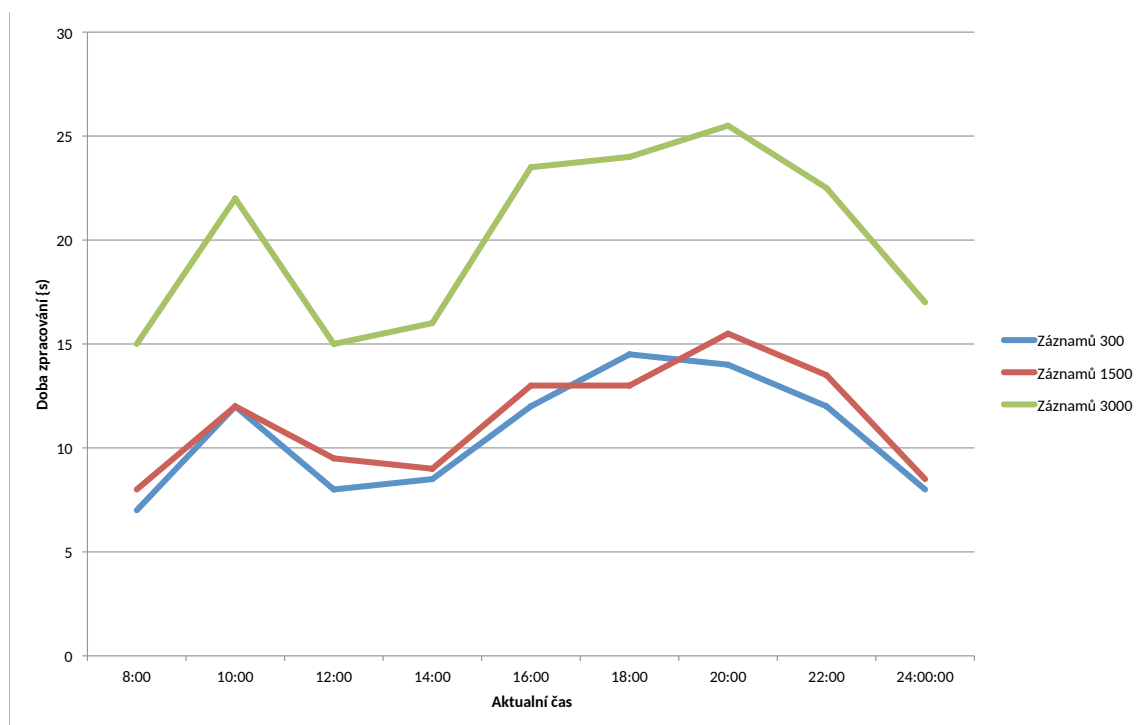
Graf na Obrázku 5.1 popisuje průběh nahrávání odpovědí pro testovací dotazník. Jak z grafu můžeme vyčíst, délka nahrávání závisí na aktuální době. Je to způsobeno nejspíše špičkou používání datového uložiště *iCloud*. Tato špička se dostavuje v dopoledních hodinách a poté po 16:00 odpoledne.

Provedené testy také poukazují, že doba přenosu 300 a 1500 záznamů se moc neliší. Je to způsobeno počtem vláken, které můžou souběžně zpracovávat více požadavků na straně klienta.

Při stahování výsledků jsem narazil na problém, že se nepodařilo stáhnout všechny záznamy. V průměru se stáhlo 600 záznamů, které poskytla služba *CloudKit*. V dokumentaci jsme se dočetl, že *CloudKit* vrací počet záznamů, které jsou výhodné vůči odezvě mezi serverem a klientem. Nebo se naskytuje možnost špatné implementace metody pro stahování odpovědí.

5.4 Ukázky z aplikace

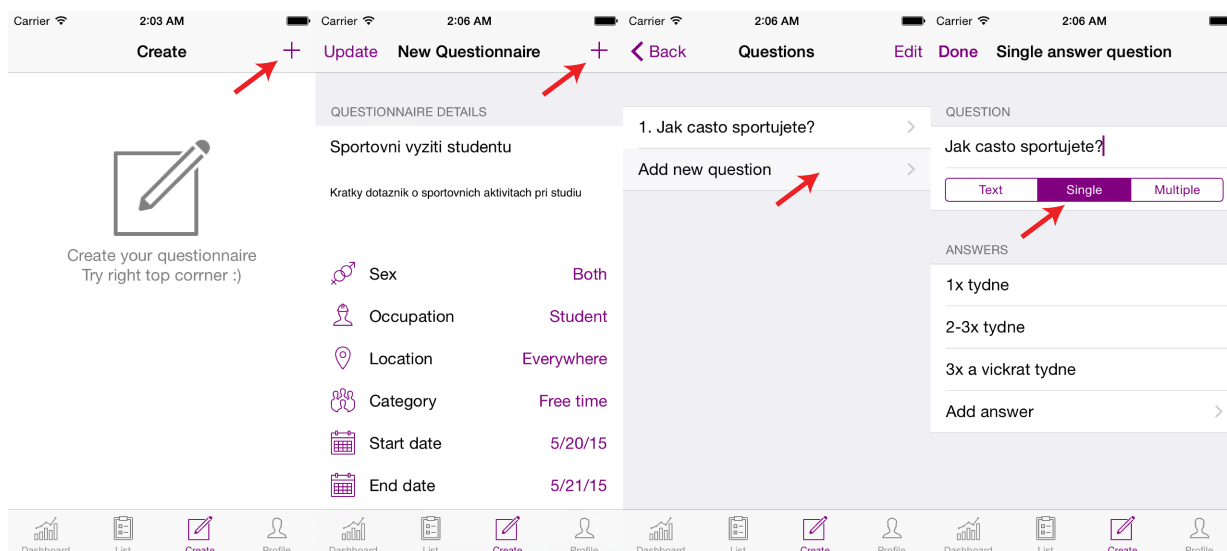
Vytvoření nového dotazníku probíhá v záložce *Create*. V pravém horním rohu najdeme tlačítko **Add (+)**, kterým vytvoříme nový dotazník. Kategorie dotazníku se vybírají pomocí



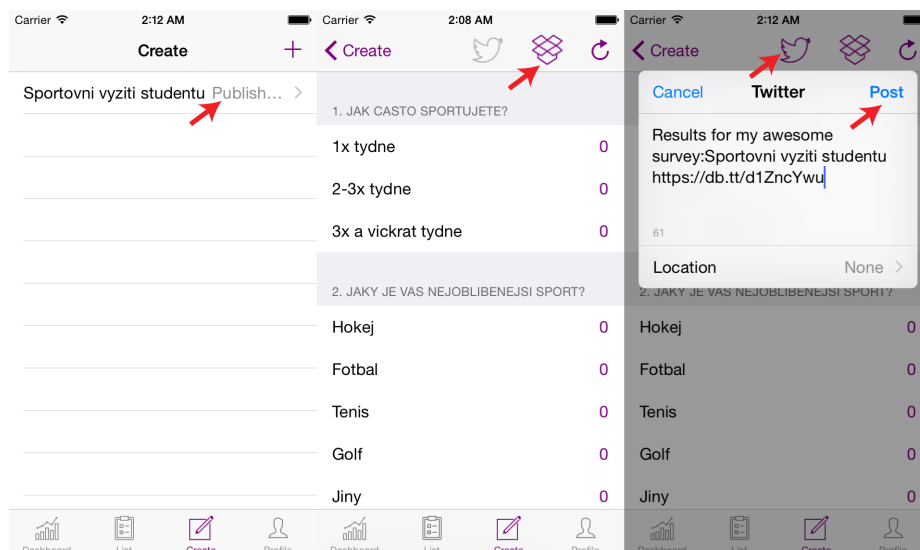
Obrázek 5.1: Doba trvání přenosu pomocí CloudKit.

pickrů a pro přidání nových otázek slouží tlačítko **Add (+)**. V seznamu otázek můžeme přidávat nebo také mazat otázky. Po přidání otázky se nám zobrazí nová nabídka. V této nabídce pomocí *Segmented control* můžeme zvolit typ otázky a případně přidat odpovědi. Postup je zobrazen na Obrázku 5.2

Publikace dotazníku probíhá také v záložce *Create*. Po zvolení publikovaného dotazníku je možnost exportovat dotazník na uložiště *Dropbox*. Pokud uživatel není přihlášený zobrazí se mu výzva k přihlášení. Po úspěšném exportování dotazníku na uložiště *Dropbox* je možné publikovat pomocí tlačítka **Twitter**. Zobrazí se modální okno s novým příspěvkem. V tomto příspěvku je uveden odkaz na publikovaný dotazník a je možné editovat příspěvek. Po stisknutí tlačítka **Post** se nahraje příspěvek na sociální síť *Twitter*. Celý postup je zobrazen na Obrázku 5.3



Obrázek 5.2: Vytváření dotazníků a nástěnka sociální sítě



Obrázek 5.3: Vytváření dotazníků a nástěnka sociální sítě

Kapitola 6

Závěr

Cílem předložené bakalářské práce bylo navrhnout a implementovat sociální síť pro komunitu zadavatelů a respondentů dotazníkových průzkumů. Sociální síť *socialQ* byla implementována pomocí experimentální knihovny *CloudKit* od společnosti *Apple Inc.*. Funkce aplikace byla ověřena na skupině uživatelů, kteří se zúčastnili testování v délce sedmi dnů. Testování ověřilo funkčnost tokenového systému a také možnosti vytváření a publikování dotazníků. Aplikace je propojena se službami Dropbox a Twitter, což umožňuje publikovat dotazníky mimo sociální síť *socialQ*.

Mezi další možnosti rozšíření aplikace patří přidání dalších sociálních sítí pro publikování dotazníků, které by přivedlo větší počet potencionálních respondentů. S větším počtem uživatelů by se mohla rozšířit struktura dotazníku, která by obsahovala obrázky, grafy a další multimediální prvky.

Z osobních zkušeností, které byly nabyty při zátěžovém testování služby *CloudKit*, můžeme usoudit, že služba prochází ještě stádiem vývoje. Nasvědčuje tomu také nedostatek odborných podkladů. Komunita okolo služby *CloudKit* je malá a určitě by si zasloužila větší pozornost.

Literatura

- [1] Apple, I.: CloudKit Framework Reference [online].
https://developer.apple.com/library/ios/documentation/CloudKit/Reference/CloudKit_Framework_Reference/, [cit. 2015-05-06].
- [2] Apple, I.: Concepts in Objective-C Programming [online].
<https://developer.apple.com/library/ios/documentation/General/Conceptual/CocoaEncyclopedia/Introduction/Introduction.html>, [cit. 2015-05-06].
- [3] Apple, I.: Concepts in Objective-C Programming [online]. <https://developer.apple.com/library/ios/documentation/General/Conceptual/CocoaEncyclopedia/Model-View-Controller/Model-View-Controller.html>, [cit. 2015-05-06].
- [4] Apple, I.: Designing for CloudKit [online]. <https://developer.apple.com/library/ios/documentation/General/Conceptual/iCloudDesignGuide/DesigningforCloudKit/DesigningforCloudKit.html>, [cit. 2015-05-06].
- [5] Apple, I.: Designing for iOS [online]. <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/>, [cit. 2015-05-06].
- [6] Dropbox, I.: <https://www.dropbox.com/developers/core/start/ios>, [cit. 2015-05-06].
- [7] Gavora, P.: *Úvod do pedagogického výzkumu*. Brno: Paido, 2010, ISBN 978-80-7315-185-0.
- [8] Hendl, J.: *Kvalitativní výzkum : základní teorie, metody a aplikace*. Praha: Portál, 2012, ISBN 978-80-262-0219-6.
- [9] Neuburg, M.: *Programming iOS 8*. Sebastopol, CA: O'Reilly Media, 2014, ISBN 978-1491908730.
- [10] Twitter, I.: <https://dev.twitter.com/twitter-kit/ios>, [cit. 2015-05-06].

Příloha A

Obsah CD

- /src/ – Zdrojové kódy demonstrační aplikace socialQ
- /latex/ – Zdrojové kódy bakalářské práce
- /thesis.pdf – Text bakalářské práce
- /dotaznik/ – Exportovaný dotazník ve formátu pdf