

Mendelova univerzita v Brně  
Provozně ekonomická fakulta

---

# **Mobilná aplikácia pre podporu organizácie konferencií**

**Bakalárska práca**

Vedúci práce:  
Ing. Dita Dlabolová

Matej Molnár

Brno 2016



Touto cestou by som chcel poďakovať vedúcej práce Ing. Dite Dlabolovej za cenné rady, usmerňovanie a ochotu počas písania. Zároveň by som rád poďakoval svojim rodičom za podporu pri štúdiu.



### **Čestné prohlášení**

Prohlašuji, že jsem tuto práci: **Mobilná aplikácia pre podporu organizácie konferencií**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

v Brně dne 22. mája 2016

.....



**Abstract**

Molnár M. Mobile application for supporting the organization of conferences. Bachelor thesis. Brno: Mendel university, 2016.

The bachelor thesis discusses the design and implementation of a mobile application for supporting the organization of conferences for the iOS platform developed in the Xcode environment. The purpose of the application is to replace the printed version of the conference brochure and to provide attendees with all the important information. The thesis describes the development process for the iOS platform. It also includes a subjective evaluation of the products already in existence offering similar services.

**Keywords** — iOS, Xcode, conference, harmonogram of agenda

**Abstrakt**

Molnár M. Mobilná aplikácia pre podporu organizácie konferencií. Bakalárska práca. Brno: Mendelova univerzita, 2016.

Bakalárska práca sa zaoberá návrhom a implementáciou mobilnej aplikácie pre podporu organizácie konferencií pre platformu iOS vo vývojovom prostredí Xcode. Účelom aplikácie je nahradiť brožúru konfrence vo vytlačenej podobe a sprostredkovať návštevníkom všetky podstatné informácie. V práci je opísaná problematika vývoja pre platformu iOS. Súčasťou práce je aj subjektívne zhodnotenie existujúcich produktov ponúkajúcich podobné služby.

**Kľúčové slová** — iOS, Xcode, konferencia, harmonogram programu





## Obsah

<b>1</b>	<b>Úvod a cieľ práce</b>	<b>12</b>
1.1	Úvod . . . . .	12
1.2	Cieľ práce . . . . .	12
<b>2</b>	<b>Systém iOS</b>	<b>13</b>
2.1	História . . . . .	13
2.2	Architektúra . . . . .	13
2.3	Životný cyklus aplikácie . . . . .	16
2.4	SDK . . . . .	18
2.5	Xcode . . . . .	18
2.6	Objective-C . . . . .	18
<b>3</b>	<b>Harmonogram programu</b>	<b>20</b>
3.1	Brožúra . . . . .	20
3.2	PDF . . . . .	20
3.3	Mobilná aplikácia . . . . .	20
3.4	Existujúce riešenia mobilných aplikácií . . . . .	21
<b>4</b>	<b>Metodika</b>	<b>23</b>
4.1	Požiadavky na aplikáciu . . . . .	23
4.2	Návrh . . . . .	24
<b>5</b>	<b>Realizácia</b>	<b>27</b>
5.1	Práca s API . . . . .	27
5.2	Rozloženie obrazovky . . . . .	28
5.3	Prihlasovacia obrazovka . . . . .	29
5.4	Dokončenie registrácie . . . . .	31
5.5	Resetovanie hesla . . . . .	32
5.6	Hlavná obrazovka . . . . .	32
5.7	Program . . . . .	33
5.8	Profil . . . . .	36
5.9	Informácie . . . . .	37
5.10	Oblíbené . . . . .	38
5.11	Twitter wall . . . . .	41
<b>6</b>	<b>Diskusia</b>	<b>43</b>
<b>7</b>	<b>Záver</b>	<b>44</b>
<b>8</b>	<b>Literatúra</b>	<b>45</b>
	<b>Prílohy</b>	<b>47</b>

OBSAH	10
A Diagram aktivít	48
B Storyboard	49

## Zoznam obrázkov

Obrázok 1: Podiel mobilných operačných systémov na trhu (Net Applications, 2016)	13
Obrázok 2: Kľúčové objekty aplikácie (Apple Inc., 2015)	17
Obrázok 3: Stavby aplikácie (Apple Inc., 2015)	18
Obrázok 4: Model databáze	24
Obrázok 5: Detail Storyboardu	25
Obrázok 6: Landing page - Game developers session	26
Obrázok 7: Časti obrazovky	28
Obrázok 8: Obrazovka - Prihlásenie	29
Obrázok 9: Obrazovka - Registrácia	31
Obrázok 10: Obrazovka - Resetovanie hesla	32
Obrázok 11: Obrazovka - Program, Výber dní	33
Obrázok 12: Obrazovka - Detail, Hodnotenie	34
Obrázok 13: Obrazovka - Profil	36
Obrázok 14: Obrazovka - Informácie	37
Obrázok 15: Obrazovka - Oblíbené	38
Obrázok 16: Twitter obrazovka	41
Obrázok 17: Diagram aktivít - Prihlásenie cez Facebook	48
Obrázok 18: Storyboard aplikácie	49

# 1 Úvod a cieľ práce

## 1.1 Úvod

Počet užívateľov smartphonov stále narastá a s ním aj dopyt po mobilných aplikáciách. Tie sa postupom času stávajú štandardnou súčasťou rôznych produktov a služieb (Gartner Inc., 2015). Vzhľadom na praktickosť smartphonov sú aplikácie bezpochyby skvelým nástrojom na upriamenie pozornosti potencionálnych zákazníkov voči daným produktom. Zároveň dokážu užívateľom sprostredkovať informácie v prehľadnej podobe.

Organizátori konferencií a ďalších podujatí chcú držať krok s dobou, a preto sa ich čo raz viacej snaží poskytnúť svojim návštevníkom šikovnú aplikáciu, kde by našli všetko podstatné. Avšak produkty, ktoré ponúkajú podobné služby, často nespĺňajú ich predstavy alebo požiadavky.

Azda prvým úspešným operačným systémom na smartphone bol iOS, ktorý zavítal na trh s mnohými revolučnými funkciami. Aj to je jeden z dôvodov, prečo sa táto práca zameriava práve na platformu iOS.

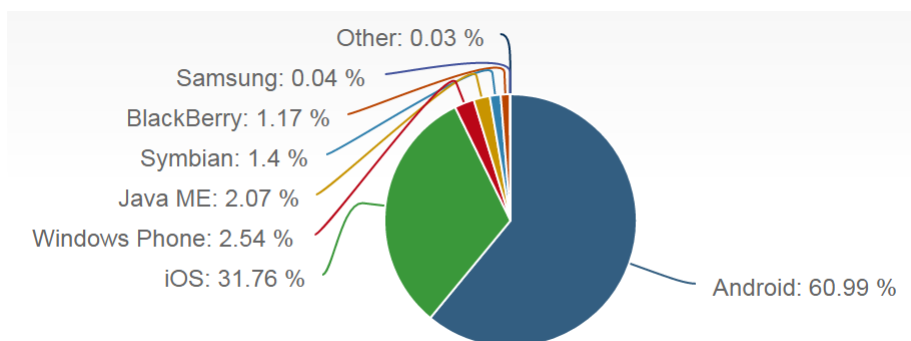
## 1.2 Cieľ práce

Cieľom tejto bakalárskej práce je návrh a následná implementácia mobilnej aplikácie pre platformu iOS splňujúca požiadavky pre konferenciu Game Developers Session v Prahe.

Účelom aplikácie je nahradiť brožúru konferencie vo vytlačenej podobe. Je určená pre návštevníkov danej konferencie, ktorým by mala zjednodušiť prístup k informáciám o prednáškach aj prednášajúcich v podobe prehľadného časového harmonogramu. Zároveň by mala používateľom poskytnúť spôsob hodnotenia a komentovania jednotlivých prednášok. V neposlednom rade musí aplikácia umožňovať organizátorom jednoduchým spôsobom informovať účastníkov o celkovom dianí pomocou tzv. Twitter wallu.

## 2 Systém iOS

iOS je v poradí druhým najpoužívanejším operačným systémom pre mobilné zariadenia (Net Applications, 2016). Je vytvorený a vyvíjaný firmou Apple Inc., a preto je určený výhradne pre ich produkty. Momentálne ho podporujú zariadenia ako iPhone, iPod touch, iPad a najčerstvejší prírastok Apple TV.



Obrázok 1: Podiel mobilných operačných systémov na trhu (Net Applications, 2016)

### 2.1 História

Systém bol z počiatku pomenovaný iPhone OS. Jeho vznik sprevádzalo predstavenie iPhone, jedného z prvých dotykových telefónov, v januári 2007. Vzhľadom na obrovský úspech, ktorý zožal, bolo samozrejmé, že sa bude naďalej vyvíjať. V roku 2008 nasledoval iPhone 3G s verziou iPhone OS2, ktorá bola doplnená o možnosť využívania aplikácií tretej strany. To položilo základ nového zamerania pre programátorov. Až s príchodom štvrtej verzie v roku 2010 dostal systém označenie iOS. Aktuálna je verzia 9, ktorá od verzie 4 prešla rôznymi dizajnovými zmenami a zároveň nadobudla mnoho funkcií, ako napríklad multitasking, ukladanie dát na iCloud, Siri, atď (Rhiannon Williams, 2015).

### 2.2 Architektúra

Vzhľadom na to, že aplikácia nekomunikuje priamo s hardvérom, musí komunikovať s množinou systémových rozhraní, tým pádom je jednoduchšie vyvíjať aplikácie pre zariadenia s rôznymi hardvérovými kapacitami. Architektúra iOS je rozdelená na vrstvy. Jednotlivé vrstvy sú zoradené od najnižšej (low-level) až po najvyššiu (high-level). Vývojárom sa odporúča pracovať s najvyššími vrstvami ako je Cocoa Touch, a len v nevyhnutných prípadoch siahať do nižších vrstiev. Je to z toho dôvodu, že nižšie vrstvy poskytujú základné služby, ktorých používanie je zbytočne náročné, na rozdiel od vyšších vrstiev, s ktorými sa pracuje omnoho sofistikovanejšie. Zároveň sa pri ich používaní redukuje množstvo kódu (Apple Inc., 2014).

### Core OS Layer

Obsahuje nízko úrovňové služby, ktoré nemusia byť priamo využívané aplikáciou, ale pravdepodobne ich využívajú ostatné frameworky.

- System - obsahuje ovládače, nízko úrovňové UNIX rozhrania a jadro, ktoré je zodpovedné za chod operačného systému. Stará sa napríklad o vlákna, manažment virtuálnej pamäte a súborový systém.
- Core Bluetooth - skenovanie dostupného bluetooth príslušenstva v okolí, pripojenie/odpojenie zariadenia, notifikácia pri zmene dostupnosti bluetooth rozhraní, broadcast informácií zo samotného zariadenia.
- Network Extension - zabezpečuje konfiguráciu VPN (Virtual Private Network) tunelov.

(Apple Inc., 2014)

### Core Services Layer

Táto vrstva poskytuje aplikáciám základné služby, ktoré umožňujú vývojárom pracovať s lokáciou (Core Location), sociálnymi sieťami, alebo aj s iCloudom (CloudKit).

- Core Foundation Framework - manažuje dátum a čas, reťazce, kolekcie, zväzky, porty a sockety.
- EventKit - podpora práce s kalendárom a pripomienkami. Umožňuje CRUD operácie s udalosťami a pripomienkami v kalendári.
- Multipeer Connectivity Framework - podpora peer-to-peer funkcie pomocou bluetooth. Slúži na nadviazanie spojenia s okolitými zariadeniami a následnú výmenu dát. Najčastejšie sa využíva pri multiplayer hrách.
- In-App Purchase - zaručuje zákazníkovi bezpečné a overené online nákupy priamo z aplikácie. Využívané najmä pri tzv. micro transactions.
- Ad Support Framework - podpora zobrazovania reklám. Aj napriek tomu, že sú otravné, je to často jediná forma, akou aplikácia zarába, najmä v prípade, ak sa jedná o bezplatnú verziu.

(Apple Inc., 2014)

### Media Layer

Zastrešuje nielen audio a video technológie, ale aj celkový grafický výzor.

1. Grafické technológie - obsahujú grafické frameworky, ktoré vykresľujú text a elementy na obrazovke. Stará sa aj o priebeh animácií a prácu s I/O obrázkov.

2. Audio technológie - poskytujú prístup do iTunes a následné nahrávanie zvukov (AV Foundation) ale aj prehrávanie (Media player) skladieb vo formátoch AAC, Apple Lossless (ALAC), A-law, IMA/ADPCM (IMA4), Linear PCM,  $\mu$ -law, DVI/Intel IMA ADPCM, Microsoft GSM 6.10, AES3-2003.
3. Video technológie - prezentácia videa na obrazovke vo full-screen alebo partial-screen móde. Podporuje nasledujúce video a kompresné formáty :
  - H.264 video, až do 1.5 Mbps, 640 na 480 pixelov, 30 fps, AAC-LC audio až do 160 Kbps, 48 kHz
  - H.264 video, až do 768 Kbps, 320 na 240 pixelov, 30 fps, AAC-LC audio až do 160 Kbps, 48 kHz
  - MPEG-4 video, až do 2.5 Mbps, 640 na 480 pixelov, 30 fps, AAC-LC audio až do 160 Kbps, 48 kHz

(Apple Inc., 2014)

### Cocoa Touch Layer

Ako bolo spomínané Cocoa Touch je najvyššia vrstva architektúry iOS. Táto vrstva je súborom frameworkov pre podporu vývoja aplikácií s bohatým grafickým užívateľským rozhraním. Skladá sa z troch hlavných frameworkov:

1. Foundation framework - Poskytuje základnú vrstvu Objective-C tried, ale aj mnoho tried primitívnych objektov akými sú reťazce, polia, atď. Hierarchia tried v tomto frameworku je ukotvená v koreňovej triede s názvom NSObject (Apple Inc., 2013).
2. Application Kit framework - Appkit zaobstaráva komunikáciu medzi hardvérovými zariadeniami a obrazovkou, ale aj celkový proces vykreslovania objektov. Tým pádom obsahuje všetky objekty a triedy, ktoré sa nejakým spôsobom prejavujú v užívateľskom rozhraní. Sú to napríklad okná, tlačítka, textové polia a iné (Apple Inc., 2014).
3. Core Data framework - účelom Core Data je uchovávanie dát pomocou SQLite databázy. Hlavnou výhodou tohoto frameworku je pristupovanie k dátam cez objekty. Používaním Core Data sa bežne zníži množstvo kódu o 50 až 70 percent (Apple Inc., 2015).

## 2.3 Životný cyklus aplikácie

Každá aplikácia sa začína funkciou `main`, ktorú vytvorí Xcode automaticky pri založení projektu. Vyzerá nasledovne:

```
1 | #import
2 | #import "AppDelegate.h"
3 |
4 | int main(int argc, char * argv[])
5 | {
6 |     @autoreleasepool {
7 |         return UIApplicationMain(argc, argv, nil,
8 |                                 NSStringFromClass([AppDelegate class]));
9 |     }
```

Kód 1: Funkcia `main`

Jedinou úlohou `main`u je zavolať triedu `UIApplicationMain`, ktorá zaobstará nasledujúce funkcie:

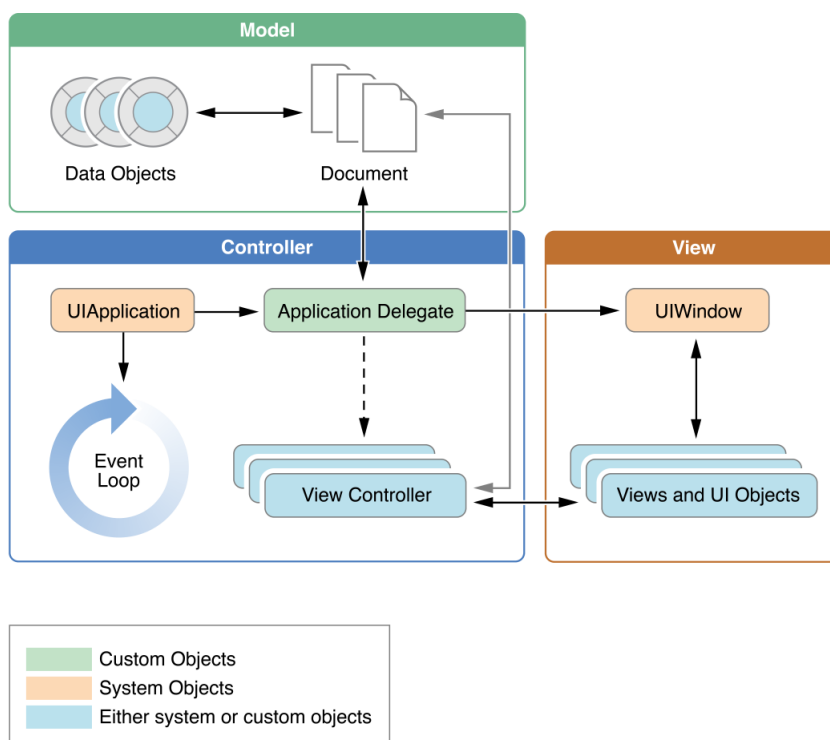
- načítanie užívateľského rozhrania
- načítanie základných objektov
- zavolanie vlastného inicializačného kódu
- vytvorenie a spustenie hlavného cyklu

### Štruktúra aplikácie

Aplikácie využívajú tzv. model-view-controller štruktúru (viď obrázok 2). Táto štruktúra je podstatnou súčasťou iOS, pretože rozdeľuje dáta a logiku od vizuálnej prezentácie. Tým eliminuje problém vizuálnej prezentácie rovnakých dát na obrazovkách s rozdielnymi rozlíšeniami. Model pracuje s dátami, controller tvorí logiku aplikácie a view zobrazuje výstupy užívateľovi.

Centrom aplikácie je `UIApplication`, ktorý sprostredkováva komunikáciu medzi systémom a ostatnými objektmi. Pomocou hlavného cyklu spracováva udalosti postupne tak, ako ich prijíma (asynchrónne) (Apple Inc., 2015).



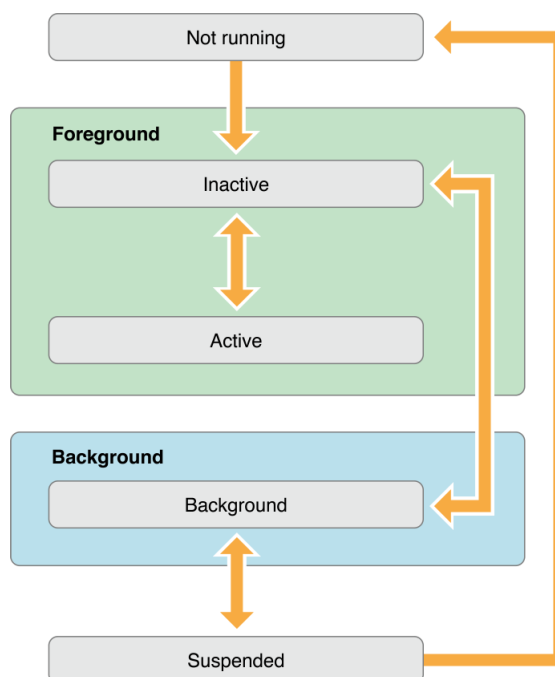


Obrázok 2: Kľúčové objekty aplikácie (Apple Inc., 2015)

### Stavy aplikácie

Existuje 5 rôznych stavov, v ktorých sa môže aplikácia nachádzať (viď obrázok 3). Tieto stavy určuje systém na základe prijímaných udalostí. Pri prechode z jedného stavu do druhého sa zavolá metóda špecifická pre každý jednotlivý stav. V tele týchto metód má programátor možnosť reagovať na meniaci sa stav napr. pri prechode z pozadia na popredie stiahne aktuálne dáta zo serveru a prekreslí danú obrazovku (Apple Inc., 2015).

- Vypnutá (Not running) - aplikácia ešte nebola spustená, alebo ju systém ukončil.
- Neaktívna (Inactive) - aplikácia beží v popredí, ale nezaznamenáva žiadne udalosti.
- Aktívna (Active) - normálny stav aplikácie pri používaní. Beží v popredí a zaznamenáva udalosti.
- Pozadie (Background) - aplikácia je v pozadí a zároveň môže vykonávať príkazy. Avšak užívateľ má oprávnenie v systémových nastaveniach zákázať aplikácii akokoľvek pracovať v pozadí.
- Pozastavená (Suspended) - aplikácia zostáva zachovaná v operačnej pamäti, ale nemôže vykonávať žiadne príkazy. Ak nastane situácia, že systém bude potrebovať viac pamäte, tak aplikáciu ukončí, aby uvoľnil miesto v popredí.



Obrázok 3: Stavby aplikácie (Apple Inc., 2015)

## 2.4 SDK

iOS SDK je dostupné všetkým bez poplatku. Taktiež vývoj aj testovanie aplikácií na simulátore je zdarma. Avšak pre nahrávanie a testovanie na reálnom mobilnom zariadení je nutné byť členom Apple Developer Programu. Členom sa môže stať ktokoľvek, koho zaujíma vývoj pre Apple platformy a je ochotný platiť 99 USD ročne. Členstvo ďalej umožňuje publikovať aplikácie cez App Store a zapájanie sa do rôznych špecifických developerských konferencií (Apple Inc., 2016).

## 2.5 Xcode

Xcode je veľmi prepracovaný nástroj určený k vývoju OS X a iOS aplikácií. Umožňuje spúšťanie projektov vo všetkých verziách iPhonov a iPadov pomocou simulátora, ale aj kompiláciu, debugging, atď. Xcode poskytuje zdarma firma Apple. Zároveň sa stará aj o jeho aktualizácie pri každom vydaní novej verzie systému (Kochan , 2011).

## 2.6 Objective-C

Objective-C je objektovo orientovaný programovací jazyk, najčastejšie používaný pri vývoji aplikácií pre OS X a iOS systémy od firmy Apple. Bol navrhnutý začiatkom 80 tých rokov a oficiálne licencovaný v roku 1988 firmou Next software. Neskôr, v

roku 1996 firma Apple odkúpila Next software a začala využívať Objective-C pre vlastné účely. Objective-C bol založený na jazyku SmallTalk-80. V zásade sa jedná o nadstavbu jazyka C. To znamená, že obsahuje všetky dátové typy aj funkcie jazyka C, a navyše je doplnený o schopnosť práce s objektami (Kochan , 2011). Avšak v poslednej dobe začal Apple propagovať úplne nový programovací jazyk Swift, ktorý pomaly vytláča Objective-C z trhu.

## 3 Harmonogram programu

Harmonogram programu je neodmysliteľnou súčasťou konferencií, ale aj kultúrnych podujatí, akými sú napríklad koncerty, festivaly, atď. Uľahčuje účastníkovi výber prednášok/koncertov, ktorých sa zúčastní a tým pádom aj celkovú orientáciu na danej udalosti.

### 3.1 Brožúra

Brožúra je klasickou a najpoužívanjšou formou harmonogramu. Každý obdrží jeden výtlačok pri vstupe do areálu a riadi sa podľa neho. Brožúra môže byť pomerne nepraktická, pokiaľ sa nevmestí do vrecka a človek nemá po ruke svoju tašku alebo batoh. Taktiež sa môže jednoducho roztrhnúť, premočiť alebo stratiť. Predsa len si človek dá skôr pozor na svoj mobilný telefón, ako na kus papiera. V neposlednom rade brožúra nedokáže pracovať s aktuálnymi dátami, pretože grafický návrh musí byť vytvorený vždy niekoľko dní pred udalosťou kvôli samotnej tlači. Do aplikácie stačí nahráť dáta cez webové rozhranie aj deň pred konaním a aktualizovať ich kedykoľvek v priebehu podujatia.

### 3.2 PDF

Program v PDF formáte je novodobejším riešením. Je možné si ho stiahnuť do notebooku alebo mobilného zariadenia a prezeráť ho pomocou dostupného PDF readeru. Problém nastane až vo chvíli, keď má užívateľ záujem si prečítať aj podrobnejšie informácie o niektorom bode v programe a musí ho v PDF dokumente vyhľadať

### 3.3 Mobilná aplikácia

Mobilná aplikácia je navhodnejším riešením, pokiaľ vlastní väčšina návštevníkov smartphony. Avšak táto situácia je dnes už bežnou realitou. Jednou z najcennejších funkcionalít pre organizátorov je feedback od užívateľov, ktorí majú možnosť hodnotiť a komentovať dané prednášky. Pravdepodobne jedinou nevýhodou je azda výdrž batérie, na ktorú je nutné dbať na rozdiel od klasickej brožúry.

Ďalšie výhody:

- aktualizácia harmonogramu v prípade zmeny programu počas podujatia
- prezeranie Twitter príspevkov od organizátorov aj účastníkov
- zostavovanie vlastného harmonogramu

### 3.4 Existujúce riešenia mobilných aplikácií

Na trhu sa už samozrejme vyskytujú produkty, po ktorých môžu organizátori konferencií siahnuť. Tieto produkty poskytujú spôsob, akým sa dá prezentovať obsah udalostí ich návštevníkom cez mobilé platformy. Na základe požiadaviek pre aplikáciu (viď sekcia 4.1) som stanovil sedem kritérií, podľa ktorých som dané produkty hodnotil.

1. Dizajn - týka sa predovšetkým vizuálnej stránky, napríklad rozmiestnenie prvkov, použitie komponentov, animácie a farebná zladenosť.
2. Prehľadnosť - do prehľadnosti je zarátaná aj náročnosť manipulácie a celková orientácia vo vnútri aplikácie. Snažil som sa na aplikáciu pozrieť z ulhu bežného používateľa a zistiť, ako rýchlo sa v nej naučím pohybovať a dostať sa k informáciám, o ktoré mám záujem.
3. Režim viacerých sálov - horizontálne zobrazenie harmonogramu prednášok každého sálu v riadkoch pod sebou.
4. Twitter wall - zobrazenie príspevkov z daného Twitter účtu.
5. Oblíbené - možnosť pridávať si obľúbené prednášky.
6. Hodnotenie - možnosť hodnotiť/komentovať prednášky a zároveň si zobrazit komentáre ostatných ľudí.
7. Cena - udávaná v USD za prezentáciu jednej udalosti.

Vybral som štyri najpopulárnejšie služby, ktorých aplikácie som jednotlivo otestoval, a následne zhodnotil podľa stanovených kritérií.

- Attendify ([www.attendify.com](http://www.attendify.com)) - na ich webovej stránke som sa zaregistroval a vytvoril testovaciu konferenciu z pohľadu organizátora. V mobilnej aplikácii sa mi konferencia zobrazila po zadaní kódu vygenerovaného na webe. Vďaka pekne spracovanému dizajnu a prehľadnosti je Attendify určite najlepšou voľbou spomedzi týchto štyroch produktov. Vytkol by som jedine vertikálne zobrazenie harmonogramu viacerých sálov, ktoré sa môže stať pomerne nepraktické.
- EventMobi ([www.eventmobi.com](http://www.eventmobi.com)) - som testoval pomocou demo aplikácie, ktorá je síce prehľadná, ale po grafickej stránke nezaujímavá. Neumožňuje ani prezeranie Twitter príspevkov priamo v aplikácii. Vo výsledku neposkytuje tak dobrú službu ako Attendify za rovnakú cenu.
- quickMobile ([www.quickmobile.com](http://www.quickmobile.com)) - žiaľ demo verzia nefungovala, takže som hodnotil na základe screenshotov a videí z aplikácie, preto som nebol schopný posúdiť režim viacerých sálov. Z dizajnového hľadiska pôsobila veľmi zaujímavo. K cene som sa však na ich webe tiež nedostal.
- eventsXD ([www.eventsd.com](http://www.eventsd.com)) - je jedinou bezplatnou variantou, ale o tom samozrejme svedčí aj kvalita spracovania. Aplikácia je verejne dostupná a sú v nej nahrané rôzne konferencie, v ktorých je možné vidieť reálne dáta. Vizualne prevedenie aplikácie je doslova katastrofálne a ani celková funkcionálnosť a prehľadnosť to žiaľ nie sú schopné kompenzovať.

Názov	Design	Prehľadnosť	Režim viacerých sálov	Twitter wall	Obľúbené	hodnotenia	Cena (USD)
<b>Attendify</b>	7	8	NIE	ÁNO	ÁNO	ÁNO	999
<b>EventMobi</b>	5	7	NIE	NIE	ÁNO	ÁNO	999
<b>quickMobile</b>	7	7	N/A	ÁNO	ÁNO	ÁNO	N/A
<b>eventsXD</b>	2	4	NIE	NIE	ÁNO	ÁNO	Zdarma

Tabuľka 1: Zhodnotenie existujúcich riešení

Z vyhodnotenia vyplýva, že organizátori majú možnosť poskytnúť svojim návštevníkom nie veľmi vydarenú aplikáciu bezplatne alebo kvalitnú aplikáciu v prepočte za 24 100 kč. Pokiaľ však majú náročnejšie požiadavky, sú nútení priplatiť si za vývoj vlastnej aplikácie na mieru.

## 4 Metodika

### 4.1 Požiadavky na aplikáciu

Aplikácia musí byť podporovaná verziou systému iOS 8.0 a vyššie. V súčasnosti to znamená, že aplikácia je určená pre nasledujúci zoznam modelov zoskupených podľa rovnakej veľkosti:

- iPhone 6 plus, iPhone 6 plus s
- iPhone 6, iPhone 6s
- iPhone 5, iPhone 5s, iPhone SE
- iPhone 4s

Vzhľadom na odlišné veľkosti modelov sú rozdielne aj rozlíšenia obrazoviek (viď tabuľka 2). Preto je nutné rozloženie elementov na rôznych zariadeniach programovo prepočítavať s ohľadom na výšku a šírku displeja.

	iPhone 6 plus	iPhone 6	iPhone 5	iPhone 4s
<b>Rozlíšenie (px)</b>	1920 x 1080	1334 x 750	1136 x 640	960 x 640
<b>Výška</b>	158.2	138.3	123.8	115.2
<b>Šírka</b>	77.9	67	58.6	58.6

Tabuľka 2: Veľkosti zariadení

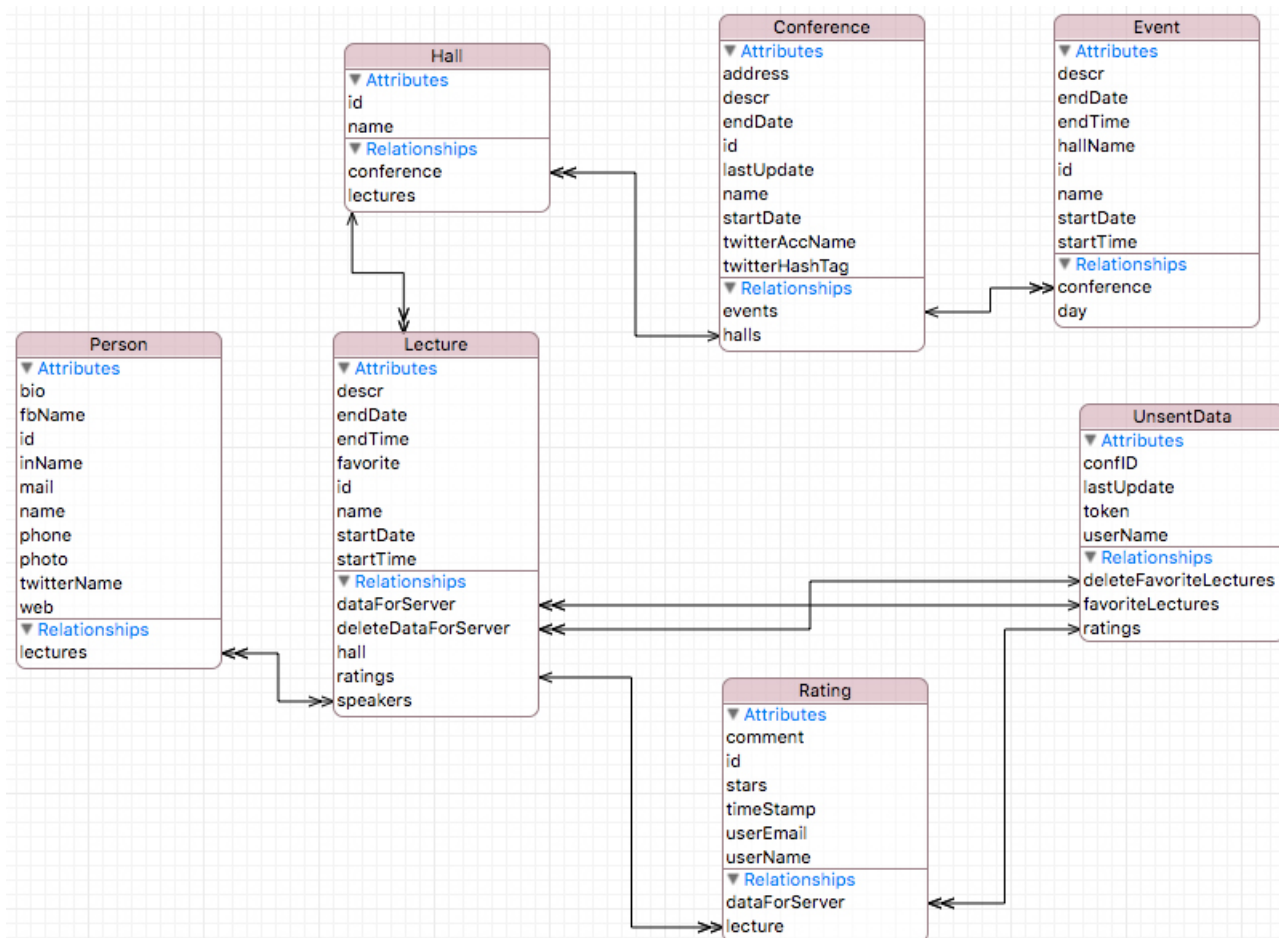
Základnou úlohou aplikácie je komunikácia so serverom pomocou API volaní. Aplikácia si stiahne dáta danej konferencie vo formáte JSON a zobrazí harmonogram, ktorý musí byť interaktívny. To znamená, že užívateľovi sa po kliknutí na príslušné políčko zobrazí obrazovka s podrobnejšími informáciami či už o samotnej prednáške alebo o jej prednášajúcom. Hlavným konceptom je intuitívne používanie, jednoduchosť a prehľadnosť. Používateľ by mal byť schopný ihneď po nainštalovaní ovládať aplikáciu a porozumieť jej funkciám. Po spustení má možnosť registrácie či už pomocou Facebooku, alebo e-mailu. Prihlásenie je dobrovoľné, používanie aplikácie je sprístupnené aj anonymným užívateľom. Obaja registrovaní aj anonymní užívatelia si môžu navoliť obľúbené prednášky a zostaviť si tak vlastný program. Registrácia však so sebou prináša výhody v podobe možnosti hodnotenia, komentovania ale aj synchronizácie obľúbených prednášok v prípade používania viacerých mobilných zariadení. Všetky stiahnuté dáta sa ukladajú do internej databázy, čím je umožnená práca s aplikáciou aj v offline režime. Nevýhodou tohto režimu je znemožnené sťahovanie aktuálnych dát v prípade hocakej zmeny programu. V prípade offline používania nie je žiaľ sprístupnená ani komunikácia organizátorov cez Twitter účet, ktorý je priamo zabudovaný do aplikácie.

## 4.2 Návrh

Po analýze požiadaviek a naštudovaní API som začal s návrhom databázy a tzv. storyboardu.

### Databáza

Ako som už vyššie spomínal, iOS aplikácie si uchovávajú dáta v SQLite databáze pomocou Core Data Frameworku. Pri návrhu som bral do úvahy všetky dáta posielané serverom a na ich základe som vytvoril nasledovný entitno-relačný model.



Obrázok 4: Model databáze

- Konferencia (Conference) - je to hlavná entita. Okrem základných informácií ako je meno, popis, začiatok a koniec, obsahuje aj názov twitter účtu (twitterAccountName) a hashtag konferencie (twitterHashTag). Atribút lastUpdate je string časovej značky (time stamp) poslednej stiahnutej verzie dát. Konferencia má dve 1:n väzby - sály (halls) a udalosti (events).



- Sála (Hall) - je najjednoduchšou entitou v modeli. Obsahuje len svoj názov a 1:n väzbu prednášky (lectures).
- Prednáška (Lecture) - atribút obľúbená (favorite) je dátového typu boolean a označuje záujem užívateľa o danú prednášku. Väzba prednášajúci (lecturers) k entite Osoba je jediná m:n väzba v celom modeli.
- Udalosť (Event) - entita udalosť je podobná prednáške, ale neviaže sa na konkrétny sál. Môžeme ju vnímať ako nejakú prestávku medzi prednáškami, napríklad obed.
- Hodnotenie (Rating) - atribúty komentár (comment), časová známka (timeStamp), užívateľské meno (userName), užívateľský email (userEmail) sú typu string. Hviezdičky (stars) je typu integer, ktorý môže nadobudnúť hodnoty od 0 do 5.
- Osoba (Person) - Fotka (photo) sa neodkazuje na miesto v databáze, kde je uložený obrázok, je typu string a obsahuje webovú URL na daný obrázok.
- Neodoslané dáta (UnsentData) - pri práci užívateľa v offline režime si uchováva odkazy na hodnotenia a prednášky. Tieto dáta sa po obnovení pripojenia posielajú na server kvôli synchronizácii.

## Storyboard

Storyboard je nástroj využívaný na modelovanie grafického vzhľadu jednotlivých obrazoviek a ich väzbami (viď prílohu B). Reprezentuje akúsi štruktúru obrazoviek a prechodov medzi nimi. To znamená, že do istej miery definuje logiku aplikácie.

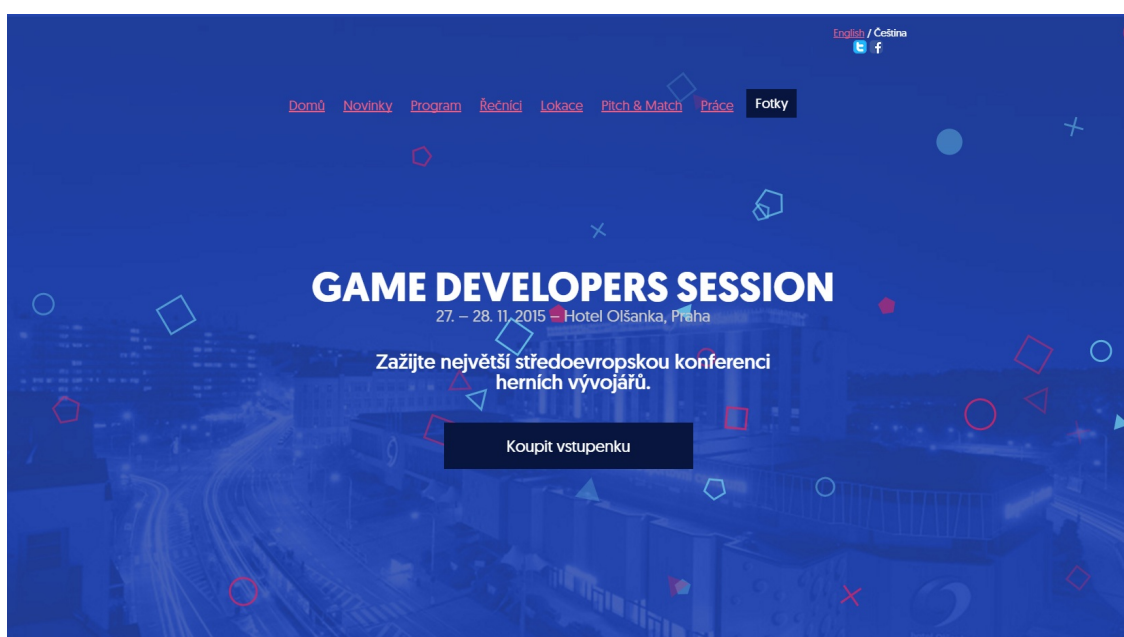


Obrázok 5: Detail Storyboardu

## Dizajn

Dizajn bol inšpirovaný najmä systémovými a komerčne najpoužívanejšími aplikáciami, kvôli zachovaniu dizajnových zásad iOS platformy. Neumožňuje zbytočné viacnásobné zanorovanie obrazoviek, pri ktorých sa dá ľahko stratiť prehľad. Okrem klikania na tlačítka aplikácia podporuje aj niektoré bežné gestá, ako ťahanie a pustenie (drag and drop) alebo posnutie (swipe). Vďaka týmto vlastnostiam poskytuje jednoduché užívateľské rozhranie a používanie aplikácie by tým pádom malo pôsobiť natívne.

Samotná farebná paleta je tak isto veľmi podstatná pre celkový dojem, a preto bola prispôbena farbám použitým na webovej stránke konferencie.



Obrázok 6: Landing page - Game developers session

## 5 Realizácia

### 5.1 Práca s API

API bola vyvíjaná kolegom, ktorý ju zdokumentoval pomocou služby Apiary. V dokumentácii sú uvedené URL odkazy na všetky volania vrátane vstupných a výstupných parametrov vo formáte JSON.

Na vytváranie HTTPs requestov som použil externú knižnicu AFNetworking. Najskôr sa inicializuje manažér (`AFHTTPRequestOperationManager`). Kvôli identifikácii a autorizácii každého HTTPs requestu sa do parametrov hlavičky nastavuje užívateľský email a bezpečnostný token (Kód 2). Týmto spôsobom je server ochránený pred neautorizovaným čítaním a zapisovaním dát do databáze.

Po inicializácii manažéru je možné volať na server cez URL adresy. V kóde 3 je uvedený príklad volania metódou GET s parametrami vo formáte JSON.

```

1 //HTTPmanager initialization
2 AFHTTPRequestOperationManager *manager;
3 manager = [AFHTTPRequestOperationManager manager];
4 manager.responseSerializer.acceptableContentTypes = [manager.
    responseSerializer.acceptableContentTypes setByAddingObject
   :@"text/html"];
5 manager.requestSerializer = [AFJSONRequestSerializer serializer];
6
7 //setting HTTP headers
8 user = [User getCurrentUser];
9 if (user.mail.length > 0) {
10     [manager.requestSerializer setValue:user.mail
    forHTTPHeaderField:@"userMail"];
11 }
12
13 if (user.token.length > 0) {
14     [manager.requestSerializer setValue:user.token
    forHTTPHeaderField:@"userToken"];
15 }

```

Kód 2: Inicializácia manažéru

```

1 - (void)getConferenceInfoWithSuccess:(void (^)(void))success
2     failure:(void (^)(NSError *error))
3     failure
4 {
5     NSDictionary *params = @{@"conferenceID": [confID stringValue
6     ]};
7     NSString *urlString = @"https://example.api.com/get/conf";
8
9     [manager GET:urlString parameters:params success:^(
10     AFHTTPRequestOperation *operation, id responseObject) {
11         NSDictionary *confDict = responseObject;
12         [database loadConferenceIntoDatabase:confDict completion
13        :^(BOOL finished) {
14             success();
15         }];
16     }];
17 }

```

```

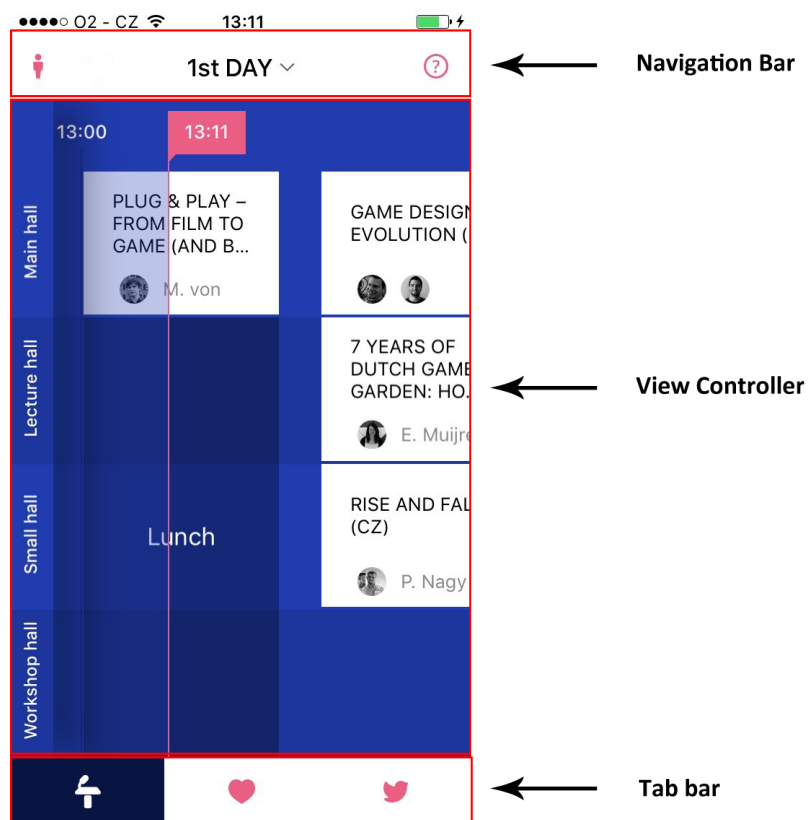
11     }]];
12
13     } failure:^(AFHTTPRequestOperation *operation, NSError *error)
14         {
15         failure(error);
16     }]];
17 }

```

Kód 3: HTTP request

## 5.2 Rozloženie obrazovky

Každá obrazovka je tvorená View Controllerom obsahujúcim jedno hlavné View, v ktorom sa vykresluje všetok obsah. Úlohou View Controlleru je okrem vykreslovania obsahu aj riadenie vstupov od užívateľa, preto je v ňom naimplementovaná všetka funkcionálnosť. Niektoré z obrazoviek obsahujú aj Navigation Bar v hornej časti alebo Tab Bar v spodnej časti. Konkrétne príklady sú uvedené ďalej v tejto kapitole.



Obrázok 7: Časti obrazovky

## 5.3 Prihlasovacia obrazovka

Po spustení aplikácie sa užívateľovi zobrazí obrazovka s možnosťami prihlásenia (Obrázok 8). Prihlásenie však nie je povinné, dá sa preskočiť pomocou tlačítka skip v pravom hornom rohu obrazovky. Užívateľ tak ostane po dobu používania aplikácie anonymný. Ak sa však rozhodne pre založenie svojho účtu, tzv. user account, môže tak učiniť dvoma spôsobmi - cez email alebo Facebookový účet.



Obrázok 8: Obrazovka - Prihlásenie

### Autentifikácia cez Email

Je to klasický spôsob registrácie a prihlasovania, ktorý sa v poslednej dobe už nevyužíva tak často ako kedysi. Je nielen zdĺhavejší, ale aj menej bezpečný. Vo vnútri aplikácie sa oba užívateľské vstupy najskôr overujú podľa regulárnych výrazov.

- Email - Regulárnych výrazov pre email je napísaných mnoho. Niektoré majú aj stovky znakov, ale ja som nechcel zachádzať do extrémov a rozhodol som sa pre jednoduchšiu variantu: `[A-Z0-9a-z._%+~]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}`

- Heslo - Musí obsahovať aspoň jedno veľké a jedno malé písmeno, a minimálne jednu číslicu:  $(?=.*[0-9])(?=.*[a-z])(?=.*[A-Z]).\{8,128\}$

Pri registrácii cez email nesmie chýbať možnosť resetovania zabudnutého hesla. Tlačítko Zabudli ste svoje heslo? (Forgot your password?) sa odkazuje na obrazovku pre resetovanie hesla (viď 5.5).

### Autentifikácia pomocou Facebooku

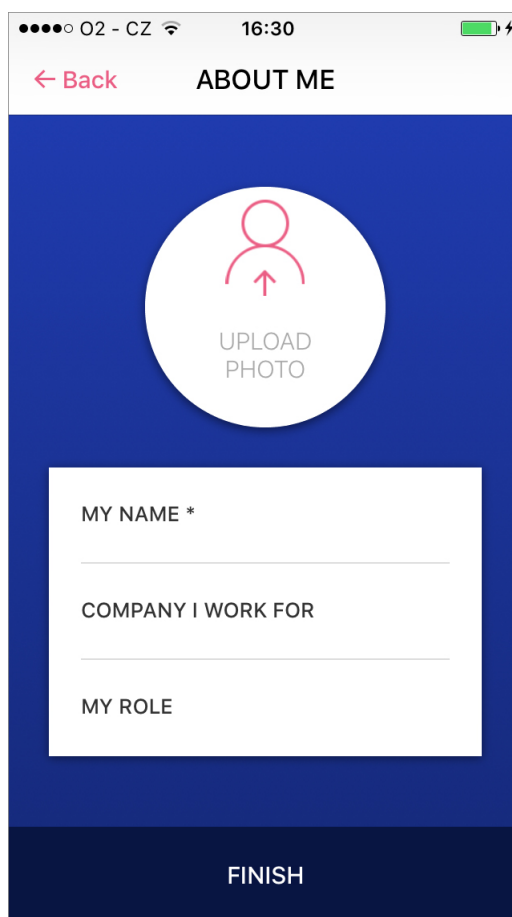
Vzhľadom na to, že služba Facebook je celosvetovo veľmi rozšírená, je tak isto tento spôsob autentifikácie čo raz obľúbenejším a stáva sa akýmsi štandardom. Implementácia je rozhodne náročnejšia ako pri predchádzajúcom spôsobe, ale z pohľadu užívateľa je rýchla a bezstarostná. Zároveň je táto varianta aj o veľa bezpečnejšia.

Postup autentifikácie pomocou služby Facebook (znázornený v prílohe A):

Po kliku na tlačítko Prihlásiť cez Facebook aplikácia načíta Facebook framework, ktorý zobrazí tzv. Web View (modálne okno s internetovým prehliadačom Safari) a nasmeruje ho na správnu URL. Po potvrdení vstupu do aplikácie, framework obdrží potrebné údaje o užívateľovi a automaticky zatvorí modálne okno. Aplikácia následne pomocou API volania posielá serveru Facebookový email, ID a token. Ten overí korektnosť údajov. Pokiaľ Facebookový token nepatrí užívateľovi s daným identifikačným číslom, tak vracia chybu (Failure). V opačnom prípade si vytiahne užívateľa z databázy a vracia jeho údaje vrátane vlastného tokenu. Nakoniec si samotná aplikácia tieto dáta uloží a pustí užívateľa na hlavnú obrazovku (viď 5.6).

## 5.4 Dokončenie registrácie

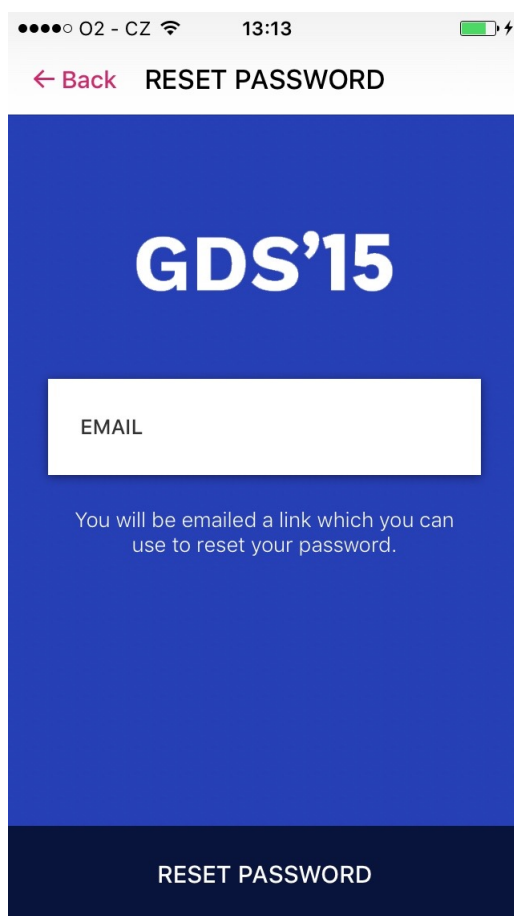
Na obrazovke sa nachádzajú 3 textové polia - meno (my name), firma (company I work for) a pozícia (my role). Meno je jediný atribút, ktorý treba povinne vyplniť. Ďalej sa tu nachádza tzv. Image view s obrázkom a textom nahraj fotografiu (upload photo). Pri kliknutí sa zobrazí ponuka akcií (**ActionSheet**), v ktorej je možnosť výberu obrázku z galérie zariadenia alebo sa priamo vyfotografovať. Keď je užívateľ spokojný s vyplnenými údajmi, odošle ich na server tlačítkom dokončiť (finish). Ešte pred samotným odoslaním je nutné konvertovať vybranú fotografiu do formátu Base64. Jedná sa o zakódovanie bitov reprezentujúcich obrázok do stringu. Ak registrácia prebehne úspešne, dáta si uloží a pustí užívateľa na hlavnú obrazovku (viď 5.6).



Obrázok 9: Obrazovka - Registrácia

## 5.5 Resetovanie hesla

Na obrazovke sa nachádza jedno textové pole, ktoré slúži na zadanie emailovej adresy. Tá je taktiež overovaná podľa regulárneho výrazu. Po kliku na tlačítko Reset Password (Resetuj heslo) sa zavolá API, ktorá prikáže serveru odoslať email s inštrukciami a URL odkazom na danú adresu. Tu nastáva nevyriešiteľný problém autorizácie. Vzhľadom na to, že užívateľ nie je v dobe resetovania hesla prihlásený, musí byť toto volanie bez overenia. Tým pádom má ktokoľvek možnosť "zaspamovať" email niekoho druhého.



Obrázok 10: Obrazovka - Resetovanie hesla

## 5.6 Hlavná obrazovka

Hlavnú obrazovku tvorí Tab Bar Controller. Je to klasická iOS komponenta ktorej súčasťou je vyššie spomínaný View Controller a Tab Bar (viď sekcia 5.2). Väčšinu plochy zaberá View Controller daného tabu. Pod ním sa nachádza spomínaný Tab Bar s tlačítkami, ktoré umožňujú užívateľovi rýchle prepínanie medzi viacerými View Controllermi. V tomto konkrétnom prípade obrazovka obsahuje 3 taby.



## 5.7 Program

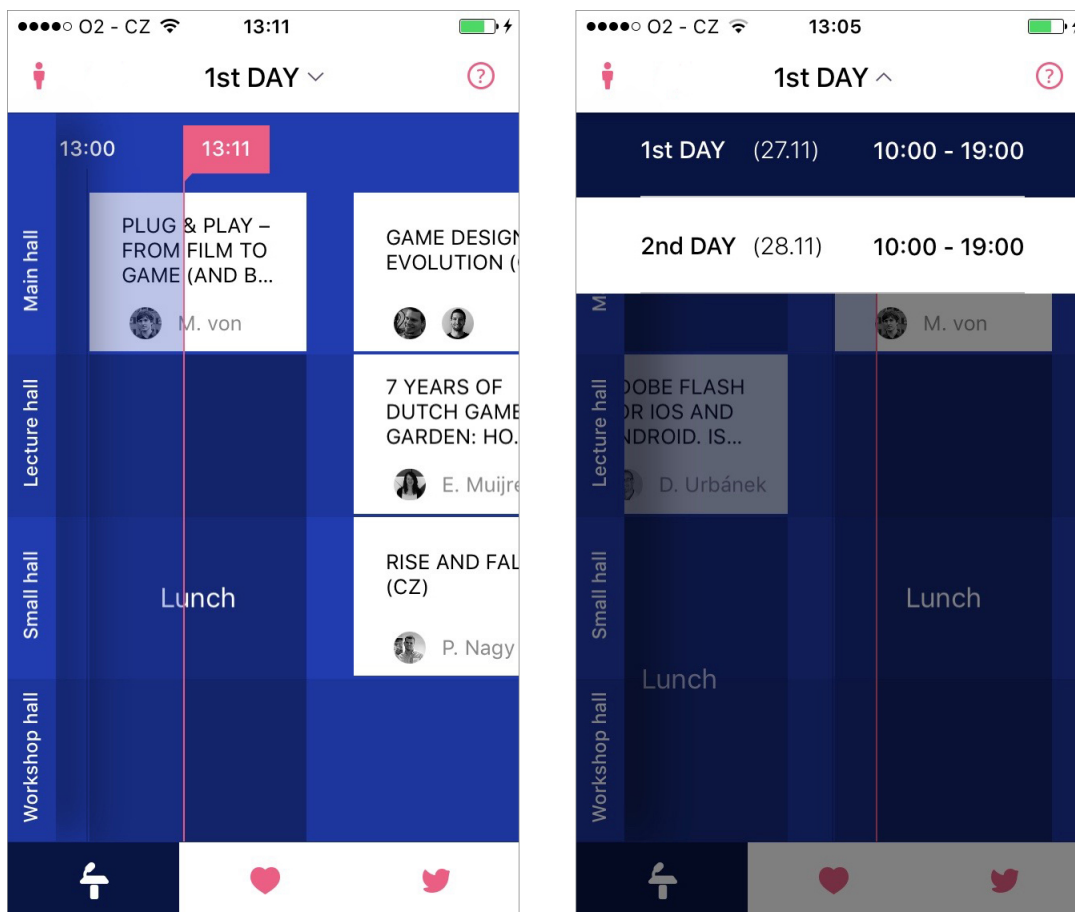
Program je najpodstatnejšou časťou aplikácie. Obsahuje Scroll view, ktoré umožňuje horizontálne aj vertikálne posúvanie. V jeho vnútri sa vykresľuje samotný harmonogram prednášok súbežne s horizontálnou časovou osou. Jednotlivé riadky sú určené prednáškovým sálom, ktoré sú jednoducho identifikovateľné vďaka úzkemu panelu v ľavom okraji obrazovky (viď obrázok 11).

V hornej časti obrazovky sa nachádza Navigation Bar s tromi tlačítkami:

**Profil** - vysunie sa modálne okno obrazovku profilu (viď sekcia 5.8).

**Informácie** - odkazuje sa na obrazovku s informáciami (viď sekcia 5.9).

**Výber dní** - pod Navigation Barom sa objaví zoznam dní konania konferencie s dátumom a časovým údajom o začiatku a konci programu. Užívateľ je oprávnený prepínať medzi dňami. Po prepnutí sa vymaže aktuálny obsah View Controlleru a načíta sa obsah pre novo vybraný deň.

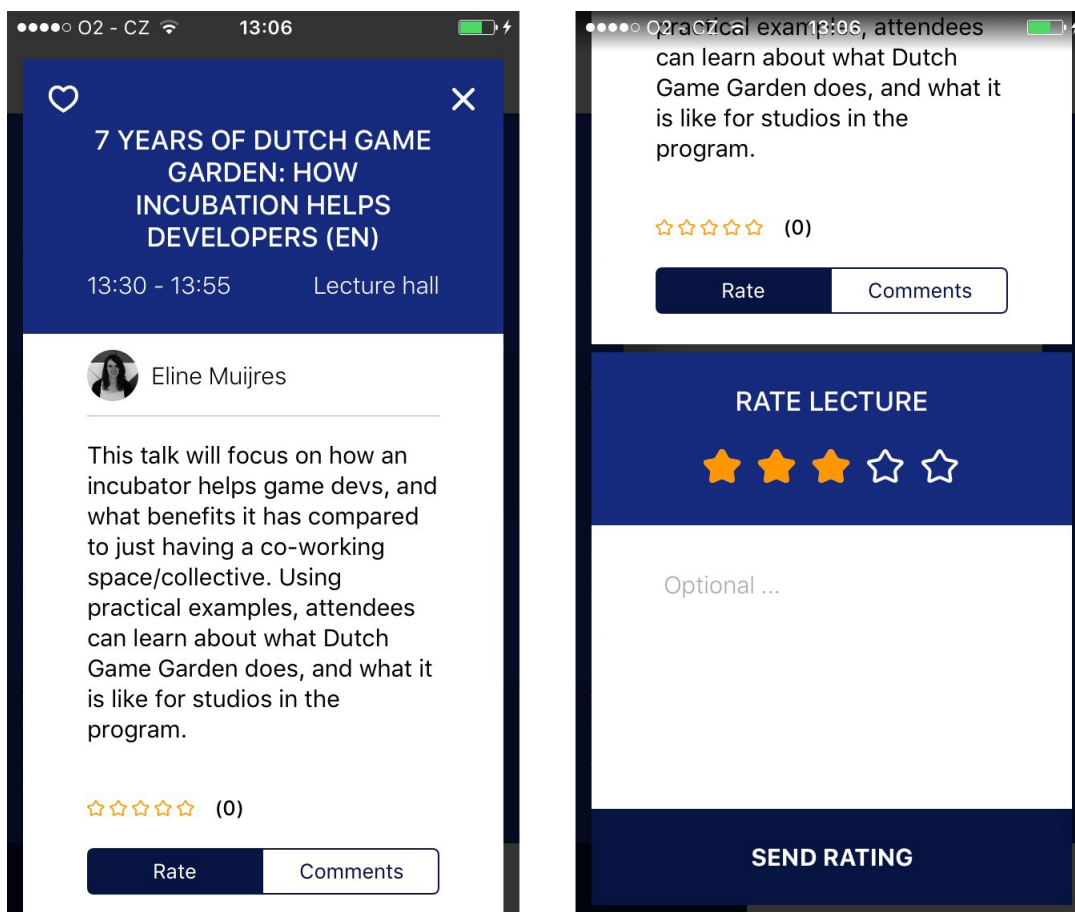


Obrázok 11: Obrazovka - Program, Výber dní

Prednášky sú zobrazené ako biele obdĺžniky obsahujúce názov prednášky, ale aj meno a fotku prednášajúceho. Prestávky sú taktiež zobrazené ako obdĺžniky s tým

rozdielom, že môžu byť natiahnuté cez viacero riadkov a majú tmavé priehľadné pozadie, aby boli kontrastne graficky odlišené od prednášok. Detail prednášky a prestávky sa otvorí po kliku na príslušný obdĺžnik.

### Detail prednášky



Obrázok 12: Obrazovka - Detail, Hodnotenie

- Tlačítka pridať do obľúbených - obrázok srdca v ľavom hornom rohu. Ak je prednáška v obľúbených, srdce je vyplnené, inak z neho ostane len obrys.
- Prednášajúci - fotka a meno prednášajúceho. Po kliku sa otvorí jeho detail.
- Popis prednášky - je zobrazený vo formáte HTML s podporou nasledujúcich značiek - tučné písmo, kurzíva, odkaz (s možnosťou otvorenia Web View), nadpis, zoznam, odstavce
- Priebežné hodnotenie - hviezdičky sú ukazovateľom priemerného hodnotenia. Za nimi je v zátvorke je udávaný počet komentárov.

- Komentáre - po otvorení detailu prednášky si aplikácia v pozadní mimo hlavného vlákna stiahne aktuálne hodnotenia a komentáre pre danú prednášku. Až po uložení týchto dát je povolená interakcia s tlačítkom, ktoré slúži ako prepínač medzi vlastným hodnotením a komentármi ostatných užívateľov.
- Vlastné hodnotenie - Každý riadne prihlásený užívateľ má možnosť pridať a editovať svoje hodnotenie. Pred odoslaním je potrebné označiť počet hviezdíčiek od jednej do piatich po prípade aj vyplniť textové pole komentáru. Možnosť hodnotenia je dostupná aj v offline režime tak, že sa dáta uložia do zariadenia a po znovu naviazaní internetového pripojenia sa v pozadí odošlú na server.

### Detail prestávky

Prosté zobrazenie názvu prestávky a popisu vo formáte HTML. Dizajn je rovnaký ako pri detaile prednášky.

### Drag and drop

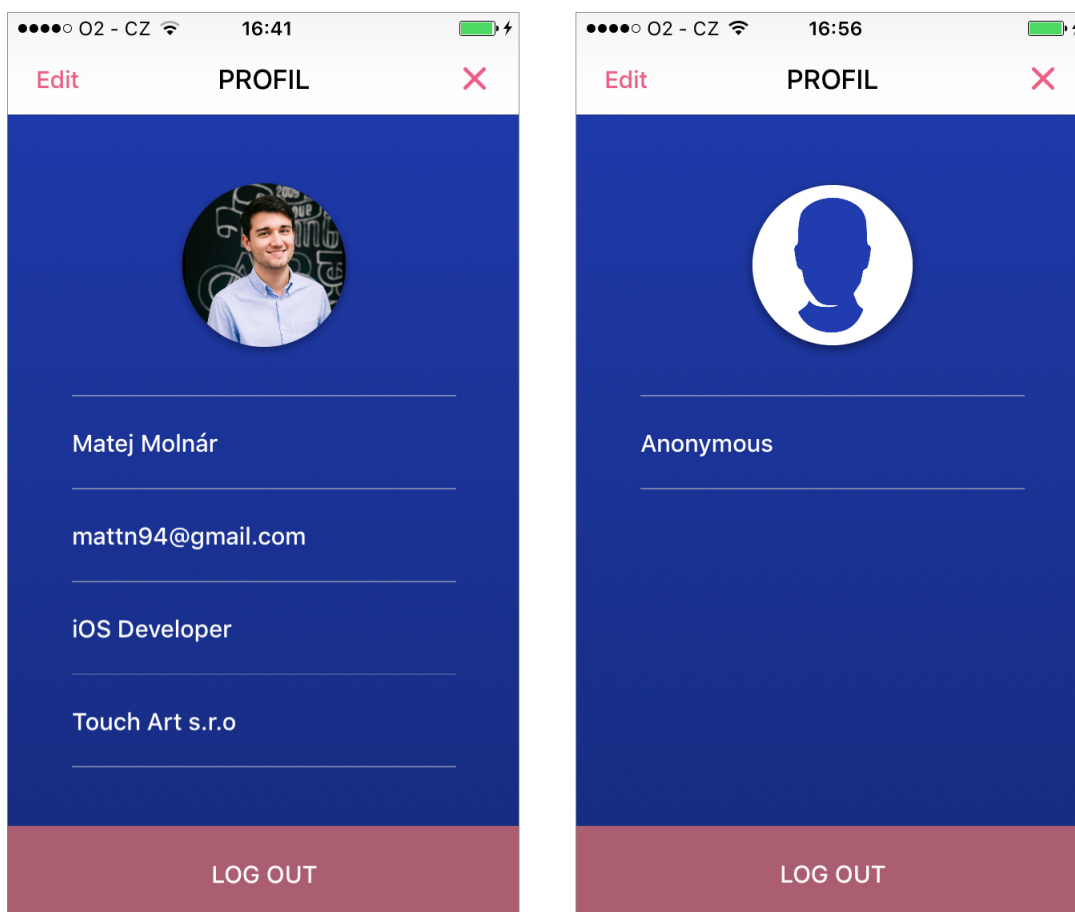
Kvôli väčšej praktickosti je v programe okrem kliku na prednášku naimplementované aj gesto drag and drop pre pridávanie prednášok do obľúbených bez potreby zobrazenia detailu. Funguje to tak, že vo vnútri Scroll View programu je pridaný tzv. Gesture Recognizer (konkrétne `UILongPressGestureRecognizer`), ktorému sa nastaví vlastne vytvorená metóda (`cellLongPressed`) volaná v momente, kedy aplikácia zaregistruje dané gesto. Ďalej je potrebné nastaviť delegáta (`setDelegate`) a minimálnu dobu stlačenia na zaregistrovanie gesta (`setMinimumPressDuration`).

```
1 |
2 | UILongPressGestureRecognizer *longPressGesture = [[  
   |     UILongPressGestureRecognizer alloc] initWithTarget:self  
   |     action:@selector(cellLongPressed:)]];  
3 | [longPressGesture setDelegate:scrollView];  
4 | [longPressGesture setMinimumPressDuration:0.3];  
5 | [scrollView addGestureRecognizer:longPressGesture];
```

Kód 4: Nastavenie Gesture Recognizeru

## 5.8 Profil

Modálne okno s údajmi, ktoré užívateľ vyplnil pri registrácii, vrátane profilovej fotografie. Pokiaľ je užívateľ prihlásený anonymne, namiesto fotografie sa zobrazí silueta. Samozrejme tu nesmie chýbať ani tlačítko pre odhlásenie (Log out). Po jeho stlačení je zobrazená systémová hláška pre potvrdenie alebo zrušenie voľby.



Obrázok 13: Obrazovka - Profil

## 5.9 Informácie

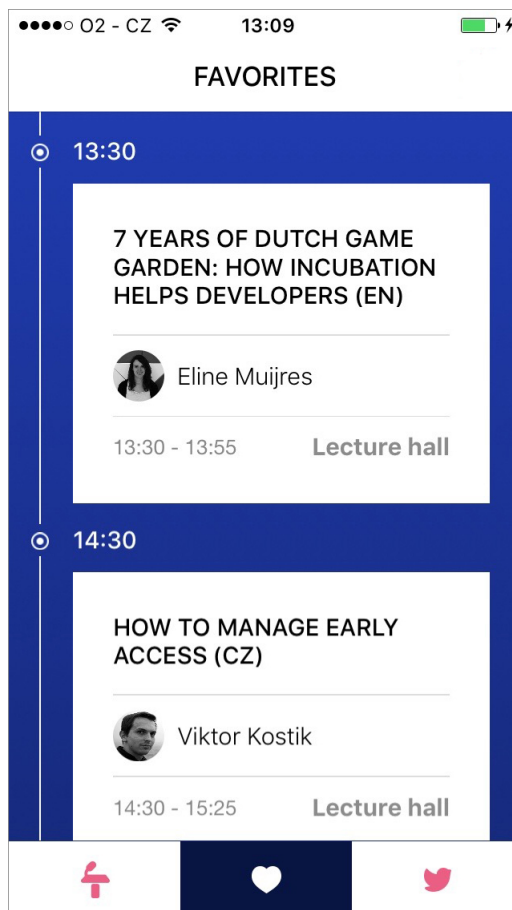
Obsahuje základné informácie o samotnej konferencii a zároveň aj o firme Touch art s.r.o, pre ktorú som túto aplikáciu vyvíjal.



Obrázok 14: Obrazovka - Informácie

## 5.10 Oblíbené

Prehľad obľúbených prednášok (valstného harmogramu) vo vertikálnom časovom slede. Tu je použitá komponenta Table View (tabuľka) s dvoma typmi buniek - času a prednášky. Kliknutie na buňku prednášky otvorí rovnaký detail ako v programe (viď sekcia 5.7). Pokiaľ v obľúbených nie je žiadna prednáška, tak je pozadie tvorené obrázkom a textom "You have no favorite lectures".



Obrázok 15: Obrazovka - Oblíbené

Pri práci s tabuľkou je potrebné naimplementovať vnútro štyroch delegátnych metód, pomocou ktorých tabuľka zistuje, čo má vykresliť a ako sa má správať:

1. `numberOfSections` - vráti tabuľke počet sekcií

```

1 |
2 | - (NSInteger)numberOfSectionsInTableView:(UITableView *)
   |   tableView
3 | {
4 |     return 1;
5 | }
```

Kód 5: Počet sekcií

2. `numberOfRowsInSection` - vracia počet riadkov pre danú sekciu.

Pred načítaním tabuľky sa do pola (`tableArray`) vložia všetky objekty, ktoré má za úlohu vykresliť. Pri volaní metóda vráti počet objektov tohto pola.

```
1 | - (NSInteger)tableView:(UITableView *)tableView
   |     numberOfRowsInSection:(NSInteger)section
2 | {
3 |     return [tableArray count] ;
4 | }
```

Kód 6: Počet riadkov

3. `heightForRowAtIndexPath` - vracia výšku riadku nachádzajúceho sa na indexe predaného parametrom (`indexPath`). Najskôr pristúpi k objektu v poli `tableArray` na danom indexe. Následne si zistí triedu instance tohto objektu. V prípade prednášky (`Lecture`) sa vypočítava výška názvu (`titleHeight`) a výška pre zobrazenie prednášajúcich (`speakersHeight`). Inak je vrátená konštanta = 40.

```
1 | - (CGFloat)tableView:(UITableView *)tableView
   |     heightForRowAtIndexPath:(NSIndexPath *)indexPath
2 | {
3 |     id object = [tableArray objectAtIndex:indexPath.row];
4 |
5 |     if ([object isKindOfClass:[Lecture class]]) {
6 |         //height for LectureCell
7 |         Lecture *lecture = [tableArray objectAtIndex:indexPath
   |                             .row];
8 |         NSInteger numberOfSpeakers = [lecture.speakers count];
9 |         NSInteger speakersHeight = numberOfSpeakers*50;
10 |        CGFloat titleHeight = [self getTextHeight: lecture.
   |                                name];
11 |
12 |         return titleHeight + speakersHeight;
13 |     }
14 |     else {
15 |         //height for TimeCell
16 |         return 40;
17 |     }
18 | }
```

Kód 7: Výška riadku

4. `cellForRowAtIndexPath` - vracia buňku konkrétneho typu, ktorú zobrazí na riadku. Postup je podobný ako v predchádzajúcej metóde. Pristúpi k objektu na danom indexe, zistí jeho triedu a vráti adekvátne nakonfigurovanú buňku.

```
1 - (UITableViewCell *)tableView:(UITableView *)tableView
   cellForRowAtIndexPath:(NSIndexPath *)indexPath {
2
3     id object = [tableArray objectAtIndex:indexPath.row];
4
5     if ([object isKindOfClass:[Lecture class]]) {
6         //object is Lecture
7         LectureCell *cell = (LectureCell *)[self->table
           dequeueReusableCellWithIdentifier:@"LectureCell"
8         ];
9
10        return [self configureLectureCell:cell withObject:
11        object];
12    }
13    else {
14        //object is TimeString
15        TimeCell *cell = [self->table
16        dequeueReusableCellWithIdentifier:@"TimeCell"];
17
18        return [self configureTimeCell:cell withObject:object
19        ];
20    }
21 }
```

Kód 8: Konfigurácia buňky



## 5.11 Twitter wall

Vertikálne zobrazenie príspevkov z Twitteru. Riešenie je podobné ako v obrazovke Oblúbené (viď sekcia 5.10) pomocou Table View s buňkami. Pre používanie Twitter REST API som sa musel zaregistrovať a vytvoriť si aplikáciu na webe apps.twitter.com. Táto Twitter aplikácia má vygenerovaný jedinečný consumer key a secret key, ktoré sa overujú pri každom API volaní.



Obrázok 16: Twitter obrazovka

Kliknutie na buňku je registrované delegátnou metódou, v ktorej je implementovaná funkcionálna (Kód 9). Najskôr si zistí ID twitter príspevku (`tweetId`) a následne zisťuje či je v zariadení nainštalovaná oficiálna Twitter aplikácia a otvorí príspevok v nej, inak ho otvorí vo Web View. Odkazovanie sa na ďalšiu aplikáciu je možné s použitím metódy `openURL` triedy `UIApplication`. Parametrom tejto metódy je string URL adresy, ktorá má definovaný formát uvedený v dokumentácii pre vývojárov.

```
1 - (void) tableView:(UITableView *)tableView
   didSelectRowAtIndexPath:(NSIndexPath *)indexPath
2 {
3     TwitterCell *cell = (TwitterCell *)[tableView
   cellForRowAtIndexPath:indexPath];
4     if ([cell.tweetId length] > 0)
5     {
6         NSURL *openTwitterURL = [NSURL URLWithString:[
   NSString stringWithFormat:@"twitter://
   status?id=%@", cell.tweetId]];
7
8         if ([[UIApplication sharedApplication] canOpenURL:
   openTwitterURL] == YES){
9             [[UIApplication sharedApplication] openURL:
   openTwitterURL];
10        }
11        else {
12            NSURL *httpURL = [NSURL URLWithString:[NSString
   stringWithFormat:@"https://twitter.com/%@/
   status/%@", cell.tweetScreenName, cell.tweetId
   ]];
13            [[UIApplication sharedApplication] openURL:httpURL];
14        }
15    }
16 }
17 }
```

Kód 9: Kliknutie na Twitter buňku

## 6 Diskusia

Konferenciu Game Developers Session v Prahe navštívilo približne 600 ľudí, z toho 124 si stiahlo iOS aplikáciu a 98 Android aplikáciu. Celkové ohlasy boli veľmi priaznivé. Aj to je jeden z dôvodov, prečo sa bude aplikácia postupne vyvíjať, zlepšovať a pravdepodobne upravovať na ďalšie konferencie a podujatia, ktoré budú mať o tento produkt záujem.

Ďalším charakteristickým črtom, ktorým sa aplikácia odlišuje, je spomínaný harmonogram viacerých sálov. Počas jeho pomerne zložitej implementácie však nastal problém. Pri vykresľovaní väčšieho množstva sálov a prednášok dokázala aplikácia spotrebovať také množstvo operačnej pamäte, že ju bol nútený systém ukončiť. Nakoniec bolo potrebné vytvoriť nový návrh vykresľovania a takmer celú implementáciu prepísať.

Čo sa týka nedostatkov aplikácie, nároční užívatelia by mohli namietat, že obsahuje príliš málo možností oproti konkurenčným produktom. Na druhú stranu, práve obmedzením prebytočnej funkcionality bolo možné dosiahnuť požadovaný efekt jednoduchého vzhľadu a manipulácie. Napriek tomu aplikácia určite nie je dokonalá a je nutné ju stále obohacovať, preto uvediem zopár funkcií, ktoré by aplikáciu mohli vylepšiť:

- Mapa areálu - je potrebná ak sa udalosť koná vo väčších areáloch s viacerými prednáškovými miestnosťami. Podľa mňa by mohla byť interaktívna, to znamená, že by obsahovala points of interests (body záujmu) s popisom, po prípade aj možnosť zoomu.
- Notifikácie - užitočné pre rolu organizátora. V prípade zmeny programu, alebo inej dôležitej správy, ktorú by rád sprostredkoval účastníkom.
- Networking - nedeliteľnou súčasťou konferencií je hlavne spoznávanie nových ľudí pohybujúcich sa v rovnakom odbore. Nadviazanie kontaktov a spolupráce je často to najdôležitejšie, čo si účastníci z konferencie odnesú. Preto by jednoduchý spôsob výmeny kontaktných údajov priamo v aplikácii mal jednoznačne svoje využitie. Dokonca by sa dalo uvažovať aj o implementácii chatu.

Zároveň by som rád sprehladnil samotný zdrojový kód a spísal dokumentáciu aj pre ostatných vývojárov, ktorí by sa mohli podieľať na vývoji.

## 7 Záver

Na základe analýzy požiadavkov a analýzy existujúcich produktov bol vytvorený návrh databázy a storyboardu (logická a vizuálna časť) aplikácie. Pred samotnou implementáciou bolo potrebné dôkladné naštudovanie REST API umožňujúcu prácu so serverom. Aplikácia bola písaná v jazyku Objective-C vo vývojovom prostredí Xcode s použitím rôznych open source knižníc, čo je štandardom v rámci vývoja pre platformu iOS. Na komunikáciu so sociálnymi sieťami Facebook a Twitter boli integrované ich oficiálne frameworky.

Aplikácia bola testovaná na všetkých modeloch iPhonov podporujúcich verziu operačného systému 8.0. Taktiež aj rozloženie a veľkosť prvkov vykresľovaných na obrazovke bolo prispôbené pre rozlíšenia jednotlivých zariadení. Po otestovaní bola aplikácia odoslaná do spoločnosti Apple na schválenie, po ktorom bola uverejnená na App Store (online portál pre nakupovanie a sťahovanie softvérových aplikácií).

Ako je spomenuté v diskusii, aplikácia bola uvedená do praxe priamo na konferencii a vďaka pozitívnym ohlasom sa nachádza v stave kontinuálneho vývoja. Taktiež sa plánuje spolupráca s ďalšími organizátormi konferencií.

## 8 Literatúra

APPLE INC. *AppKit Framework*. [online]. [cit. 2014-10-03]. Dostupné z: [https://developer.apple.com/library/mac/documentation/Cocoa/Reference/ApplicationKit/ObjC\\_classic](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/ApplicationKit/ObjC_classic).

APPLE INC. *Core Data Programming Guide*. [online]. [cit. 2015-17-9]. Dostupné z: <https://developer.apple.com/library/tvos/documentation/Cocoa/Conceptual/CoreData>.

APPLE INC. *Foundation Framework*. [online]. [cit. 2013-22-10]. Dostupné z: [https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/ObjC\\_classic](https://developer.apple.com/library/mac/documentation/Cocoa/Reference/Foundation/ObjC_classic).

APPLE INC. *The App Life Cycle*. [online]. [cit. 2015-16-09]. Dostupné z: <https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle/TheAppLifeCycle.html>.

APPLE INC. *Core OS Layer*. [online]. [cit. 2014-17-09]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreOSLayer/CoreOSLayer.html>.

APPLE INC. *Core Services Layer*. [online]. [cit. 2014-17-09]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html>.

APPLE INC. *Media Layer*. [online]. [cit. 2014-17-09]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html>.

APPLE INC. *Cocoa Touch Layer*. [online]. [cit. 2014-17-09]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html>.

APPLE INC. *Membership*. [online]. [cit. 2016-01-01]. Dostupné z: <https://developer.apple.com/support/compare-memberships/>.

APPLE INC. *iOS Architecture*. [online]. [cit. 2014-17-09]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>.

APPLE INC. *iOS Technology Overview*. [online]. [cit. 2014-17-09]. Dostupné z: <https://developer.apple.com/library/ios/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>.

GARTNER INC. *Demand for Enterprise Mobile Apps*. [online]. [cit. 2015-20-07]. Dostupné z: <http://www.gartner.com/newsroom/id/3076817>.

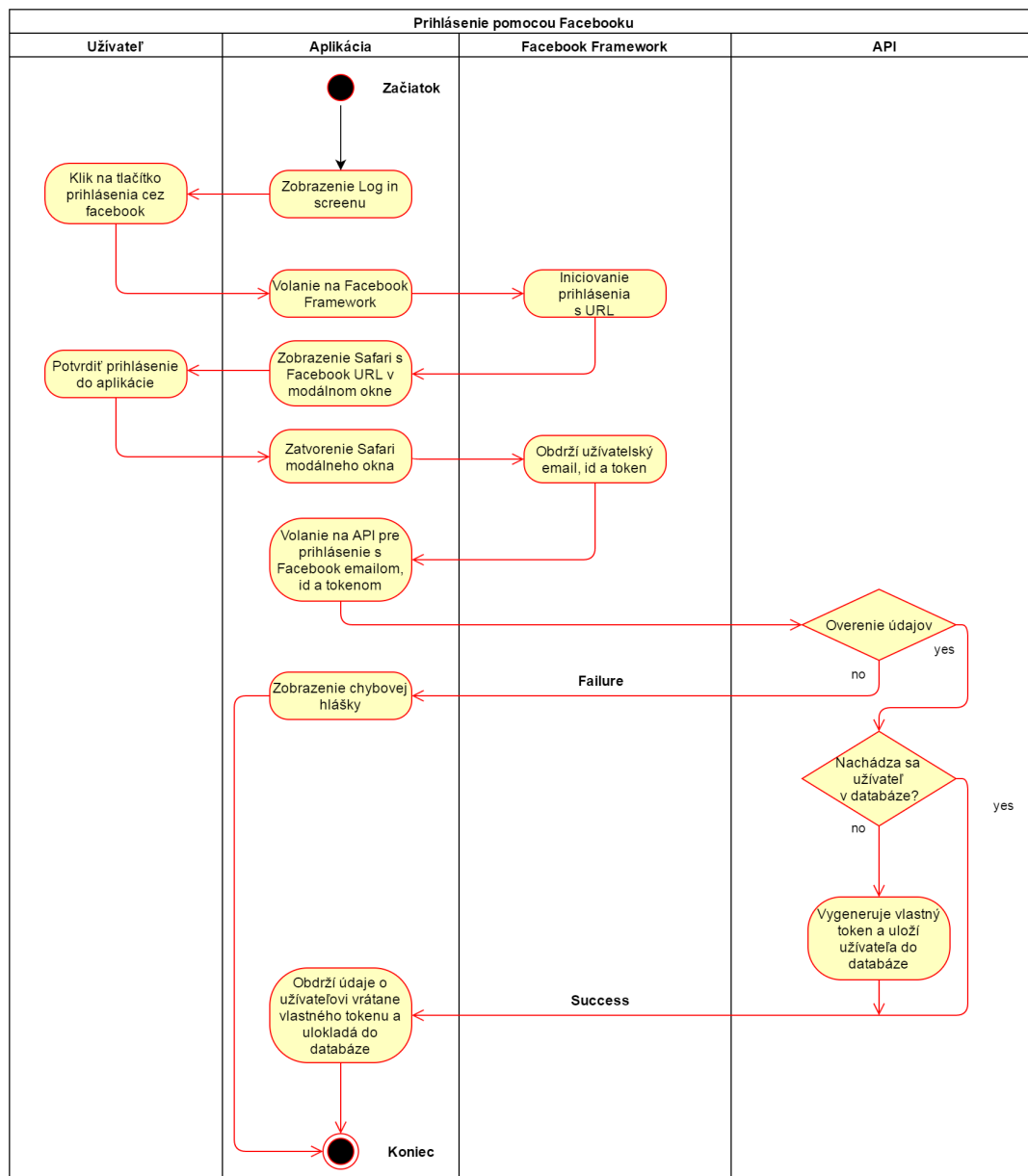
KOCHAN, S. *Objective-C 2.0: výukový kurz programování pro Mac OS X a iPhone*. Brno: Computer Press, 2010. 550 s. ISBN 978-80-251-2654-7.

NET APPLICATIONS. *Market Share*. [online]. [cit. 2016-01-3]. Dostupné z: <https://www.netmarketshare.com/operating-system-market-share.aspx?qprid=8&qpcustomd=1>.

RHIANNON WILLIAMS. *Apple iOS: a brief history*. [online]. [cit. 2017-09-17]. Dostupné z: <http://www.telegraph.co.uk/technology/apple/11068420/Apple-iOS-a-brief-history.html>.

## **Prílohy**

## A Diagram aktivít



Obrázok 17: Diagram aktivít - Prihlásenie cez Facebook



# B Storyboard



Obrazok 18: Storyboard aplikácie