

Česká zemědělská univerzita v Praze

Technická fakulta



Campus Navigator pro iPhone

Diplomová práce

Vedoucí diplomové práce: doc. Ing. Vojtěch Merunka, Ph.D.

Vypracoval: Bc. Radek Jonáš

PRAHA 2013

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra informačního inženýrství

Technická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Jonáš Radek

Informační a řídicí technika v agropotravinářském komplexu

Název práce

Campus Navigator pro iPhone

Anglický název

Campus Navigator for iPhone

Cíle práce

Vytvořte aplikaci pro iPhone, která bude uživateli poskytovat základní orientační informace o kampusu ČZU a případně další informace hodící se nováčkovi na ČZU. Aplikace nebude muset vyžadovat on-line připojení. Pokud bude možné, pokuste se implementovat detekci polohy.

Metodika

Analýza a vyhodnocení možných metod programování pro iPhone v dnešní době.

Harmonogram zpracování

1. Úvod a cíl práce
2. Literární rešerše
3. Metodika práce
4. Vlastní práce (návrh a vytvoření aplikace)
6. Závěr



Rozsah textové části

40-50 stran

Klíčová slova

Apple, iOS, iPhone, iPhone OS, Objective-C, Cocoa, Xcode, AppStore,

Doporučené zdroje informací

COSTANICH, Bryan. Developing C# apps for iPhone and iPad using MonoTouch: iOS apps development for .NET developers. New York: Apress, 2011. ISBN ISBN 978-1-4302-3174-5.

DEVOE, Jiva. Objective-C / Jiva DeVoe. Indianapolis: Wiley, 2011. Developer reference. ISBN ISBN 978-0-470-47922-3.

ISAACSON, Walter. Steve Jobs. 1st pub. United States: Simon & Schuster, 2011. ISBN ISBN 978-80-7252-352-8.

STARK, Jonathan. Building iPhone apps with HTML, CSS, and JavaScript: making App Store apps without Objective-C or Cocoa. Sebastopol: O'Reilly, 2010. ISBN ISBN 978-0-596-80578-4.

STEVENS, Chris. Designing for the iPad : building applications that sell. 1st pub. Chichester: Wiley, 2011. ISBN ISBN 978-0-470-97678-4.

WENTK, Richard. Cocoa / Richard Wentk: Developer Reference. Indianapolis: Hoboken, 2010. ISBN ISBN 978-0-470-49589-6.

MARK, Dave a Jeff LAMARCHE. iPhone SDK: průvodce vývojem aplikací pro iPhone a iPod touch. Vyd. 1. Brno: Computer Press, 2010, 480 s. ISBN 978-80-251-2820-6; 88025128206.

Vedoucí práce

Merunka Vojtěch, doc. Ing., Ph.D.

Termín odevzdání

březen 2013

Ing. Martin Pelikán, Ph.D.
Vedoucí katedry



prof. Ing. Vladimír Jurča, CSc.
Děkan fakulty

V Praze dne 28.3.2013

Čestné prohlášení

Prohlašuji, že jsem diplomovou práci na téma: „Campus Navigator pro iPhone“ vypracoval samostatně pod vedením pana doc. Ing. Vojtěcha Merunky, Ph.D. a využil jsem pouze citovaných pramenů, které jsou uvedeny v seznamu použité literatury. Informace, u kterých není uvedena citace, vycházejí z vlastních zkušeností a znalostí autora práce nebo z informací odborníků pracujících v oboru vývoje aplikací pro mobilní operační systém iOS.

V Praze dne:

.....

Bc. Radek Jonáš

Poděkování

Děkuji vedoucímu diplomové práce panu doc. Ing. Vojtěchu Merunkovi, Ph.D. za cenné rady a odbornou pomoc při zpracování této diplomové práce. Dále bych chtěl poděkovat univerzitním novinám iZUN, za poskytnuté podklady a Paulu Hegartovi ze Standfortské univerzity v USA, za jeho názorné a propracované video lekce dostupné na iTunes U.

V Praze dne:

.....

Bc. Radek Jonáš

Campus Navigator pro iPhone

diplomová práce



Abstrakt:

Obsahem této diplomové práce je seznámení s historií vzniku iPhone a mobilního operačního systému společnosti Apple Inc., popsání dosavadních verzí systému a jeho evoluce v průběhu let. Dále seznámení s konkurenčními mobilními operačními systémy a jejich vlivem na inovaci iOS samotného. V neposlední řadě pak analýza možností pro vývoj aplikací pro iOS a hlubší seznámení s programovacím prostředím Xcode s iOS SDK. Pomocí těchto nástrojů jsem se rozhodl vytvořit jednoduchou aplikaci pro nové studenty České zemědělské univerzity v Praze, která by jim pomohla s jednoduchou orientací po kampusu univerzity a která umožní podat základní potřebné informace o samotné škole. A s postupem její tvorby pak nakonec čtenáře seznámím.

Klíčová slova: Apple, iOS, iPhone, Objective-C, Cocoa, Xcode, AppStore, iOS SDK

Abstract:

The content of this diploma thesis is to introduce the history of inventing the iPhone and mobile operating system of Apple Inc., its current operating system versions and its evolution over the years. Introducing competitions mobile operation systems and their impact in iOS innovation process. Then analyze the options for programming applications under iOS & deeper introducing to Xcode programming environment with iPhone SDK. With its help I decided to make an easy application that can help to newcomers to CULS. It could help them with the basic orientation on Campus and usable informations about the university itself. Manual of making the app is the last part of thesis.

Key words: Apple, iOS, iPhone, Objective-C, Cocoa, Xcode, AppStore, iOS SDK

Obsah

1	Úvod.....	1
2	Cíl práce a metodika	3
2.1	Cíl práce	3
2.2	Metodika	3
3	Apple: cesta k mobilům	5
3.1	Převratný iPod	5
3.2	iPhone přichází	10
3.3	iPhone OS	13
3.4	iOS.....	15
3.5	Jailbreak.....	17
4	Konkurence na mobilním trhu.....	18
4.1	Google a Android.....	18
4.2	Mobilní Windows	19
4.3	BlackBerry.....	20
4.4	Ostatní systémy	21
5	Možnosti vývoje aplikací pro iOS.....	22
5.1	Úvod	22
5.2	Vývojáři .NET a jazyk C#.....	22
5.3	Kodéři HTML, CSS a JavaScriptu	24
5.4	Objective-C & Cocoa Touch	27
5.5	Xcode 4.6.....	28
6	Campus Navigator pro iPhone.....	30
6.1	Návrh aplikace	30
6.2	Xcode projekt	31
6.3	Uživatelské rozhraní	33
6.4	Úpravy vzhledu	37
6.5	Doplnění detekce polohy do kódu	39

7 Závěr	42
8 Citovaná literatura	44
9 Přehled obrázků.....	46
10 Seznam zkratk.....	46

1 Úvod

Ve své bakalářské práci jsem se hodně věnoval společnosti Apple Inc. a jejímu desktopovému operačnímu systému Mac OS X. Jsem dlouholetý uživatel Mac OS X, a proto jsem chtěl zaplout ještě více do hlubin této platformy. Seznámil jsem se základy objektově-orientovaného programovacího jazyku Objective-C, který je hlavní jazykem obou platforem Applu. Seznámil jsem se i podrobněji s vývojovým prostředím Xcode, které mě hodně zaujalo, hlavní svým intuitivní a uživatelsky přívětivým ovládáním. Už v té době jsem přemýšlel, že bych chtěl někdy v budoucnu pokračovat v objevování druhé platformy Applu, právě mobilního systému iOS.

Proto jsem si vybral diplomovou práci na toto téma. Objective-C se na naší univerzitě nevyučuje, českých publikací, které by byly aktuální se také na trhu moc nevyskytuje. Většina zde citovaných zdrojů je mnou přeložených z v původního anglického jazyku. Ke zdárnému konci mi snad pomůžou i video online lekce z amerických univerzit, kde vyučování programování pro iOS a Android je v dnešní době samozřejmostí a klade se na něj velký důraz. Jako lehce vzdělaný fanoušek Apple jsem se tedy pustil do tajů vývoje aplikací pro iOS na vlastní pěst. Chtěl jsem se o této platformě dozvědět co nejvíce, lépe jí pochopit a pokusit se jí dostat na naší univerzitě více do podvědomí.

Pokusím se vytvořit jednoduchou aplikaci pro iPhone, která by uživateli podala všechny potřebné základní informace o univerzitě a jejím kampusu. Sám z vlastní zkušenosti vím, jak problematické občas v prvním ročníku bylo nalézt některou z učeben, nebo přednáškovým sálů v tomto velkém areálu. A mít alespoň jednoduchou offline mapku s popisem, kde jaké učebny jsou by bylo hodně nápomocné. Dále vím, že některé informace o fakultách jako jsou například kontakty na studijní oddělení a jejich otevírací doby by se občas hodilo mít v mobilním telefonu bez potřeby internetového připojení. Stejně tak informace o Menze ČZU a odkazy na weby školy a studijní portály potřebné pro studium na ČZU mít všechny pohromadě by nebyl špatný nápad. Pokusím se i implementovat určení polohy pomocí GPS, kdy už ale bude potřeba připojení k internetu pro načtení mapových podkladů. Případná distribuce přes Apple App Store není bez zaplacení vývojářského ročního poplatku a registrace u Applu možná, tudíž by zatím tato aplikace sloužila jako ukázka nápadu a mě osobně by sloužila jako hlubší prohloubení znalostí fungování iOS.

Nakonec mé práce budu popisovat pomocí screenshotů z mého počítače celou realizaci tvorby mé, a jím podobných aplikací. Chtěl bych jí podat co možná nejsrozumitelněji a polopaticky. Doufám, že se díky této diplomové práci ještě více rozumím celému ekosystému Apple. Osobně vidím v této platformě velký potenciál a myslím si, že v budoucnu mobilní systémy budou těmi

primárně využívanými operačními systémy a desktopové postupně budou ustupovat do pozadí a budou čím dál tím více používány jen pro specifické operace, které nebudou dostatečně pohodlné na řešení na mobilním zařízení. Proto si myslím, že rozšíření rozhledu na univerzitách o tomto odvětví, v podobě vzdělávání v programování pro iOS a Android, je do budoucna velice důležitá věc. Byl bych moc rád, kdyby moje práce podpořila myšlenku vyučování programovacího jazyku Objective-C a vývoje aplikací pro iOS na ČZU. Doufám že se práce bude líbit a bude dostatečně srozumitelná.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je seznámit čtenáře s mobilním operačním systémem společnosti Apple, jeho historií a následnou evolucí. Dostat ho trochu více do podvědomí. Seznámit ho s konkurečními mobilními operačními systémy na trhu a jejich historií na trhu. Analyzovat možnosti programování pro iOS pro různé programovací jazyky. Představit čtenáři novou verzi programovacího prostředí Xcode a soubor nástrojů pro vývoj aplikací iOS SDK. A nakonec se s jejich pomocí pokusit vytvořit jednoduchou aplikaci pro základní orientaci po kampusu ČZU a popsat postup její tvorby, pokud možno srozumitelnou cestou od návrhu až po testování.

2.2 Metodika

Metodika, kterou se budu řídit v této práci vypadá následovně. Vývoj jakékoliv věci, programu nebo aplikace, má své jasné postupy. Pokud si vyberu nějakou platformu, je dobré o ní vědět její historii a seznámit se s její konkurencí na trhu. Pokud i nadále této platformě zůstanu věrný, je třeba zamyslet se nad tím, jestli aplikace, kterou jsem si vybral bude mít potenciální zákazníky. Jestli má naději zaujmout svými funkcemi, podanými informacemi, nějakou cílenou skupinu lidí, tedy zákazníků. Dále je důležité si zjistit, jestli náhodou podobné aplikace již dávno nejsou na trhu. Pokud ano, je na snadě analyzovat svojí budoucí konkurenci, jejich silné a slabé stránky, porovnat je se svými vlastními a vyhodnotit, jestli moje plánovaná aplikace může být konkurenceschopná. Pokud vyhodnotím, že aplikace může být konkurenceschopná a rozhodnu se projekt realizovat, je pro mě důležité rozhodnout se, v jakém programovacím jazyce ji budu vyvíjet. Pro každou platformu existuje víc možností programování, záleží na našich předchozích znalostech, nebo osobních preferencích. Nejlepší je umět nativní programovací jazyk, ale pokud nám nesedí, nebo jsme dobří v nějakém jiném, který je platformou a prostředky pro vývoj podporovaný, nic nám nebrání v tom, vybrat si jiný, než je ten nativní a šitý platformě tzv. „na míru“. Až se rozhodnu pro programovací jazyk, vybereme si některé z vývojových prostředí, pokud jich máme na výběr více, které tento jazyk podporuje a nám bude vyhovovat. Pokud se rozhodneme, nestojí nám nic v cestě začít s návrhem.

Nejlepší je na začátek dostat nápad na papír. Jednoduchým sketchem aplikace a jejích vlastností můžeme narazit na možné problémy, nebo možná vylepšení, také nám tento náčrtek může pomoci v budoucím návrhu vzhledu aplikace ve vývojovém prostředí. Následuje shromáždění

potřebných souborů pro aplikaci, například vytvoření ikon aplikace, pozadí aplikace, případné audio soubory, pokud je budem používat. Následuje spustění vývojového prostředí a tvorba aplikace. Základem bývá nejdřív funkcionality aplikace, a základní sestavení, až máme toto hotové, přecházíme k úpravě vzhledu a estetické stránky aplikace. V neposlední řadě je testování a případné opravování a řešení chyb. Jednoduché menší úpravy funkcionality, designových nedostatků a všelijakých neduhů. Pokud je aplikace hotova, testována a plně funkční, je třeba s ní podstoupit schvalovací proces například Applem, nebo Googlem. Pokud náš kód a aplikace celková projde schválením, nestojí nám už nic v distribuci přes jejich distribuční kanály.

3 Apple: cesta k mobilům

3.1 Převratný iPod

Když se v roce 1997 vrátil Steve Jobs do čela Apple jako prozatímní ředitel, měl Apple širokou nabídku produktů, které prodával. Až příliš širokou, usoudil Jobs – zákazníkům dělalo problémy se v ní zorientovat. Řada produktů se lišila kombinacemi pamětí, velikostí a provedením skříně, taktováním procesoru. Jen fanouškové měli chuť rozlišovat rodinu Power Macintoshů podle čísel. Sám Jobs později uvedl, že i jemu museli lidé z produktové divize tři týdny vysvětlovat, co se pod kterým číslem skrývá a pro koho je ten produkt určen. Sice svým způsobem přeháněl, ale v tom zásadním měl pravdu: pro zákazníka bylo těžké si v nabídce Apple vybrat počítač. Ještě v roce 1997 zastavil Jobs produkci dvou třetin modelů, k lítosti fanoušků včetně osobního digitálního asistenta jménem Newton. [1]

Jobs dlouhodobě zastával stanovisko, než dát na trh produkt, který je nedokonalý v některém z klíčových parametrů, je lepší jej nenabízet. Sekyra se nad hlavou produktu Newton mihla hned počátkem roku 1997, kdy Jobs jeho divizi omezil, koncem roku ale seznal, že ačkoliv je produkt slibný, Apple momentálně nemá sílu jej dotlačit do stavu, kdy by s ním firma mohla být spokojená. A Apple se potřeboval soustředit na hlavní příjmové segmenty a vrátit se do zisku. Za fiskální rok 1997 přišel Apple o více jak miliardu dolarů a Jobs později připustil, že firma byla tři měsíce před bankrotem. Nešlo si dovolit sentimentální lpění na produktu, jakkoliv byl převratný a fanoušky dobře přijímaný, když nevydělává a potřebuje zásadní restrukturalizaci. [1]

Jobs se rozhodl, že produkty Apple musí být rozdělitelné do jednoduché matice a jako takové se budou i marketovat a prodávat. Do této doby vládla v Apple metoda reakce na požadavky obchodníků: obchodníci hlásili, a technici úpravou stávajícího modelu vytvořili novou modelovou řadu. Tak vznikala fragmentace, nepřehlednost, která nakonec neprospívala ani prodejm, protože požadavku několika prodejců sice sestava vyhovovala, ale ostatní jí nevěnovali pozornost. Jobsovo rozhodnutí mělo v první řadě produkty zpřehlednit, ale následně přineslo i další kýžený efekt: lepší práci se skladovými zásobami, schopnost rychleji reagovat na požadavky, protože možností. Jobsova matice produktů stanovovala dvě hlavní kategorie počítačů: Zákaznické a Profesionální, přičemž měly existovat jejich dvě varianty, Desktop a Přenosný. Tedy celkem čtyři produkty. Z toho, že v roce 1997 Apple představil patnáct různých počítačů a řadu dalších vyráběl ještě z modelových řad předchozích roků, je zřejmé, že šlo o zásadní řez. Profesionální řada počítačů napříště vycházela z Power Macintosh G3, v přenosných počítačích se Apple soustředí na PowerBook G3, ale v oblasti počítačů pro koncové zákazníky neměl Apple co nabídnout.

Přitom právě v segmentu koncových zákazníků poptávka vzrůstala a bylo zde možno lehce realizovat Jobsovu vizi „digitálního hubu“. Bylo potřeba začít hledat, konstruovat. V roce 1997 totiž Apple neprodával žádný počítač za méně než 2 000 dolarů a to bylo pro segment koncových uživatelů příliš. Bylo potřeba tuto hranici stlačit níže, ideálně k tisícidolarové hranici. Nakonec padlo rozhodnutí začínat u nejlevnějších počítačů s cenou 1 200 dolarů. [1]

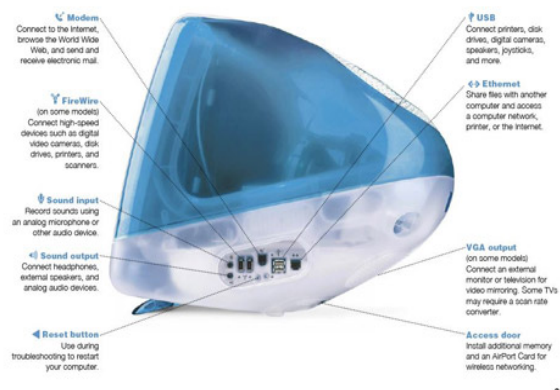
Jobs si představoval počítač typu „all in one“. Nenáviděl propojování jednotlivých prvků počítače dráty a v minulosti Apple mnohokrát zabodoval konstrukcí, kde zobrazovací jednotku vestavěl do samotného těla počítače. A tak se konstrukční tým vrátil do roku 1984, kdy vznikl původní Macintosh, počítač, který měl monitor kombinovaný se samotným počítačem. Běžný uživatel, soudil konstrukční tým, nebude chtít počítač demontovat, aby provedl nějaký upgrade. Spíše ocení, že počítač bude fungovat i bez jeho větší investice a propojování kabelů, hned po vybalení. To byl první předpoklad, který vedl k modelu all-in-one. Díky tomuto předpokladu se ale také prosadilo nové rozhraní, v té době v počítačích pro běžné uživatele nevídané: USB. Standard USB byl velmi čerstvý, teprve počátkem roku 1996 se objevila jeho finální verze 1.0, která byla konečně použitelná a dovozovala přenosovou rychlost až 12 Mb/s, záhy ji následovala verze 1.1, která se měla rozšířit a po níž měl sáhnout i Apple. [1]

Apple se rozhodl novým rozhraním nahradit dosavadní směsici sběrnicí zahrnující SCSI či firemní ADB. A iMac (viz. Obr. 3-1; 3-2) se stal prvním počítačem, který USB používal, velmi významně přispěl k jeho rozmachu a z jeho rozmachu také profitoval.

Obr. 3-1: iMac



Obr. 3-2: iMac rozhraní



Zdroj: photo.net

Spolu s osekáním přebytečných rozhraní přišel šéf hardwarové divize Jon Rubinstein spolu s Jobsem na myšlenku vypustit z počítače disketovou jednotku, v té době neodmyslitelnou součást každého počítače. Jenže v tu dobu používané tříapůlpalcové disketové mechaniky měly standardní kapacitu 1,44 MB a to bylo příliš málo. Nový iMac dostal namísto disketové mechaniky standardně vestavěný modem o rychlosti 56 kb/s a počítalo se s tím, že uživatelé si menší soubory přenesou, jak byli zvyklí, po internetu. Pokud by disketovou jednotku vyloženě potřebovali, mohli si dokoupit její verzi s USB. Radikální krok byl později při představení iMacu kritizován jako příliš radikální, řada komentátorů a recenzentů se nemohla smířit s tím, že na novém iMacu nebudou schopni své soubory zapsat na nic, co by mohli jako fyzické médium přenést na jiný počítač. Praxe ale ukázala, že uživatelé opravdu používali pro přenosy běžných souborů více internet. [1]

Steve Jobs představuje nový iMac 6. května 1998 ve Flintovu auditoru v De Anza Community College v Cupertino, tedy na stejném místě, kde byl představen původní Macintosh. Firemní legenda praví, že se iMac dočkal famózního přijetí. Což byla do značné míry pravda, počítač měl velmi kladné recenze a i mezi fanoušky převládalo nadšení, že se Apple konečně našel a Jobs je zpět. Když se Jobs v roce 1999 rozhodl s iMacem orientovat na zpracování amatérského videa, chtěl po Adobe, aby připravilo balík nástrojů na stříh videa pro Mac OS. Byl ale odmítnut, což byla jedna ze součástí budoucí roztržky s Adobe, která o dalších deset let později vedla k pádu ambicí Adobe proniknout s Flashem do mobilů. Jobs se cítil dotčen oprávněně, kdyby nebylo Apple, nebylo by Adobe se svými softwarovými balíky tam, kde bylo. Jenže to se nepočítá, Apple tedy začal chystat v roce 1999 vlastní balík software, který měl z počítačů iMac učinit domácí digitální hub. [1]

Na přelomu tisíciletí se objevují první MP3 přehrávače, nejrozšířenější jsou MP3 discmany, přehrávače CD, které umí kromě běžných audio CD přehrát i CD s vypálenou hudbou v MP3. Výhoda je jasná, mají kapacitu cca 700 MB dat, tedy zhruba deset alb. V tu dobu nejlepší flash MP3 přehrávače mají paměť 64 MB, tedy pojmu stěží jedno album, stojí ale několiknásobně více. Všechny typy dosavadních MP3 zařízení spojují dvě vlastnosti: jsou mizerně ovladatelná a nahrát do nich hudbu je zdlouhavé. V zásadě platilo, že jste hudbu do přehrávače přánášeli tak dlouho, jak dlouho sama trvala. Tím se MP3 přehrávače omezovali spíše na fanoušky technologií, než fanoušky hudby. Flashové přehrávače nedovolovaly vybírat hudbu a vešlo se do nich jedno album. Diskové zas na druhou stranu byly moc velké a aktivnějším pohybem se přehrávání zdrhovávalo. Obecně bylo zřejmé, že zájem na trhu je, jde jen o to, vyřešit uživatelské problémy. Apple si v roce 2000 díky Napsteru, síti pro výměnu a stahování hudby, všiml, že hudební pole čeká boom, kterého se zatím nikdo pořádně nechopil a bohužel však nebylo ani zcela jasné, jak jej uchopit. Bylo jasné, že lidé budou chtít hudbu do svých přehrávačů stahovat, šlo jen o to,

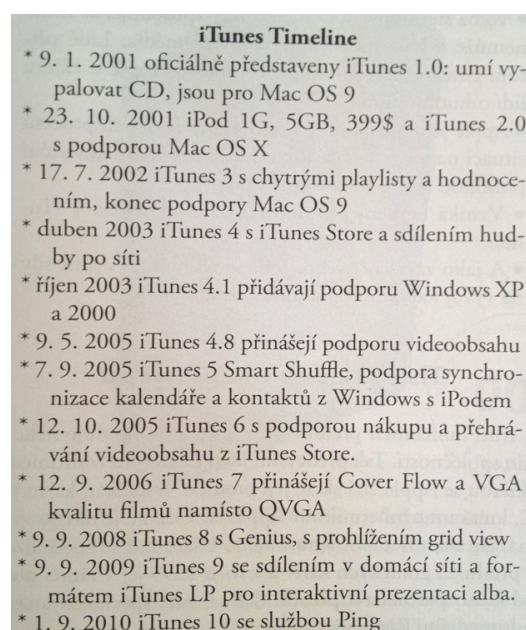
nabídnout ucelenou alternativu, která umožní bezproblémově dodávat hudbu až do přehrávače uživatele. To vše se lehkou řečnou, ale v roce 2000 se to už tak lehkou udělat nedalo. V té době Apple neměl ani pořádný hudební přehrávač pro své Mac. Quicktime mohl sloužit pro jednoduché přehrávání, ale obsloužit, nebo snad zpracovat a katalogizovat obsáhlejší sbírku hudby neuměl a Jobs rychle pochopil, že než se vydávat nejistou cestou interního vývoje, bude lepší koupit někoho na trhu, kdo už tuší, jak má pořádný hudební přehrávač vypadat. [1;2]

Na trhu existovala trojice slušných přehrávačů hudby pro Mac OS X: Audion od Panic, MacAmp a SoundJam MP. Jobs měl nejdříve zájem o Audio, ti však vyjednávali s firmou AOL o použití jejich kódu, a z tohoto důvodu schůzku o možném společném jednání odmítli. Proto Apple začal vyjednávat se SoundJamem, jeho spoluzakladatel dříve pracoval v Applu a byli stále v kontaktu. V roce 2001 přichází Apple s první verzí iTunes, postavené na základě SoundJamu. Jobs se v té době zachoval překvapivě: koupil za nemalé peníze firmu především pro její lidi a její produkt od základů a pod velkým tlakem předělal. Ostatní společnosti, například Microsoft, nekupovali v té době produkty, tedy něco, o čem se domnívaly, že mohou v podstatě beze změny vše začlenit do své nabídky a profitovat. Bylo těžko představitelné, že by za tyto peníze koupila tým a jejich produkt vyhodila a představila nový. Tato operativnost právě znamenala pro Apple zásadní konkurenční výhodu. A to vše Applu pomohlo zaútočit na trh hudebních přehrávačů a hudebního průmyslu. Níže lze spatřit iTunes 1.0 s původním SoundJam (viz. Obr. 3-3) a vývoj iTunes v průběhu let (viz. Obr. 3-4). [1]

Obr. 3-3: iTunes 1.0 a SoundJam



Obr. 3-4: iTunes Timeline



Zdroje: macnn.com ; [1]

V té době jsou na trhu tři hlavní zájemci o post úspěšného MP3 přehrávače. Je jím společnost Rio, Creative Labs a RCA Lyra (Thompson konsorcium). Ani jeden z výrobců nebyl zrovna vedoucí společností v audiotechnice, špiše výrobci počítačových komponent, kteří považovali MP3 za zajímavou a relativně jednoduše zkonstruktovatelnou věc, která otevírá dveře do nového segmentu. Postupně se trh začal zaplňovat asijskými společnostmi. Začaly nastupovat i přehrávače s pevným diskem. Právě Personal JukeBox 100 (PJB-100), který obsahoval ve své druhé verzi pevný disk s kapacitou 20 GB, významně posloužil v mnoha ohledech jako inspirace při konstrukci prvního iPod. Apple tak nevstupoval do neznámé země. Existovaly už úspěšné přehrávače, zajímavá řešení, ale těch horších, ze kterých si mohl vzít ponaučení. Apple vycházel z prací svých předchůdců, za což se s Creative Labs dostal až k soudu. Jobs v roce 2000 s pomocí svých kolegů analyzoval konkurenci, výdrž baterie, kde zjistil, že PJB-100 díky úspoře energie, které dosahuje díky vypínání pevného disku, když je skladba načtena do vyrovnávací paměti, je schopen přehrávat až 10 hodin. Používal standardní 2,5" pevný disk od Toshiba do notebooků, Jobs by ale raději viděl ještě menší a lehčí disk. [1;3]

Zástupci Toshiba ukazují roku 2001 prototyp nového disku o velikosti 1,8" a kapacitě 5 GB. Toshiba ale nemá pro tak malý disk uplatnění a nechce ho zavádět do výroby bez vize velkých odběrů. Apple tak využívá své hrubé finanční síly a s Toshiba uzavírá dohodu, kdy bude mít veškerou výrobní kapacitu těchto disků a Toshiba zase na oplátku zálohu 10 milionů dolarů. Tím Apple může inicializovat výrobu součástí, které by jinak na trh přišly mnohem později a za vyšší ceny. Této technologické a cenové exkluzivity od té doby v budoucnu bude využívat stále častěji. Apple si zajistil harwarové řešení s firmou PortalPlayer, která měla zkušenosti s přehrávači s pevnými disky a řešením problémů s vyrovnávací pamětí, kterou konkurenci tížilo nejvíce. Dohromady použili tak větší vyrovnávací paměť, která dovolila přehrávat 20 minut bez spuštění disku a tím šetřit energii a zároveň tím Apple vypracoval systém predikcí toho co bude chtít uživatel slyšet, na základě toho, které skladby přeskakoval. Dohodou s PortalPlayer a firmou Pixa, která holdovala systému umožňující rychlý návrh a vývoj vestavěného operačního systému, který potřeboval, tak měl náhle Apple jak softwarové tak hardwarové řešení prakticky ihned k použití. Zbávalo už jen dodělat maličkosti: design a ovládání. Obojí mělo vycházet ze zavedených hodnot Applu, tedy maximální jednoduchosti. Jobs zakázal tlačítko pro vypínání, iPod se měl vypínat sám. Další zvláštností bylo, že nešlo hudbu z iPodu přenést na jiný počítač. Jobs nechtěl, aby jeho zařízení jakkoliv podporovalo krádeň hudby. Design dostal pod své ruce Jonathan Ive, který je v dnešní době hlavním designérem Applu. Design byl tak minimalistický a elegantní. [1;3]

Steve Jobs představil první iPod 23. října 2001 na produktové prezentaci, k níž zval pozvánkou napovídající „a nebude to iMac“. Nebyl. Jobs na pódiu poplácal svojí kapsu u džín a prohlásil: „Mám ho tady v kapse,“ „Tisíc skladeb se mi vejde do kapsy,“ pokračoval a pozvané novináře zcela ohromil. Následovaly různé, až diametrálně odlišné odezvy ze širšího spektra lidí. První verze iPodu se fenoménem nestala, prodávala se do poloviny roku 2002, druhá verze s dotykovým ovládacím kolečkem, kterým jsou dodnes iPody známé konečně přinesla Applu vytoužené ovace. Od té doby Apple spustil hudební obchod iTunes Store. Jobsovi se podařilo po delším řešení dotáhnout iTunes Store k funkčnímu a hlavně právně ochráněnému stavu. Každý rok pak uváděl nové verze iPodů a jejich velikostní odnože, které s nástupem větších flash pamětí mohli být menších rozměrů než hlavní iPod s pevným diskem. Díky promyšleným a cíleným reklamám se tak iPody dostávali více a víc do podvědomí zákazníků a do roku 2004 zažívaly velký boom. Vlevo první verze iPod s pevným diskem (viz. Obr. 3-5) a vpravo druhá generace s dotykovým ovládacím kolečkem (viz. Obr. 3-6).[1]

Obr. 3-5: Původní iPod



Obr. 3-6: iPod verze 2



Zdroj: <http://www.apple-history.com/ipod>

3.2 iPhone přichází

S nástupem mobilních telefonů s integrovanými přehrávači MP3 se však začalo všechno v roce 2004 měnit. Pokles prodeje iPodů a MP3 přehrávačů obecně dohnal Apple k razantnímu jednání. O svém vlastním řešení a vizi mobilního telefonu už průběžně přemýšlel, dokonce doménu iPhone.org založil už roku 1999. Vytvořit vlastním řešením a prorazit s ním do vod mobilních telefonů však v té době pro Apple nebylo finančně možné. Musel se tedy unáhlit ke spolupráci. Vznikl tak totální kiks v podobě Motorola ROKR. Zpočátku zajímavý nápad: spolupracovat na uvedení hudby do mobilního telefonu s lídrem oboru. Šéf Motoroly Ed Zander Jobse seznámil

s tehdy neuveřejněným prototypem RAZR. Jobsovi se koncept líbil, Zander ho tak zasvětil do tajů mobilních telekomunikací. Pro Jobse to byl zcela nový svět. A tak se dohodl se Zanderem na spolupráci: Apple zpřístupní Motorole iTunes pro její telefony, Motorola vyrobí „iPod v mobilu“ a zajistí partnerství s operátory. Dohoda však měla hlavní výhodu pro Zandera, který se tím chtěl ukázat akcionářům a tuto informaci použil v tiskovém prohlášení. Tím však Jobse rozzlobil, protože prozradil, co se připravuje a vyfoukl mu tak jeho vlastní moment překvapení. Motorola stále měnila své názory, nakonec použila i jiný verzi mobilu, protože zjistila, že tato nebude ta pravá. Oficiální výsledek spolupráce byl představen 7. Zářím 2005, jemnoval se Motorola ROKR E1 a měl pracovat s novými iTunes 5. Ve stejnou dobu Steve ohlásil, že nové iPody Mini půjdou cestou flash pamětí. Motorola ROKR se dočkala příšerného přijetí. Časopis Wired na se své obálce dokonce posměšně ptal „*Tomu říkáte telefon budoucnosti?*“, v podobném duchu se nesly i další reference. ROKR byla narychlo stvořená splácanina, o vinu se dělil shodně Apple s Motorolou. Oproti předchozímu mobilu firmy měl ROKR bílé provedení a tlačítko pro spuštění přehrávače. To byly jediné změny co Motorola provedla. Apple se ale také vyznamenal, na poslední chvíli donutil Motorolu omezit maximální počet skladeb na stovku, to bylo oproti původním 25 skladbám, co Jobs chtěl ještě „velkorysý ústupek“. Apple se bál, že by ROKR mohl konkurovat jeho iPodům, byl totiž vybaven 512 MB kartou, ale hudba byla omezena na 100 skladeb. Ještě směšnější ale bylo, že v mobilu byl přehrávač s podporou Javy, který dokázal přehrávat neomezený počet skladeb, ale ne ty zabezpečené Apple. S dalším omezením přišel i operátor Cingular, prodejní partner Motoroly. Ten zakázal používat skladby z iTunes jako vyzvánění, bál se totiž, že přijde o své zisky z prodeje svých vyzváněcích tónů. Všechny tři firmy na sebe navzájem svalovaly vinu za neúspěch jejich „experimentu“. [1;2]

Partnerství s Motorolou Jobsovi ukázalo, jak jsou telekomunikační společnosti těžkopádné, přes všechny proklamace málo progresivní a vyvíjet s nimi produkt je něco, co psychicky nezvládá. Také zjistil, že je důležité nadchnout pro takový produkt operátora a donutit ho, aby umožnil zákazníkům nabízet určité služby, ze kterých by i Apple mohl mít provizi. A toto zjištění mělo pro budoucnost iPhone a telekomunikací dalekosáhlé důsledky. Vyrobit vlastní mobil v roce 2005 nebylo těžké. Steve dokonce s asijskými továrnami spolupracoval, ale on chtěl především spojit telefon s iPodem. Apple již v té době počítal se spuštěním videa v iTunes Store, takže potřeboval, aby jeho zařízení mohlo komfortně přehrávat i video. Klávesnice přes půlku zařízení, kterou měli všichni výrobci takové potřebě nevyhovovala. Nejlepší volbou se tedy ukázalo dotykové ovládání, a mít tak k dispozici pro zobrazování maximum plochy. Nápověda k ovládacím prvkům by se měnila podle potřeby přímo na displeji, což klasická klávesnice nedovolovala. Takový nápad se už dříve používal, ale nikdo ho nedokázal dotáhnout do konce. Existovalo několik technologií, jak

odečítat z displeje bod stisku. Nejvíce se používala rezistivní (odporová). Tyto displeje se skládali z několika vrstev, přičemž přitlačení vrstev k sobě značilo stisk. Displeje nebyly drahé, ale nebyly také nejspolehlivější, postupem času ztráceli kvaitu určení místa stisku, zkreslovaly obraz a barvy, bylo potřeba stylusu, ostrého hrotu pro jasné určení body stisku. Nešlo ovládat dvěma prsty, ani označovat simultánně více bodů. Takové displeje používala celá řada zařízení od Palm, ale i Windows Mobile zařízení. Bylo třeba se porozhlédnout jinde.[1]

Apple nakonec našel společnost FingerWorks, ta se zbývala dotykových plošek iGesture pro ovládání počítačů a přístrojů, které dokázaly poznat dotyk na více místech a vyhodnotit je jako rozdílné příkazy. Měli zkušenosti s konstrukcemi ovládanými dotykem, ale také měli vyvinutou celou řadu ovládání formou gest. Apple společnost v tichosti koupil se všemi zaměstnanci i patenty. Apple tak mohl urychleně dokončit konstrukci jiného typu displeje schopného rozpoznat dotyk. Kapacitní displej dokázal rozpoznat dotyk změnou kapacity vodivé vrstvy, stačilo tak jen přiložit prst na displej. Zachovával věrné barvy, bylo ho možné ovládat více prsty a konstrukčně byly spolehlivější. Jen je nikdo nevyráběl. Opět se ale vyplatilo použití volných finančních zdrojů v hotovosti a zaplacení továrně za to, že uvede jejich displej do výroby exklusivně pro ně. Apple musel ještě vyřešit ovládání prsty, hlavně psaní na klávesnici. Vyřešili to tak, že nepočítali jen střed oblasti dotyku, ale i místo kde začal, místo přiblížení a oddálení od displeje. Poté doladil i další neduhy s novým řešením psaní na klávesnici. Apple virtuální klávesnici nevynalezl, ale byl tím, kdo ji učinil použitelnou pro psaní prsty. Apple měl tedy displej, který se ovládal výborně prsty, ale chtěl ještě jednu změnu od dosavadních řešení. Do té doby byly displeje kryty průhlednou plastovou deskou. Byly ohebné, nelámaly se v kapse, ale lehko se odíraly a poškrábaly. Podle jobse by sklo dělalo lepší dojem. Sehnat ale takové sklo nebylo jednoduché. Nakonec se podařilo najít dodavatele, společnost Corning měla sklo, bez využití. Říkali mu gorilí, bylo neuvěřitelně odolné a přesně vystihovalo potřeby Jobse. Využil tedy opět finančních prostředků a domluvil s Corningem, aby mu vyrobil dostatek kusů. iPhone se stal prvním mobilem, který používal sklo místo plastu na krytí displeje. Jobs 9. ledna 2007 vstoupil na pódium sanfranciského MacWorld Expa a po úvodních shrnutích úspěchů a revolucí, které Apple od počátku své existence dokázal, oznámil, že má 3 nové revoluční produkty. iPod se širokoúhlým dotykovým displejem, revoluční mobilní telefon a internetový mobilní komunikátor. S každým produktem dostal odpověď v podobě potlesku, ale lidé stále nechápali, když pak uvedl na pravou míru, že tyto produkty jsou vlastně součástí jednoho celku, všichni byli v naprostém údivu. Zpočátku nikdo netušil, jestli taková věc určitě může uspět, ale čas ukázal, že heslo provázející iPhone „*redefining mobile phones*“ není zas tak daleko pravdě. V dnešní době v podstatě všechny dotykové mobily vděčí iPhone za jejich existenci. V dnešní době je na trhu již 6. generace iPhone.

Na obrázku níže je vidět první iPhone 2G v ruce Steva Jobse při jeho představení (viz. Obr. 3-7) a na druhém je zobrazen poslední generace iPhone 5 ve srovnání s první generací (viz. Obr. 3-8). [1;2]

Obr. 3-7: Představení iPhone 2G



Obr. 3-8: iPhone 2G vs. iPhone 5



Zdroje: fewrandomrantings.wordpress.com ; todaysiphone.com

3.3 iPhone OS

Pro úspěch mobilního telefonu bylo důležité i to, jaký bude mít operační systém. V té době se nejvíce prodávaly telefony s jednoúčelovým firmware, kdežto „chytré“ telefony nebyly moc oblíbené. Jobs ale chtěl od svého telefonu značnou možnost budoucího rozšiřování a flexibilitu ve vývoji. A také pokud možno výbornou kompatibilitu s platformou Mac. Báł se, že vývojem dalšího systému se firma zahltí. Rozhodl se adaptovat svůj operační systém Mac OS X pro svůj mobilní telefon. Neměl příliš mnoho jiných voleb a možná budoucí konvergence zařízení přispěla k tomuto kroku. K urychlení vývoje také vznikly dva nezávislé týmy. Hardwarový měl za úkol rychle zkonstruovat samotný mobil, druhý se pak měl zaměřit na adaptaci operačního systému OS X. Základním jádrem systému se stal Darwin vytvořený firmou NeXTSTEP, kterou založil Jobs, když byl vyhozen z Apple v polovině 90. let minulého století. Operační systém iPhone má čtyři abstraktní vrstvy: Jádro operačního systému, vrstvu Core Services, vrstvu Média a vrstvu Cocoa touch. Přitom vrstva Cocoa touch je postavena nadevšemi ostatními. Operační systém musel prioritně vyhodnocovat uživatelskou volbu. Apple se nijak moc nezabýval možností spouštět aplikace třetích stran v iPhone. Systém podporoval preemtivní multitasking, ochranu paměti a další pokročilé funkce moderních operačních systémů, což bylo v kontrastu s konkurenčními systémy. Apple pracoval celou dobu v utajení, tudíž hledání nových vývojářů nebylo jednoduché. Například testovací skupina byla často jen virtuální, kontaktní byli pouze e-mailem a dlouho

dobu ani nevěděli, že pracují pro Apple. Velkým zásadním „neduhem“ pro široké okolí byla absence Flash od Adobe. To mělo za následek zamezení přístupu zajímavému videoobsahu, zejména na YouTube. Odhadovalo se, že Apple přijde s vlastní verzí Flashe. Místo toho se Apple dohodl s Google na tom, že videa v Youtube se překódují do formátu H.264 pro mobilní telefony. Flash je velice rozšířený, ale patří také mezi nejmizernější moderní formáty pro přehrávání videa, video dekóduje přímo procesor zařízení, tudíž je náročný na napájení a má velký vliv na výdrž baterie mobilních zařízení. Jobs viděl totiž velký potenciál v novém HTML 5. S postupem času se v aktualizacích se přidala možnost cizích klavesnic, možnost více-uživatelských SMS, upravování Home screen a možnost více ploch. Apple pak vypustil začátkem roku 2008 do světa iPhone SDK (dnes již iOS SDK), nástroje pro tvorbu aplikací třetích stran. [1;3;4]

Druhá verze operačního systému s označením iPhone OS 2 přišla v roce 2008 s novým iPhone 3G. Podporoval GPS a 3G síť, pokud byl hardware přítomen. S novou verzí systému přišla možnost instalovat aplikace a hry třetích stran jak do iPhone, tak iPodu Touch. Aplikace a hry byly odtud dostupné v novém obchodě App Store. Dále podpora Microsoft Exchange. V následujících aktualizacích se kromě oprav také objevilo zrychlení instalace aplikací, rychlejší načítání a hledání v kontaktech, lepší výdrž baterie, rychlejší záloha do iTunes a zlepšení funkčnosti mailového klienta. V posledních aktualizacích přibylo i Google Street View, již se dala vypnout automatická oprava pravopisu a nově dvojitě zmáčknutí Home tlačítka zobrazila první plochu iPhone. [4;5;6]

iPhone OS 3 přišel s vylepšenou verzí iPhone 3GS v roce 2009. Přinesl zásadní vylepšení v podobě funkce “Copy & Paste“, která hodně uživatelům doposud chyběla, Apple se ale bránil, že nechtěl tuto funkci vypustit, dokud nevymyslí uživatelsky přívětivou implementaci. Nově se dali již posílat MMS a objevilo se vyhledávání spotlight známé z Mac OS X. Další novinkou byla aplikace Voice Memos pro nahrávání hlasových stop. Dále automatické doplnování a vyplňování údajů v Safari a blokování nechtěného obsahu stránek. V neposlední řadě Shake-to-shuffle vlastnost u přehrávače hudby a podpora 30 nových světových jazyků. Ve svých aktualizacích přinesl zásadní opravy bezpečnosti a funkčnosti systému. Poslední verze systému pod označením iPhone OS byla verze 3.2, která přinesla nativní podporu nového výrobku iPad. Ukázka první verze iPhone OS (viz. Obr. 3-9) a poslední verze označovaná jako iPhone OS (viz. Obr. 3-10). [4;5;6]

Obr. 3-9: iPhone OS 1



Zdroj: [3]

Obr. 3-10: iPhone OS 3.2

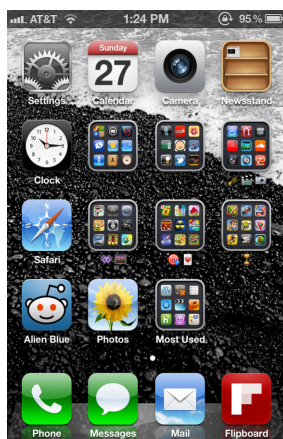


Zdroj: [3]

3.4 iOS

Apple se v roce 2010 na základě toho, že mezi jeho přenosné mobilní zařízení již v té době patřil iPhone, iPod touch a nově i tablet iPad, rozhodl přejmenovat svůj mobilní operační systém na iOS. Pro uživatele ale i širší veřejnost měl být tento název zajímavější a dával smysl pro budoucí záběr nejen v mobilních telefonech. Nová verze systému která přišla s iPhone 4 se tedy již jmenovala iOS 4 a přinesla oproti předchůdci znatelné změny. Uživatelé iPodu Touch už si nemuseli za tuto verzi platit, jak tomu u nových verzi systému bylo doposud. Viditelnou novinkou bezesporu byl nový vzhled docku s hlavními aplikacemi. Systém také přinesl dlouho očekávaný Multitasking, rychlé přepínání aplikací, možnost aplikace a hry soustředit do složek a dovolit tak mít na ploše více aplikací (viz. Obr. 3-11). Notification Centre, místo kde se schází všechny aktualizované informace o aplikacích, jako jsou nové e-maily, zprávy atd. Bylo nyní přístupné přetažením prstu z horní části displeje dolů (viz. Obr. 3-12). Nově přinášel iBook i pro iPhone a iPod Touch, dále pak podporu iAd, reklamy v aplikacích. Opustil podporu původního iPhone. Ve svých aktualizacích pak přinesl Game Center portál pro hry, HDR fotky, možnost vytvoření Hotspotu a sdílení internetu přes 3G, podporu AirPlay protokolu pro přehrávání hudby přes Wi-Fi, přidává v rychlém přepínání aplikací možnost zamknutí překlápění do horizontálního zobrazení. Nakonec pak opouští podporu iPhone 3G a druhé generace iPodu Touch. [4;5;6]

Obr. 3-11: iOS folders



Obr. 3-12: iOS Notification Centre



Zdroje: ipodrepublic.com ; i.i.com.com

iOS 5 přišlo v září roku 2011, přineslo podporu služby iCloud a iTunes Match. Velkou novinkou byly iMessages (služba Applu, která obchází operatory a dovoluje posílat sms mezi stejnými zařízeními přes datové služby – jako IM). Dále přináší notificační centrum, systémovou integraci Twitteru, aplikaci připomínek Reminders a možnost dostat se do aplikace Camera ze zamknutého iPhone jednoduše tažením ikony nahoru po displeji. V aktualizacích pak přináší opravy a optimalizaci systému pro nové verze iPadu a stávající verze přístrojů. [4;5;6]

Nejaktuálnější verze systému je iOS 6, která přišla s novým iPhone 5. Přinesla mapy Applu s možností navigace, které vystřídali mapy od Google, aplikace je povedená, avšak podklady map jsou tak žalostné, že i sám Apple na popud uživatelů uznal, že udělal chybu a bude se snažit vše opravit a dovést aplikaci k dokonalosti. Dále přinesla funkci Do-Not-Disturb, která dovoluje nastavit, aby v nějakém časovém interval nebylo možné zvukově rušit uživatele, například v noci. Nové funkce omezení přístupu k některým funkcím, zrušení aplikace YouTube, systémová integrace Facebooku a FaceTime přes mobilní operátory. [5;6]

Budoucí systém by měl podle všech přinést velké změny, pokud chce Apple držet krok s konkurencí, musí přijít velkou inovací v uživatelském prostředí. Jahathan Ive byl pověřen, aby pracoval na novém vzhledu systému, všichni tak doufají, že nová verze bude vypadat lépe a že bude mít změny ve funkčnosti, jak však všechno dopadne, na to si musíme počkat. Podle posledních informací se nový systém opozdí a to i přesto, že vývojáři OS X byli z části převedeni na vývoj iOS 7.

3.5 Jailbreak

iOS Jailbreak je proces, kdy se ruší limitace systému na mobilních Apple zařízeních. Dovoluje probourat bariéru „rootu“ systému a poté není problém instalovat neoficiální aplikace (mimo App Store) a upravovat nastavení systému, které jsou normálně zakázána. Je to typ tzv. zvýšení úrovně oprávnění pro uživatele. Jailbreak není Applem povolený. (Ne)legálnost a s ní spojené možné právní problémy závisí na ústavách a zákonech jednotlivých států. Jailbreak existuje od první verze iPhone OS, v průběhu let se měnily tvůrci jednotlivých verzí Jailbreaku, některé verze systému byly jednodušší na prolomení, některé trvaly delší dobu. Obecně se dá ale říct, že každá verze byla nakonec prolomena. Samotné nástroje jsou například: PwnageTool, redsn0w, Absinthe, evasi0n. Mezi jejich tvůrce patří: iPhone Dev Team, evad3rs, comex, Chronic Dev Team a další. Apple varuje, že použitím jailbreaku na zařízení Applu zaniká záruka na zařízení. [3;4]

4 Konkurence na mobilním trhu

4.1 Google a Android

Společnost Android Inc. byla založena v Kalifornii roku 2003. Jeho platforma pro mobilní platformy, založené na Linuxovém jádru měla řešit omezení, kterými disponovala klasická přenosná zařízení, jako výdrž baterie, menší výkonnost a nedostatek dostupné paměti. Zároveň bylo jádro systému Androidu navrženo pro běh na různém hardwaru. Systém tak může být použit bez ohledu na použitý chipset, velikost, nebo rozlišení obrazovky. V roce 2005 Android Inc. koupila společnost Google a udělal z ní svou dceřinou firmu. V roce 2007 bylo vytvořeno konsorcium s názvem Open Handset Alliance. Zahrnovalo společnosti zabývající se výrobou mobilních telefonů, čipů nebo mobilních aplikací, např. Google, HTC, Intel, Texas Instruments, Qualcomm, Motorola, NVIDIA, Samsung, a dalších 26 společností. Jejich cílem bylo vyvinout otevřený standard pro mobilní zařízení. V ten samý den Open Handset Alliance ohlásil svůj první produkt, Android, otevřenou mobilní platformu postavenou na jádře Linux verze 2.6. Google pak oznámil, že jejich vizí je platforma, která bude moci být použita na tisících rozdílných modelech mobilních zařízení. O týden později byl vydán první Android SDK, nástroje pro vývojáře pod licencí open-source. [8]

První prototype Androidu bežel na mobile podobném BlackBerry s hardwarovou klávesnicí a nedotýkovým displejem. První mobil s dotykovým displejem a systémem Android přišel na trh roku 2008. Steve Jobs Android považoval za kopii iOS a za velikou zlodějinu. Příchod tohoto systému považoval za pověstnou kudlu do zad od Google. Do té doby totiž firmy úzce spolupracovaly a řada služeb Googlu se těšila v iPhone velké pozornosti. Bývalý CEO a současný výkonný předseda Googlu Eric Schmidt, byl totiž také členem představenstva společnosti Apple. Zvolen byl v roce 2006 a až v roce 2009 rezignoval kvůli možnému konfliktu zájmů. V chování Schmidta viděl Jobs silnou zradu. V životopise Steva Jobse Walter Isaacson píše, že nejhorší období Steva bylo v roce 2010, kdy HTC uvedlo na trh mobilní telephony s Androidem, které v mnoha věcech kopírovali funkce iPhone. V jednom z rozhovorů do životopisu dokonce řekl: *"Pokud to bude potřeba, budu se bit do posledního dechu a utratím každý cent ze 40 miliard dolarů, které má Apple v bance, abych napravil toto bezpráví,"* a velice rozzloben pak dále pokračoval: *"Zničím Android, protože je to ukradený produkt. Klidně kvůli tomu rozpoutám termonukleární válku."* Následná schůzka se Schmidtem nic nezachránila, Jobs dal jasně najevo, že nechce jejich peníze, že těch má dost, jediné co chce, aby přestali používat nápady Applu v jejich Android systému. Android je v dnešní době největší rival iOS, má velkou část trhu, díky tomu, že skoro všichni výrobci ho dávají do svých mobilů. Nejnovější verze 4.2 (Jelly Bean) (viz. Obr. 4-1) je už hodně vyspělá, v hodně věcech podle některých ohlasů předčí iOS. Google take přišel

s oficiálních obchodem Google Play ve stylu Apple App Store, který napomáhá k ještě silnějšímu upevnění místa na pomyslném vrcholu v tomto odvětví. [9;10]

4.2 Mobilní Windows

Společnost Microsoft je velkým rivalem Applu, jak na poli stolních operačních systémů, tak i na poli systémů mobilních. Jeho zatím posledním mobilním operačním systémem nese označení Windows Phone. Jeho verze WP 7 byla ještě stále postavena na jádru operačního systému Microsoftu pro malé počítače Windows CE. Přinesl poprvé rozhraní Metro, je to grafické uživatelské rozhraní, k jeho zásadám patří jednoduchost, zaměření se na obsah a typografii. Základní domovská obrazovka pod názvem "Úvodní obrazovka" se skládá z tzv. dlaždic. Dlaždice jsou odkazy na aplikace, vlastnosti, funkce a jednotlivé položky (například kontakty, webové stránky, aplikace nebo mediální položky). Uživatelé je mají možnost přidávat, měnit pořadí, nebo je odstranovat. Dlaždice jsou dynamické a aktualizují se v reálném čase. Označení Metro je však ochranná známka německé firmy Metro AG, a proto již není tento název firmou Microsoft používán. Označení používané pro vývojáře je Modern UI. V současné době je odkazováno jednoduše na rozhraní Windows 8. Nejnovější verzí operačního nese název Windows Phone 8. Systémy a aplikace již nejsou zpětně kompatibilní, Windows Phone 8 totiž přešel na novou platformu, jádro systému rodiny Windows NT. Jendou z novinek, co osmá verze systému přináší je podpora nativního kódu (C a C++), tím se stává jednodušší portovat aplikace z platform jako jsou Android, Symbian a iOS. A další nespornou výhodou, kterou osmá verze má je nativní podpora aplikací pro nejnovější desktopový systém Windows 8. Microsoft se spojil s Nokii a mají v úmyslu společnou silou útočit na výsluní mobilního odvětví. Nokie tak přešla na Windows a vlastní dosavadní systém Symbian OS do budoucna pohřbívá. Jednoduchost poslední verze systému tomu může pomoci. Na následujícím obrázku jsou vidět 3 hlavní zástupci mobilních operačních systémů (viz. Obr. 4-1). [11;12;13]

Obr. 4-1: iOS, Android & Windows Phone



Zdroj: <http://www.techulator.com/.../WP-8-Android-Jellybean-iOS-6>

4.3 BlackBerry

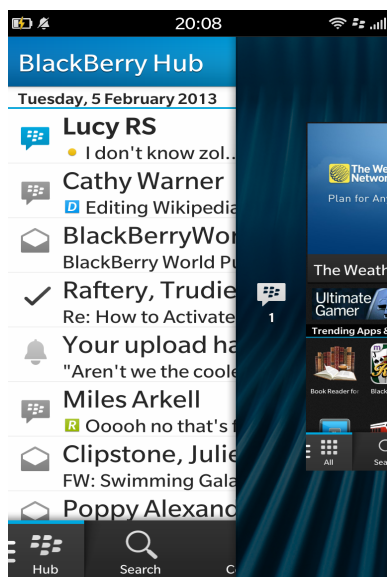
Research In Motion Limited (RIM), kanadská telekomunikační společnost byla založena roku 1984. V celé Severní Americe patřila až do nedávna k hlavnímu leaderovi ve sféře mobilních platform a řešení. Jejich mobilní telefony s hardwarovou QWERTY klávesnicí s názvem BlackBerry, díky propojení a vlastní kryptované síti se sofistikovaným psaní zpráv mezi uživateli BlackBerry a silným korporátním řešením, zaručily RIM prvenství v tomto odvětví na hodně let. Po příchodu Applu a Googlu na trh s chytrými telefony, se však všechno změnilo. Již první rok, kdy se začal iPhone prodávat v USA, se prodeje BlackBerry začaly žalostně propadat. iPhone byl také mnohými nazýván jako „BlackBerry Killer“ tzv. zabiják tohoto mobilního zařízení, respektive celé společnosti. Google se svým Androidem tomuto jevu moc nepomohl. Společnost se začala pohybovat na tenkém ledu, jejich operační systém v tom provedení jaké měl prostě nestačil své konkurenci a zákazníci si to začali uvědomovat. Skalní fanoušci a uživatelé zůstali, ale velká část přešla ke konkurenci. RIM došlo k radikálnímu řešení, začali pracovat na novém operačním systému Unix-like jádra QNX, pro dotykové i standardní mobilní telefony s jejich vlastní filosofií. Vydala v roce 2010 BlackBerry Tablet OS a v roce 2012 při velké představení akci uvedla na trh svůj nový systém BlackBerry 10. S tímto systémem přišly i dvě nové verze mobilních telefonů, jedna plně dotyková verze Z10 a druhá s klasickou klávesnicí známou pro BlackBerry. S uvedením systému také RIM ohlásilo nečekanou věc atím bylo přejmenování společnosti po svém hlavním výrobku, tedy BlackBerry. Nový operační systém obsahuje tzv. BlackBerry Hub, něco ve smyslu notifikačního centra, přístupný odkud koliv ze systému jednoduchým dotykovým gestem, který obsahuje všechny aktualizované informace napříč systémem. Má však svojí filosofii, kde všechny aplikace jsou v přístupné v jedné a jednoduše se z Hubu píšou maily, ale i ovládají další věci. Systém má i vlastní obchod s aplikacemi, a podle prvních ohlasů se nový systém BlackBerry povedl. Což je dobrá správa pro záchranu společnosti a možnému návratu zpět na výsluní. Ukázka BlackBerry hubu (viz. Obr. 4-2). Systémové vývojové nástroje umožňují programovací jazyky C, C++, HTML5, JavaScript, CSS a Java. [14;15;16;17]

4.4 Ostatní systémy

Dalším systémem na poli chytrých mobilních telefonů je WebOS společnosti Hewlett-Packard. Je založen na Linux jádru a proprietárních komponentech vyvinutých firmou Palm. Je nástupcem Palm OS. Obsahuje sadu programů pro správu osobních informací včetně webových technologií, jako je HTML 5, JavaScript, AJAX, nebo CSS. Nechybí podpora web 2.0. Prostředí webOS je založeno na systému tzv. karet, použitých k přepínání mezi spuštěnými aplikacemi. [18]

Novinkou roku 2013 a velice zajímavým řešením na trhu je mobilní verze Linuxového systému Ubuntu. Ubuntu Touch (viz. Obr. 4-3) vychází z desktopového bratříčka Ubuntu. Vývojáři šli cestou jednoduchosti, elegance a přehlednosti, kterou spojili s funkcemi a stylem používání stolního rozhraní Unity. Využívá všechny technologie jako jeho bratr. Při zapojení na externí monitor, se tak Ubuntu Touch může změnit na desktopovou verzi s plnou kancelářskou podporou, dokonce i hry běžící na ARM procesoru je možné hrát. Využívá synchronizačního úložného mraku Ubuntu One, který pomáhá synchronizovat vše jednoduše přes internet. Programovat se pro něj může jak pomocí framework Qt, tak i pomocí HTML 5. Zajímavostí je, že díky První ohlasy vypadají více než přívětivě, avšak na trhu to nebude mít jednoduché, Android si jen tak do svého teritoria nebude konkurenci chtít pustit. [19]

Obr. 4-2: BlackBerry Hub



Zdroje: [4; 19]

Obr. 4-3: Ubuntu Touch



5 Možnosti vývoje aplikací pro iOS

5.1 Úvod

Základním kamenem při vývoji jakékoliv aplikace pro jakoukoliv platformu, je určitě bezesporu programovací jazyk, který pro vývoj budeme používat. Nativním programovacím jazykem pro iOS je Objective-C, někomu však nemusí vyhovovat, nebo nechtějí přecházet, když ovládají dokonale jiný. Obecně vzato, je pro každou platformu vždy možné vyvíjet aplikace a programy pomocí více různých programovacích jazyků, nástrojů a postupů. Jediným kritériem je nativní podpora platformy pro ten určitý jazyk, nebo možnost portace na tuto platformu s dodržováním určitých zásad. Pak už je jen na nás, který z podporovaných programovacích jazyků nám sedne nejlépe, nebo kterému sami nejvíce holdujeme. Pro iOS je možností jak vyvíjet aplikace několik, a proto se s nimi seznámíme.

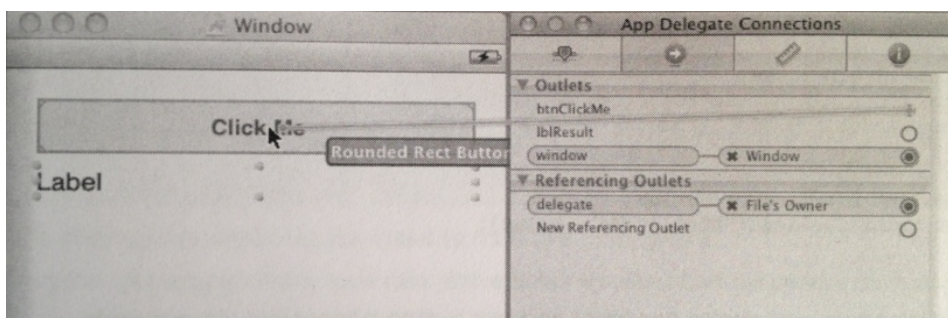
5.2 Vývojáři .NET a jazyk C#

Framework .NET společnosti Microsoft je ve světě hodně rozšířený a základna vývojářů, kteří programují v .NET filosofii je velice rozsáhlá. Nativním integrovaným vývojovým prostředím pro platformu Windows je Visual Studio. Vývojáři .NET však mohou vytvářet aplikace pro iOS také. MonoTouch Framework je součástí Mono projektu společnosti Novell. Projekt Mono je open-source implementace Microsoft .NET standardů platformy. Dovoluje tak běh .NET aplikací na skoro jakékoliv platformě, včetně iOS, FreeBSD, Linux, Unix, a dalších. Mono Touch byl představen v roce 2009 a rozšířil Mono tým, že dovilil psát aplikace pomocí jazyku C# a knihovni platformy .NET "Base Class Library (BLC)", která běží na iOS, použitím rozhraní pro programování aplikací UIKit Cocoa Touch. MonoTouch je extrémě efektní pro psaní iOS aplikací, protože kombinuje tyto vlastnosti:

- **Schopnost volat Objective-C, C, a C++:** MonoTouch využívá všechny síly nízkoúrovňových jazyků jakými jsou, Objective-C, a dalších. Dovoluje tak používat knihovny napsané v těchto jazycích přímo v MonoTouch kódu.
- **Konstrukce moderních jazyků:** Protože je MonoTouch postaven na platformě .NET, můžete tak dostat téměř všechny vymoženosti které obsahuje. Jsou jimi například automatická správa paměti, etc.
- **Moderní integrované vývojové prostředí (IDE):** MonoDevelop IDE má všechny vlastnosti, které předpokládáte v moderním vývojovém prostředí, včetně automatického doplňování kódu, integrovaného debuggeru, etc. [20]

MonoTouch uživatelské rozhraní bývají tvořena v Apple Interface Builderu (IB), v aplikaci, která je součástí iOS SDK. Ten používá Cocoa Touch (jak již víme, je to UI Framework Applu pro iOS, také znám jako UIKit) objekty, což znamená, že máme k dispozici všechny nativně vypadající ovladačí prvky v iOS pro programování. Nelze však pomocí MonoTouch za použití .NET technologie jako jsou Silverlight, WPF, nebo Winforms. Je na druhou stranu ale možné tvořit hry pomocí Microsoft XNA Toolkit, který se zaměřuje na WinPhone 7 a pak je portovat pomocí open-source XNA Touch projektu na iOS. Aplikace se pro iOS distribují, buď jako nativní aplikace přes Apple App Store, nebo pomocí ad-hoc deployment. Po stáhnutí iOS SDK od Applu a MonoTouch může tedy každý .NET vývojář začít vytvářet aplikaci pro iOS vcelku bez problému. Odkazy objektů do části kódu se dělají jednoduše přetažením myši z AppDelegate na požadovaný ovladačí prvek (viz. Obr. 5-1). [20]

Obr. 5-1: Nově vytvořené odkazy na objekty v MonoTouch



Zdroj: [20]

Ukázka pro propojení odkazu na kliknutí na tlačítko v MonoTouch vypadá následovně:

```
[MonoTouch.Foundation.Connect("lblResult")]
private MonoTouch.UIKit.UILabel lblResult {
    get {
        this._mt_lblResult=
            ((MonoTouch.UIKit.UILabel)(this.GetNativeField("lblResult")));
        return this._mt_lblResult;
    }
    set {
        this._mt_lblResult = value;
        this.SetNativeField("lblField", value);
    }
}
[MonoTouch.Foundation.Connect("btnClickMe")]
private MonoTouch.UIKit.UIButton btnClickMe {
    get {
        this._mt_btnClickMe =
            ((MonoTouch.UIKit.UIButton)(this.GetNativeField("btnClickMe")));
        return this._mt_btnClickMe;
    }
    set {
        this._mt_btnClickMe = value;
        this.SetNativeField("btnClickMe", value); }}
```


V dnešní době je MonoTouch Framework nahrazen modernějším řešením Xamarin.iOS. Má vylepšené prostředky k programování ještě více elegantních aplikací pro iOS. Programovací prostředí pro Xamarin.iOS se bylo přejmenováno na MonoDevelop a ve své verzi 3 obsahuje všechny nejnovější technické nástroje a podporuje nejnovější technologie. Základní verze MonoDevelop je zdarma, další nadstavby a licence jsou placené. Řešení pro .NET vývojáře to je však vskutku výborné. [20;21]

5.3 Kodéři HTML, CSS a JavaScriptu

Kódování webových programovacích jazyků je velice rozšířené po celém světě. Webové aplikace jsou na trhu již dlouho dobu a často jim stačí pouze webový prohlížeč. Možnost vytvářet aplikace pomocí kódovacích programovacích jazyků je možná samozřejmě i pro běh na iOS. Smyslem tohoto programování je obcházení iOS SDK. Hlavním záměřením v iOS je tak webový prohlížeč Safari. [22]

Webová aplikace (Web App):

Webová aplikace je v podstatě internetová stránka speciálně optimalizovaná pro iPhone. Může to být všechno, od standardní brožury nějakého malého biznisu, přes kalkulačku nájmu, až k vyhodnocovači denních kalorií. Obsah je irelevantní. Hlavní definující charakteristiky webové aplikace jsou, že uživatelské rozhraní je vytvořeno s webovými standardními technologiemi. Jsou k dispozici na své URL (privátní, veřejné, nebo po přihlášení), a je optimalizovaná pro specifikace iPhone. Webová aplikace není nainstalovaná na mobilním telefonu, není v nabídce Apple App Store, a není napsána v Objective-C.

Výhody:

- Web vývojáři mohou používat jejich známé nástroje.
- Můžete použít Váš vlastní webový design a vývojářské dovednosti.
- Nejste limitováni na vývoj pod Mac OS X.
- Vaše aplikace poběží na každém přístroji, co má webový prohlížeč.
- Opravovat chyby pro uživatele lze v reálném čase.
- Vývojový cyklus je rychlý.

Nevýhody:

- Nemáte přístup ke všemu výbornému vybavení telefonu.
- Musíte si sám vymyslet Váš pleticový systém, pokud chcete mít placenou aplikaci.
- Může být složité dosáhnout sofistikovaných efektů uživatelského rozhraní.

Nativní aplikace (Native App):

Ta je narozdíl od webové nainstalovaná na iPhone, má přístup k hardware (repráčky, accelerometer, kamera, etc.), a jsou psané v Objective-C. Definující charakteristikou je, že je k dispozici na App Store.

Výhody:

- Miliony majitelů kreditních karet jenom vzáleny jedno kliknutí.
- Xcode, Interface Builder a Cocoa Touch framework poskytují pro vývoj super prostředí.
- Je možné přistupovat ke všem hardwarovým vybavením iPhone.

Nevýhody:

- Je třeba zaplatit registrační poplatek Applu pro distribuci aplikace.
- Jste závislý na povolovací a testovací proces Applu.
- Je třeba umět programovat v Objective-C.
- Potřebujete k vývoji Mac.
- Nemůžete vydávat opravy chyb v rychlém sledu.
- Vývojový cyklus je pomalý, testovací cyklus je s přihlédnutím na App Store limitace. [22]

Většina stránek na webu jsou pouze textové dokumenty, které leží u někoho na PC. Text na typické webové stránce je zabalen v **HTML** tagách, které řeknou prohlížeči jak je dokument strukturovaný. Ukázka kódu takového HTML dokumentu:

```
<html>
  <head>
    <title>My Awesome Page</title>
  </head>
  <body>
    <h1>Hi There!</h1>
    <p>Thanks for visiting my page.</p>
    <ul>
      <li>Pizza</li>
      <li>Beer</li>
    </ul>
  </body>
</html>
```

Dalším základním kamenem webu je používání kaskádových stylových ploch **CSS**, je používáno pro definování vizuální prezentace HTML dokumentu. Můžete tím určit jednoduché vlastnosti, jako je barva písma, velikost, etc. a nebo i komplexní vlastnosti, jako rozložení na stránce, a podobné.

Ukázka úpravy HTML kódu s použitím CSS:

```
#highlight { background-color: yellow;}
<h1 class="loud">Hi There!</h1>
<p id="highlight">Thanks for visiting my page.</p>
  <ul>
    <li class="loud">Pizza</li>
    <li>Beer</li>
  </ul>
```

Doplní zvýraznění a styly písma. Více tak můžeme definovat celý dokument. [22]

JavaScript vám umožní, aby HTML dokument ožil a začal něco konečně dělat. Může být přidán do kódu HTML stránky a udělat tak obsah více interaktivní pro uživatele. Můžete díky tomu třeba prohledávat zadávací pole pro uživatele, a zjišťovat, jestli vložené údaje jsou správné. Můžete schovávat a zobrazovat elementy, podle toho kam uživatel klikne myší, etc. Ukázka takového oživení HTML dokumentu:

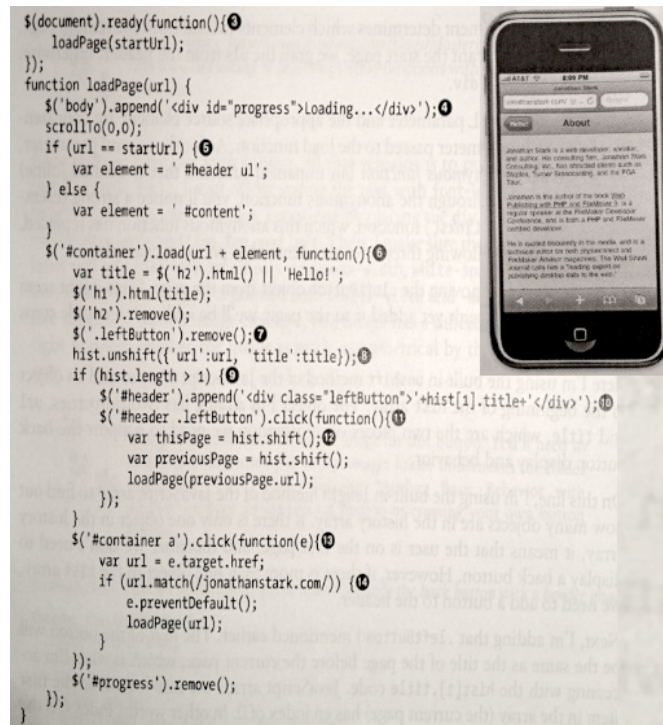
```
<html>
  <head>
    <title>My Awesome Page</title>
    <script type="text/javascript" charset="utf-8">
      function sayHello() {
        document.getElementById('foo').innerHTML = 'Hi there!';
      }
    </script>
  </head>
  <body>
    <h1 id="foo" onclick = sayHello()">Click me!</h1>
    <p>Thanks for visiting my page.</p>
  </body>
</html>
```

Přidali jsme blok scriptu do hlavy dokumentu, uvnitř bloku scriptu jsme definovali jednu JavaScript funkci s názvem „sayHello()“. Tato funkce říká prohlížeči, že má hledat v dokumentu element který má id ‘foo’ a nastavit jeho vnitřní HTML obsah ‘Hi there’ efekt tohoto je, že se text “Click me!” zamění za “Hi there!” poté co uživatel klikne h1 element. K tomu aby kód fungoval v různých prohlížečích stejně se pak využívá tzv. jQuerym relativně malé JavaScript knihovny, která dovoluje psát váš JavaScript kód aby se tak choval všude stejně. [22]

Webové aplikace lze naprogramovat tak, aby pracovaly offline a dokonce uměly ukládat data do paměti iPhone. Design se dá velice přiblížit nativním aplikacím a za použití PhoneGap open-source vývojářského nástroje, který funguje jako most mezi webovými aplikacemi a mobilními telefony, iPhone, Android, BlackBerry, etc. V dnešní době je na trhu již moderní verze HTML 5, a CSS 3, HTML 5 dovoluje neuvěřitelné věci, jednoduše umí ze stránek dělat extra interaktivní plochy s kódováním videa, atd. Pro Apple již definitivně nahradil Flash společnosti Adobe. Videa na portálu YouTube již jsou kódována v HTML 5, velké webové stránky přecházejí na novou verzi a ta

jim dovoluje dělat zajímavější a funkční internetové stránky. Pomalu se tak z normálních dokumentů HTML na webových stránkách stávají webové aplikace s plnou interaktivitou. Ukázka aplikace s tlačítkem zpět a její kód je vidět níže (viz. Obr. 5-2). [22;23]

Obr. 5-2: Web app code for back "Hello" button



Zdroj: [22]

5.4 Objective-C & Cocoa Touch

Nativním a hlavním řešením a nástroji pro vývoj aplikací pro iOS, jmenovitě pro iPhone, iPod Touch a iPad je právě objektově-orientovaný jazyk Objective-C a Framework Applu Cocoa Touch, vycházející z Cocoa Framework, známého pro desktopové systémy Applu. Toto řešení je pro případné následné prodávání aplikace přes App Store to nejideálnější pro bezproblémové vyřízení. Aby ne, je to řešení Applu pro jeho vlastní platformu a ekosystém. Pokud tedy s programováním začínáte a není tedy problém s přečtením se z jiného programovacího jazyku, nebo jste vývojář pro Mac OS X, je to nejjednodušší a nejs sofistikovnější volba pro Vás. Můžete totiž využívat bez problému všechny vymoženosti a originální prvky systému iOS, bez sebemenších známek nekompatibility. Případný přenos Cocoa aplikace pro desktopový Mac OS X na Cocoa Touch aplikaci pro iOS je poměrně jednoduchá. Stále se používá smysl MVC, Model, View, Controler (detailně popsané v mé bakalářské práci), ale business model, vývojové prostředí a design je rozdílný. Pro vytvoření dobré aplikace je třeba změnit vaše cíle a naučit se pracovat s menší obrazovkou a dotykovým rozhraním. Aplikace pro iPhone psané v Objective-C používají

podobný Cocoa Framework model. A jsou tu veliké rozdíly mezi desktopovými a mobilními aplikacemi. [23]

- **iOS aplikace** jsou menší a jednodušší aplikace, než aplikace pro OS X. Jsou daleko víc self-contained (ukládají informace a data do svých uložist) a mají limitovaný přístup k externím datům.
- **Uživatelské rozhraní** je řešeno přes jednodušší, chudší stavbu zobrazení z UIKit frameworku. Uživatelské rozhraní jsou menší a méně komplexní.
- **Hodně frameworků, vychytávek a ovladačů dat** v OS X nejsou k dispozici v iOS. Například Core Data jsou k dispozici, ale prostředí iOS není ideální pro velké jednotky dat, nebo komplexní databáze. Obyčejná grafika je tvořena pomocí Quartz 2D frameworku. Cocoa grafické třídy nejsou k dispozici.
- **Podpora pro extra hardwarové vlastnosti** – včetně multi-touch rozhraní a čidla natočení, otřesů a rotace - ty byly přidány a pro zlepšení a intuitivě zážitku z používání.
- **Obsahuje věci, které nejsou pro desktop v dnešní době ještě standardem.** Lokace polohy pomocí GPS, multi-touch ovládání obrazovky, ovládání pomocí dotykových gest, snímání orientace polohy zařízení, snímání naklonění a pohybu zařízení.

Pokud' tedy přecházíte z OS X do světa vývoje iOS je třeba pochopit, že smysl a cíl ovládání aplikací a nakládání se soubory a informace se v obou platformách hodně liší. Díky nativní podpoře jazyku Objective-C a jejího vývojového prostředí je pro tvorbu aplikací pro tento systém bezkonkurenčně nejlepší. Pokud' Vám tedy jazyk vyhovuje a nejste zapřísněný uživatel jiného řešení, ale jak jsme mohli vidět, i pro ty se najde cesta, jak vytvářet bez problému aplikace pro iOS. [23]

Objektově-orientovaném jazyku Objective-C, který je nadstavbou jazyku plain C a byl vyvinut roku 1986 spojením jazyku Smalltalk a jazyku C, jako nativní programovací jazyk operačního systému NeXTStep, jsme detailně popisoval ve své bakalářské práci na téma: "Možnosti tvorby aplikací pro Mac OS." Tudíž je zbytečné se opakovat i v této práci a raději odkáži na mojí předchozí práci.

5.5 Xcode 4.6

Je to nativní integrované vývojové prostředí (IDE) od Applu pro tvorbu iOS aplikací. Xcode aplikace zahrnuje většinu vývojářské dokumentace Apple, soubor nástrojů a knihoven pro iOS s názvem iOS SDK a hlavně testovací virtuální iPhone prostředí. V neposlední řadě pak Interface Builder, aplikaci pro konstrukci grafického uživatelského rozhraní. Podporuje mnoho rozdílných jazykových kódů. Umožňuje vývojářům bezproblémovou tvorbu jejich aplikací. Je zdarma ke stažení na Mac App Store. Při spuštění nabízí mnoho předpřipravených template, které ale můžeme

samozřejmě dále úplně měnit. Více se mu budeme věnovat při popisu postupu. Hodně budeme využívat i integrovanou a propracovanou nápovědu, která je velice intuitivní. Jednotlivé Templaty které jsou v nabídce prvního spuštění Xcode si představíme:

- **Single View Application.** Poskytuje startovací čáru pro aplikaci, která používá ke svému fungování jediné zobrazení. K tomu jí slouží View Controller. Ke správě tohoto zobrazení pak slouží například „nib“ soubor (speciální souborový zdroj, v němž se ukládá uživatelské rozhraní aplikací iOS, je to dokument Interface Builderu, hlavního tvůrce rozhraní. Díky jednoduchosti, se tato šablona hodně používá pro hry na iOS.
- **Tabbed Application.** Je to základ pro aplikace se záložkami. Poskytuje nám rozhraní pro uživatele, které obsahuje kontrolér pruhu záložek (Tab Bar Controller) a je spojen s podkontrolérem zobrazení (View Controller), pro jednotlivé záložky ve spodním pruhu Tab Bar Controlleru.
- **Utility Application.** Použijte se pro pomocné aplikace, používá zobrazení (Main View) a pak zobrazení alternativní. Na iPhone tak nastaví tlačítko „info“, kterým se do alternativního zobrazení které se ukáže v tzv. "pop-over" (je to samostatná vizuální vrstva, která pluje nad hlavním zobrazením). Nezakryje celou obrazovku a zmizí, když uživatel klepne vně tohoto zobrazení.
- **Empty Application.** Jak už název sám napovídá, jedná se o univerzální prostředí pro jakoukoliv aplikaci. Obsahuje jedno Application Delegate a prázdné okno. Hodí se, když chcete svoji aplikaci vytvářet úplně od začátku bez jakéhokoliv nadbytečného nepotřebného kódu, který s sebou mohou templaty přinést. [23]

Základní rozdělení okna a popis hlavních funkcí Xcode je vidět na obrázku (viz. tištěná Příloha č. 1).

6 Campus Navigator pro iPhone

Mým vlastním projektem měl být jednoduchý Campus navigátor, který by podával základní informace o kampusu ČZU, o umístění učeben, a další informace, které se hodí nováčkům na ČZU. Aplikace funguje offline, pouze pokud použijeme mapu s GPS, tak si potřebuje stáhnout podklady pro mapy. Pokusím se popsat co nejsrozumitelněji a polopaticky, jak jsem takovou aplikaci vytvořil, aby bylo popříkladě možné, použít některé postupy pro tvorbu aplikace čtenáře.

6.1 Návrh aplikace

Jako první krok při vývoji jakékoli aplikace, je její návrh. Nejlépe je nakreslit si aplikaci jednoduše tužkou na papír, postupně jí upravovat podle toho, jak bychom si ji představovali vzhledově, přidávat průběžně nějaké prvky, naopak nějaké ubírat, přetvářet, až se dostaneme do stadia, kdy si myslíme, že takhle by mohla zhruba vypadat. Samozřejmě je to jen návrh, skeč, který nám pouze pomáhá urovnat si myšlenky a do budoucna nám poskytne při návrhu skutečné aplikace ve vývojovém prostředí základní kámen, od kterého se můžeme odrážet.

Dále je třeba si sehnat, vytvořit designové součásti. Například spouštěcí aplikační ikonu, pod kterou se bude na obrazovce iPhone naše aplikace. Já osobně jsem si vypůjčil obrázek z webu ČZU, kde jsou pohromadě znaky více fakult naší university. Ve photoshopu jsem upravil tento obrázek a díky plug-inu pro tvorbu iOS ikon, jsem vytvořil jednoduchou ikonu, kterou uvidíte dále v postupu. Je třeba mít verzi pro normální iPhone a pak Retina displej verzi. Druhou potřebnou součástí je tzv. launch picture, obrázek, tapeta, která se zobrazuje při spuštění aplikace. Pro tu jsem použil opět stejný základ jako pro ikonu, bílý podklad tapety a do spodní části své jméno a letošní rok. Samozřejmě se fantazii žádné meze nekladou, je pouze na Vás jak svou ikonu a spouštěcí obrázek navrhnete a vytvoříte. Poté je dobré dopředu si podle návrhu, kde asi tak víme, co budeme potřebovat, sehnat obrazové podklady, které třeba použijeme v naší aplikaci. U mne osobně se jednalo například o fotografie university, menzy, kolejí a mapy kampusu školy. V neposlední řadě je třeba sehnat si podklady pro pozadí jednotlivých obrazovek, jestli to máme v úmyslu a pokud chceme originální, vlastní vzhled tlačítek, nebo ovládacích prvků, je třeba je někde zakoupit, nebo vytvořit například ve photoshopu. Pokud máme vše co budeme potřebovat, můžeme přejít na další část. Samozřejmě další potřebné věci bude možné dohledávat později přímo při tvorbě, je však lepší mít vše, nebo většinu věcí předem.

Pro tvorbu aplikace je potřeba vlastnit vývojové prostředí, pro programovací jazyk, který jsme si vybrali pro vývoj aplikace. V mém případě to je Objective-C a jako vývojové prostředí jsem tudíž zvolil Xcode verze 4.6.1, tedy tu nejaktuálnější verzi. Stáhnout se dá zdarma z Apple App Store.

Spolu s Xcode se automaticky nainstaluje i nejnovější verze iOS SDK, neboli nástrojů pro tvorbu aplikací. Po nainstalování jsme připraveni na vlastní tvorbu aplikace.

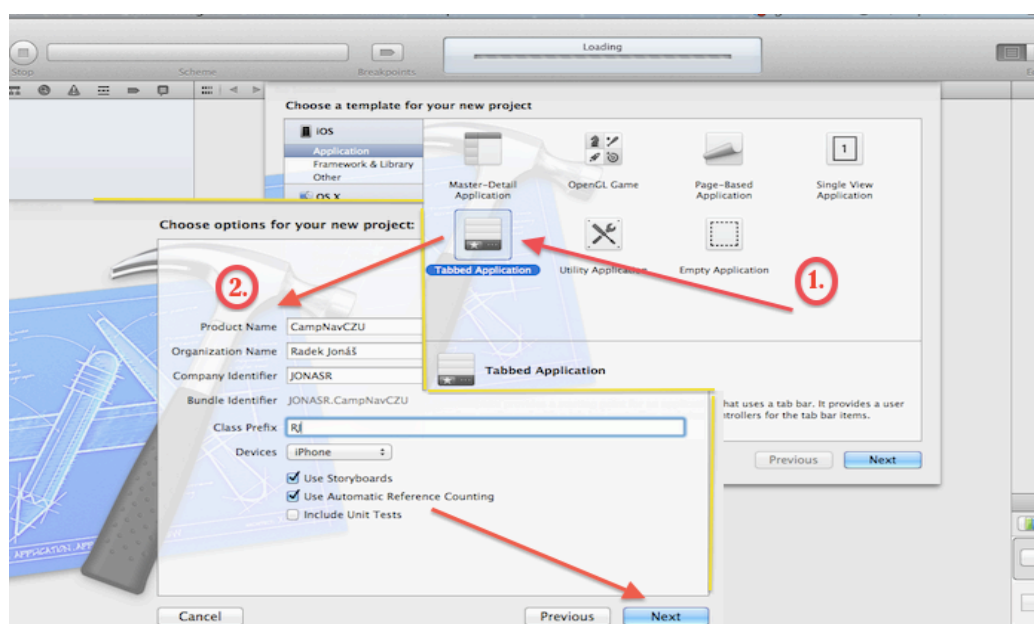
6.2 Xcode projekt

Pro lepší orientaci a srozumitelnost, jsem upravil screenshoty, kde používám různé barvy a číslice, jejich význam je tento:

- **Červené číslice a šipky** značí pořadí směr postupu, který bude popsán v textu.
- **Žluté ohraničení** značí vložený screen z další obrazovky, například potřebný pro další výklad postupu postup, kdy by bylo zbytečné použít ho do samostatného obrázku.
- **Modré ohraničení a šipky** značí zvýraznění některých míst, o kterých se budu v textu zmiňovat a jsou důležité pro postup.

Když spustíme poprvé prostředí Xcode, vyskočí na nás nabídka základních šablon (viz. Obr. 7-1).

Obr. 6-1: Xcode - vytvoření projektu

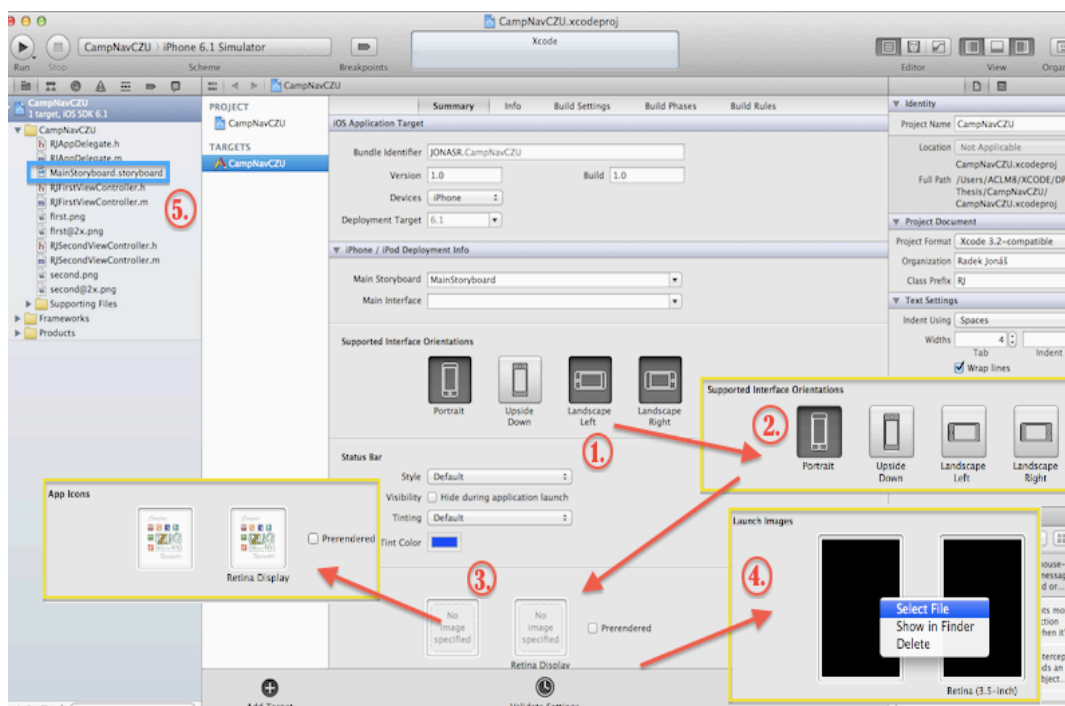


jak ale víme, jsou to jen pomocné šablony, které můžeme potom naprosto změnit přímo při tvorbě. Pro mou aplikaci se bude hodit pro začátek vybraný šablon, na který míří červená šipka (viz. bod 1), s názvem Tabbed Application v sekci iOS Application. Vysvětlení, co nám Xcode předpřipraví tímto nastavením je napsáno vlevo od tlačítka Next. Tímto tlačítkem postoupíme na další okno (viz. bod 2). Zde zadáme název naší aplikace, já svou nazval „CampNavCZU“, následuje název naší společnosti, nebo jako v mém případě jméno tvůrce. Company Identifier je identifikátor, který Xcode bude psát před název aplikace a slouží k identifikaci aplikací Vaší společnosti. Class Prefix je zkratka, kterou bude Xcode používat pro identifikaci kódu, respektive

Vašich objektů a tříd, aby oddělil objekty a funkce a třídy Applu a od Vašich. To dopomáhá k jednoduššímu hledání chyb a čitelnějšímu kódu. Zařízení je našem případě pouze iPhone, ale je možno vybrat také pouze iPad, nebo univerzální aplikaci. Necháme zaškrtnuté políčka Use Storyboard, který budeme používat a automatické počítání referencí není potřeba, ale ničím nám aplikaci moc neovlivní jeho zapnutí, takže ho necháme. Testy všech verzí aplikace nebudeme potřebovat. Pokračujem tedy po směru šipky a tlačítkem Next pak dokončíme vytvoření projektu naší budoucí aplikace.

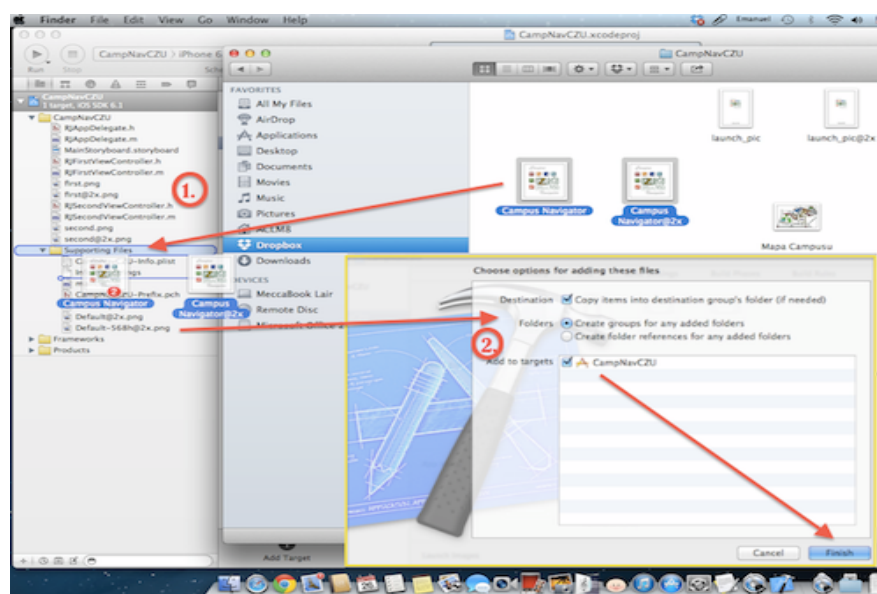
Objevíme se tak v hlavním okně vývojového prostředí Xcode (viz. Obr. 7-2). Ve sloupci v levé horní části obrazovky, vidíme všechny hlavní části a dokumenty našeho projektu. Modře je označen Story Board, ve kterém budeme navrhovat grafický vzhled a funkční část aplikace. Veprostřed nahoře vidíme informace o našem projektu a tudíž i orientace aplikace (viz. bod 1). Protože aplikace CamNavCZU bude fungovat pouze v Portrait poloze, můžeme zbytek variací deaktivovat (viz. bod 2). Dalším místem jsou pak ikony aplikace, rozdělené podle verzí iPhone, je tedy třeba mít obě verze. Jak podklady dostaneme do projektu si ukážeme na dalším screenshotu. Stejně tak pak můžeme nastavit o kousek níže tapety, které budou vidět při spuštění aplikace. Poslední zmínka ukazuje na část programu StoryBoard (viz bod 5), kde strávíme druhou velkou část projektu. V tomto hlavním okně se dají nastavit i další věci, které více přizpůsobí projekt potřebám naší aplikace, respektive si projekt upravíme aby co nejvíce vyhovoval naší aplikaci.

Obr. 6-2: Uvodní nastavení projektu



Data a obrázky se do projektu dostanou nejjednodušeji, stejně jako do jakékoliv aplikace v Mac OS X. Jednoduchým přetažením do okna aplikace (viz. Obr. 7-3). Najdeme si složku, kde máme uložené potřebné .png soubory a přetáhneme je do okna Xcode, nejlépe do složky Supporting files, jak je možno vidět na screenshotu. Ve chvíli, kdy pustíme trackpad a soubory vložíme do projektu, zobrazí se okno s nabídkou (viz. bod 2). Je to nabídka, jak má Xcode s těmito soubory nakládat. Zaškrtněte určitě horní možnost, kdy Xcode zkopíruje Váš soubor do složky projektu, kdyby jste nezaškrtnli, a poté soubor někam na disku přesunuli, Xcode už ho nebude moct zobrazovat, protože původní odkaz už na něj nebude ukazovat. Zbytek nastavení můžete nechat a pokračovat tlačítkem Finish.

Obr. 6-3: Import souborů do projektu

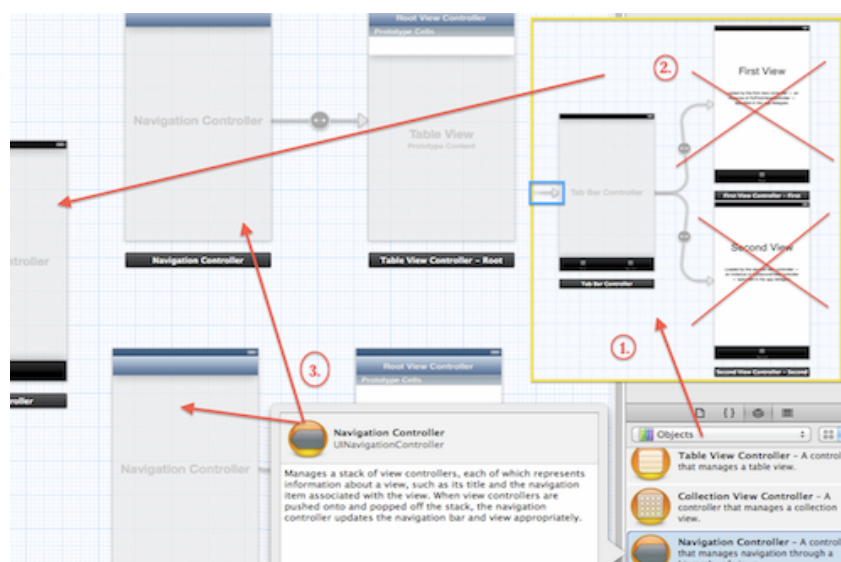


6.3 Uživatelské rozhraní

Dostáváme se do stadia návrhu a tvoření uživatelského rozhraní a funkcionality našeho projektu. K tomu nám slouží sekce “MainStoryboard”. Protože jsme si na začátku při vytváření projektu vybrali template pro aplikaci s Taby, tzv. přepínači pro různé obrazovky, po otevření StoryBoardu (viz. Obr. 7-4) vidíme okno Tab Bar Controleru a jeho dvou oken, kde každé zobrazuje jednotlivé okna Tabů. Samotný Tab Bar Controler si můžeme představit jako kontejner vlastností pro zbytek oken, které mu náleží, je to vlastně ovladač a super třída těchto oken, dědí po něm vlastnosti a přitom ještě každé okno má svoje unikátní vlastnosti. Do samotného controléru nelze vkládat žádné objekty. Je to opravdu jen kontejner a ovladač zbytku. Jeho funkce záložek se nám hodí, ale jeho pod-okna představují pouze zobrazovací okna, která už nejdu více rozšiřovat takovým způsobem, jaký mám v plánu. Začátek aplikace, to okno, které je aktivní vždy po zapnutí aplikace je značeno vstupní šipkou, na obázku je zvýrazněna modrým obdélníčkem. V našem případě je

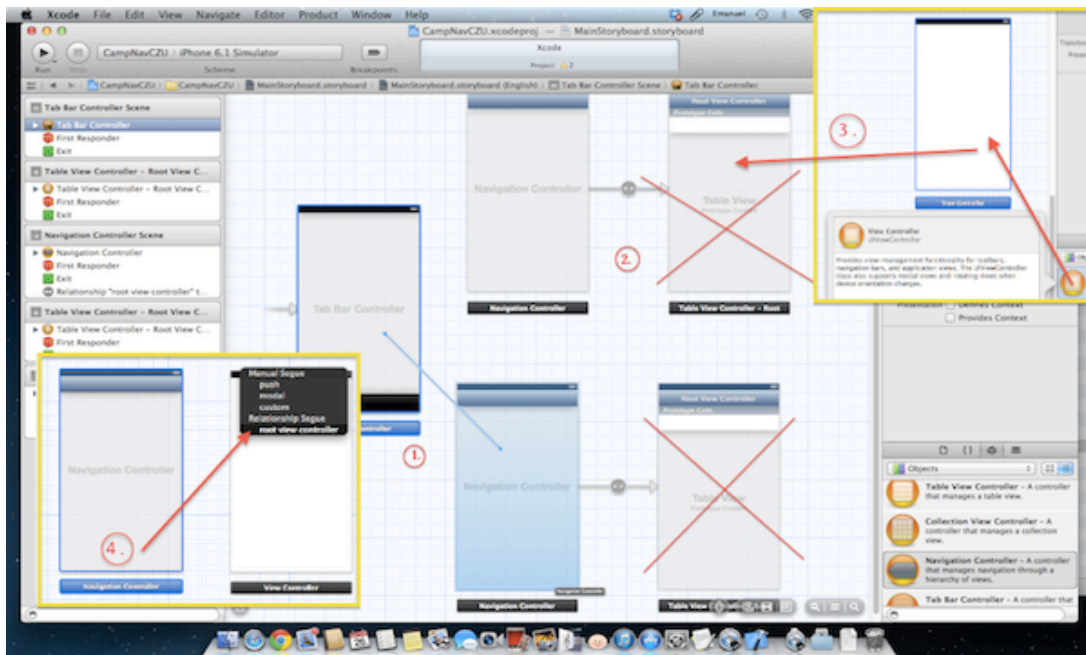
tomu tedy ovladač tabů, první se bude zobrazovat tab číslo jedna. Jelikož pro nás jeho pod-okna nejsou zajímavá, v první řadě je smažeme, jednoduše označíme a backspacem, nebo klávesou delete se jich zbavíme. (viz. bod 2). Další červená šipka nás přenáší do okna, kde už je náš controller sám a bez jakéhokoliv propojení, kromě vstupního. K němu vložíme řešení, které se nám bude pro tuto aplikaci hodit. Jsou jimi tzv. Navigation Controllery (viz. bod 3), které nám nabídnou ovládací prvek, tlačítko, které nás vždy vrátí o úroveň zpět, ať už půjdeme jakkoliv hluboko do struktury. Vložíme je dvakrát, pro každý Tab jeden controller. Navigation controller má však jako svoje podokna Table view controllery, nýbrž ovladače, které řídí okna s řešením zobrazení v Table, neboli rádkových zobrazeních. Toto řešení se nám také nebude do budoucna hodit, proto se jich zbavíme a zaměníme je.

Obr. 6-4: Navigation Controllery



Nejdříve propojíme hlavní součásti naší aplikace (viz. Obr. 7-5) přetáhnutím myši a přidržetím klávesy CTRL hlavní vstupní Controller s našimi Navigation Controllery (viz. bod 1). Vybere možnost spojení jako „root view cotroller“ je to stejná možnost jako spojení následného View Controlleru (viz. bod 4). Tato možnost nám do kódu nastaví toto logické propojení a usnadní nám tak část kódu, který bychom museli psát ručně. Jak jsem už předem oznámil, zbavíme se nechtěných Table View oken tak, že je jednoduše smažeme pomocí kláves delete, nebo backspace. Místo nich ze sekce vpravo dole, vybereme v objektech View Controller (viz. bod 3), který nám dovolí zobrazovat na obrazovce vlastně cokoliv, obrázky, scrollování, tlačítka a dá se s ním nejvíce pracovat a upravovat. Vložíme dva kusy místo našich Table View a propojíme je typem „root view controller“ s controllery, těmto obrazovkám nadsazené (viz bod 4).

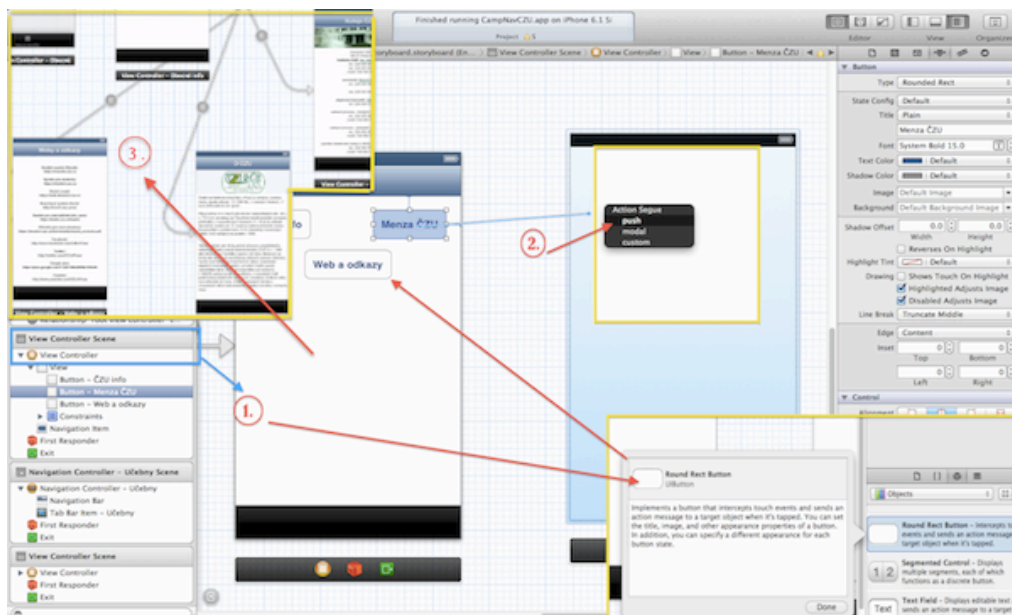
Obr. 6-5: Přidání View Controllers



Nové námi přidané View Controllery budou sloužit jako hlavní rozcestí v ovládní aplikace. Boudou to technicky právě tyto dvě obrazovky, které budou tím prvním co uvidíme po spuštění. Vytvoříme na nich tak tlačítka, které nás budou odkazovat na další obrazovky s detailnějšími informacemi na dané téma (viz. Obr.7-6). Na tomto screenshotu je modře označen typ tohoto okna, kterým je právě View Controller, v sekci „Objects“ vpravo dole vyhledáme objekt s názvem Round Rect Button, neboli jednoduše základní tlačítko (viz. bod 1). Do View části ho vložíme přetažením na vybrané místo na obrazovce, Xcode nám dokonce pomáhá zobrazením přerušované modré linky středy obrazovky, odsazení od okrajů, pokaždé, když přejedeme nad těmito body průniků tímto objektem. Zatím neřešíme moc design a vzhled obrazovky, k tomu se ještě dostaneme, spíše tedy vnitřní funkcionalitu – tlačítka tedy jednoduše někam umístíme. Každé tlačítko bude odkazovat na další View Controller podřízený tomuto hlavnímu, díky tomu že je součástí Navigation Controlleru, vždy bude v horní části ovládací prvek s názvem nadřazené (předchozí) obrazovky a lehce se tak vrátíme zpět. Vytvoříme tedy další View Controlery, které budou spojeny s tímto naším hlavním. Spojeny budou odkazem, kterého dosáhneme tak, že myší a přidržením klávesy CTRL přetáhneme světle modrou linku na obrazovku se kterou chceme udělat odkaz na tlačítko a vybereme možnost „push“. To znamená že se nám nová obrazovka po stisknutí tlačítka objeví celá, nahradí dosavadní a bude na ní ovládací prvek na krok zpět. Další možnost „modal“ (viz. bod 2) umožní to, že nová obrazovka překryje původní, ale žádné tlačítko zpět nebude jeho součástí, pak už je jen na nás, jak naprogramujeme následující obraovku a jak vyřešíme pohyb dál nebo zpět. Poslední možnost „custom“ nám nechá pouze odkaz v kódu a je už na nás jaký přechod spojení doprogramujeme. Každé následné okno pak můžeme libovolně

upravovat a plnit ho různými informacemi, které rozvinou myšlenku tlačítka, které nás na toto okno odkáže (viz. bod 3).

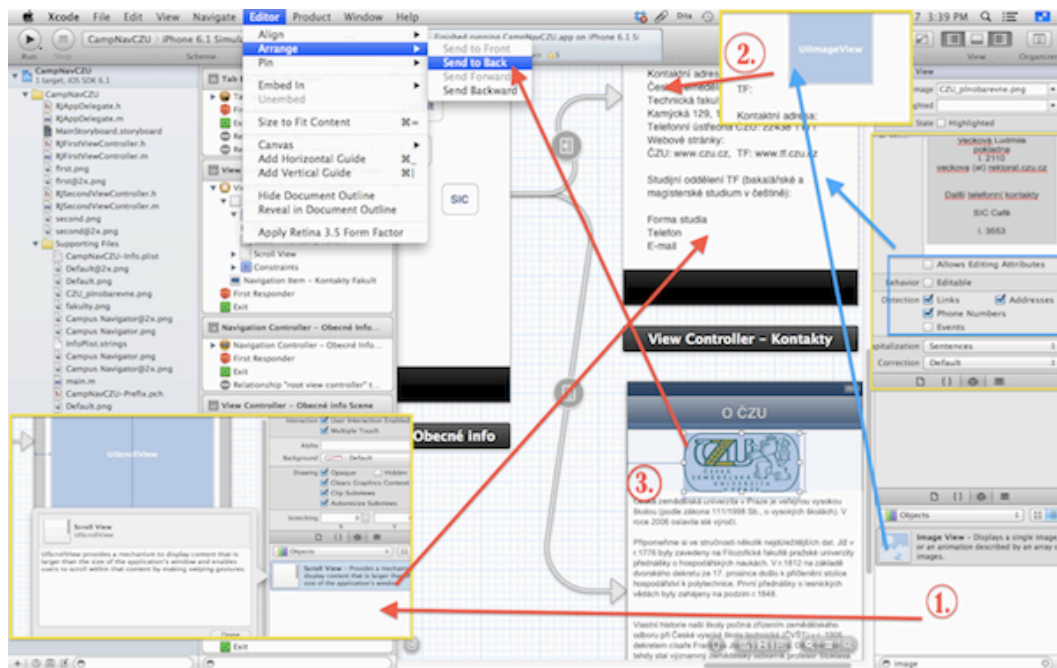
Obr. 6-6: View Controllery a tlačítka



Jak takového naplnění informacemi dosáhneme si ukáže právě teď. Informace, které jsem bral jako základní data mé aplikace pocházejí z informačních webů České zemědělské univerzity v Praze. Konečná okna s informacemi si tedy jenoduše upravíme, každé podle naší vlastní potřeby (viz. Obr. 7-7). Například do okna s informacemi o fakultách univerzity si vložíme objekt skrollovacího zobrazení „UIScrollView“ (viz. bod 1). Umožňuje nám vložit do něho text který je mnohem větší než displej iPhone a prstem scrollovat jeho obsah. Na jeho vrstvu pak vložíme objekt „UITextView“, do kterého můžeme textové informace vkládat. Pořadí závislosti přednosti UI objektů je vidět v každém controlleru nalevo v nabídce. Textový objekt by měl být v objektu scrollování, pokud je to naopak, přetáhněte text vně skrollovacího objektu. Text se vkládá vpravo nahoře v záložce data objektu, ukázka takového okna je ohraničená žlutě vpravo. Pokud chceme aby rozpoznával i mailové adresy a telefonní číslo, zaškrtneme tyto možnosti pod oknem pro vkládání textu. Můžeme samozřejmě i upravovat písmo a barvu textu. Zrušíme však možnost úpravy, aby byl text pro uživatele pouze ke čtení, to samé u možnosti úpravových atributů (viz. modré ohraničení napravo). Poté vložíme objekt „UIImageView“ nad scrollovací plochu s textem (viz. bod 2), kde pak vybereme jeho obsah, stejným způsobem jako je vidět v okně pod ním (viz. bod 3). Vpravo nahoře v záložce zdrojů dat UIImageView vybereme image z našeho projektu. Aby obrázek nekryl žádné informace, pošleme ho na pozadí, na spodní vrstvu, můžeme to provést dvouma způsoby. Buď přetáhnou vlevo v nabídce obrázek nad všechny ostatní objekty ve stromové adresáři, nebo ho označit jako tomu je na screenshotu a přes nabídku „Editor“

a „Arrange“ vybrat „Send to back“. To nám zaručí, že posuvný text nebo cokoli jiného nebude nikdy moci být zakryto nechtěně naším obrázkem, stejně tak se řeší pozadí celé obrazovky, jednoduše obrázek, tapetu pošleme na pozadí všeho. Naplníme tak i zbytek oken a základ máme hotový. Přejdeme tedy k úpravám vzhledu.

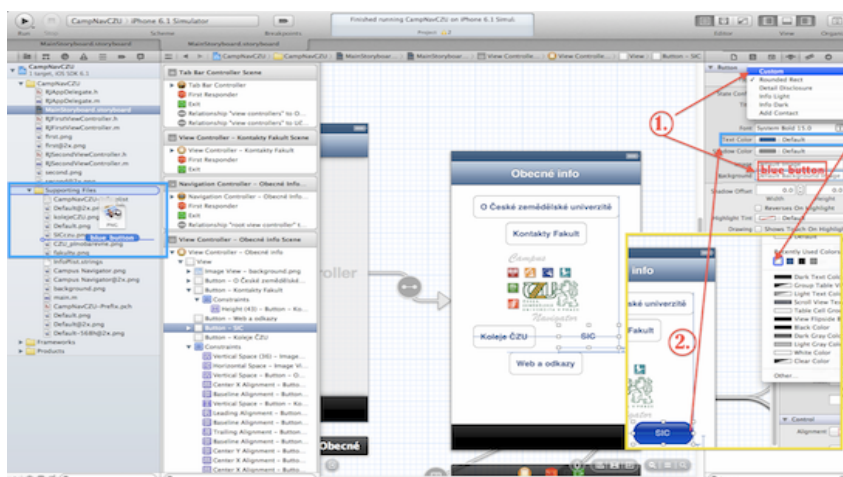
Obr. 6-7: Obrázky a naplnění informacemi



6.4 Úpravy vzhledu

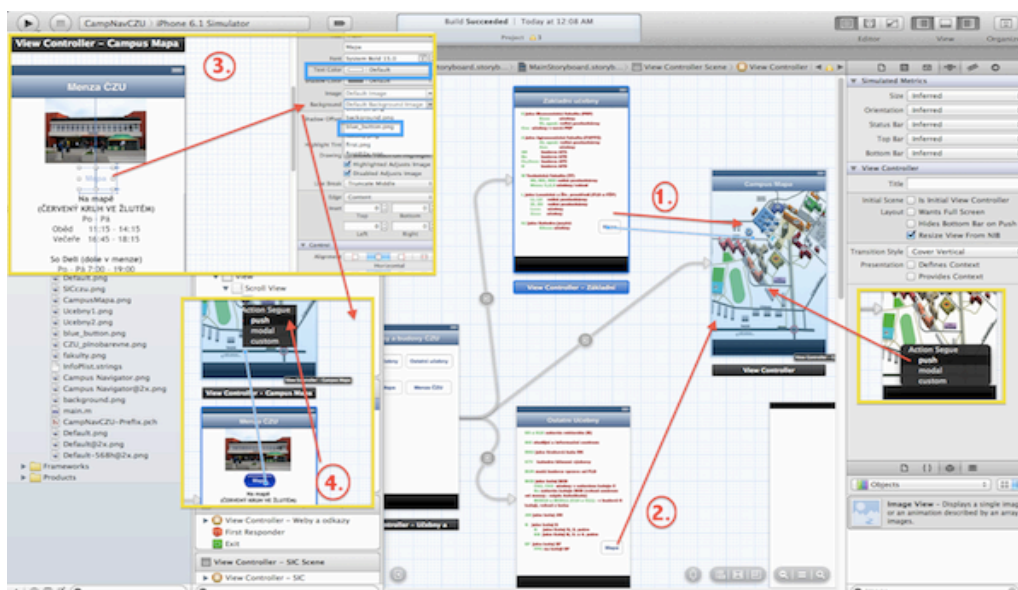
Základní funkcionalitu bychom měli, teď je ještě třeba udělat nějaké ty estetické úpravy ve vzhledu. Přesuneme se tedy na úpravu tlačítek a pozadí hlavních ovládacích obrazovek. Jako pozadí hlavních oken jsem vybral stejný základ jako pro spouštěcí obrázky, logo všech fakult ČZU a bílé okolí (viz. Obr. 7-8). Vložil jsem ho na celé okno s tlačítky odkazujícími na jednotlivé okna s informacemi a pomocí Editoru jsem poslal tento obrázek do pozadí. Okolo jsem rozmístil tlačítka. Do Projektu jsem si přenesl z Finderu můj podkladový image pro tlačítko a nechal ho zkopírovat do projektu (viz. modrý obdélník). Vzhled tlačítka změníme tak, že ho označíme, pokračujeme do jeho nastavení vpravo nahoře (viz. bod 1), tvar tlačítka nastavíme na Custom, aby se přizpůsobilo našemu podkladu, dále pak do pole background napíšeme název našeho podkladu, pokud je uložený v projektu jako v mém případě, Xcode nám po napsání „blue“ doplní automaticky název podobného souboru v projektu a tím je náš podklad s názvem „blue_bottuon.png“, požadované pozadí pro tlačítko pak pomocí klávesy „Enter“ potvrdíme. Tlačítko se změní na náš podklad, ještě je třeba změnit písmo tlačítka na bílou barvu, aby byl text dobře vidět (viz. bod 2). Stejným způsobem pokračujeme na ostatních tlačítkách, pokud máme různé podklady pro různá tlačítka, pouze vybereme rozdílné pozadí těch kterých tlačítek.

Obr. 6-8: Úprava vzhledu tlačítek



V poslední části úprav vzhledu se vrhneme na dodělení změn u druhého Tab naší aplikace a to na okno s nabídkou pro informace o učebnách (viz. Obr. 7-9). Čeká nás podobný postup, jaký jsme právě absolvovali. Na stránky s informacemi o učebnách jsem ještě dodatečně doplnil tlačítka, která nás přímo odkáží na naši offline mapu kampusu školy (viz. bod 1). Nebudeme tak muset vracet se pokaždé do hlavního menu. Tlačítko pak přetažením s přidrženou klávesou CTRL spojíme s oknem s mapou a vybereme akci „push“ pro vyvolání okna s ovládacím prvkem. Stejně tak budeme postupovat v okně s druhou částí učeben, tlačítko spojíme s offline mapou. Toto tlačítko na přímý přístup k mapě jsem přidal dodatečně i do okna s údaji o Menze ČZU. Všechna tlačítka pak upravíme na náš vlastní vzhled (viz. bod 3). Tlačítko od menzy pak propojíme s offline mapou s vlastností „push“. Všechny zbývající tlačítka doděláme podle zbytku, přearrangujem hlavní okna, aby se dostali co nejlepší možný vzhled a aplikace je skoro hotová.

Obr. 6-9: Dodatečná tlačítka



6.5 Doplnění detekce polohy do kódu

Pro možnost používání detekce polohy je třeba do našeho projektu importovat knihovnu Applu „MapKit.framework“. Tu importujeme tak, že v hlavním okně projektu (viz. Obr. 7-2), které bylo zobrazeno na začátku kapitoly ve střední části okna nahoře vybereme záložku “Build Phases” dale rozklikneme nabídku “Link Binary with Libraries”, uvidíme všechny dosavadní frameworky spojené s naším projektem, přidáme sem “MapKit.framework”. Ukázka části kódu pro zjištění a výpočet polohy zařízení:

```
// RAppDelegate.h
// CampNavCZU
//
// Created by Radek Jonáš on 2/02/13.
// Copyright (c) 2013 Radek Jonáš. All rights reserved.
//

#import <UIKit/UIKit.h>
@interface RAppDelegate : UIResponder <UIApplicationDelegate>
@property (strong, nonatomic) UIWindow *window;
-----

@synthesize window = _window;

//Vytvoříme locationManager
@synthesize locationManager=_locationManager;

#pragma mark - CLLocationManagerDelegate Methods

-(void)locationManager:(CLLocationManager *)
// Definujem locationManager

manager didUpdateToLocation:(CLLocation *)
newLocation fromLocation:(CLLocation *)oldLocation{
    NSDate* eventDate = newLocation.timestamp;
    NSTimeInterval howRecent = [eventDate timeIntervalSinceNow];
    if (abs(howRecent) < 15.0)
    {

        if(newLocation.horizontalAccuracy<35.0){

            // Horizontální přesnost <35.0 je přesná a bude stačit.
            // Nastavíme rozsah hodnot latitude a longitude a jejich přesnost.

            NSLog(@"latitude %+.6f, longitude %+.6f\n",
                newLocation.coordinate.latitude,
                newLocation.coordinate.longitude);
            NSLog(@"Horizontal Accuracy:%f", newLocation.horizontalAccuracy);

            //Je doporučeno zrusit location services, po té co dostaneme naši polohu
            [manager stopUpdatingLocation];
        }
    }
}
```


Další částí kódu, která je potřebná pro zobrazení polohy na mapě je tato následující:

```
#pragma mark - App Delegate Methods
- (BOOL)application:(UIApplication *)
application:didFinishLaunchingWithOptions:(NSDictionary *)
    launchOptions
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen]
bounds]];

    // Přepíše bod pro přizpůsobení hne po spuštění aplikace, okna s mapou.

    self.window.backgroundColor = [UIColor whiteColor];
    RJStartViewController *startVC=[[RJStartViewController alloc]
initWithNibName:@"RJStartViewController" bundle:[NSBundle mainBundle]];
    self.window.rootViewController=startVC;
    [self.window makeKeyAndVisible];

    //Pokud objekt nebyl ještě vytvořen, vytvoří ho

    if(self.locationManager==nil){
        _locationManager=[[CLLocationManager alloc] init];

        //Používáme ARC, automatické počítání referencí, nemusí tedy paměť uvolňovat my.
        _locationManager.delegate=self;

        //Included in the prompt to use location services
        _locationManager.purpose = @"Ukážem vám, kde se nacházíte"

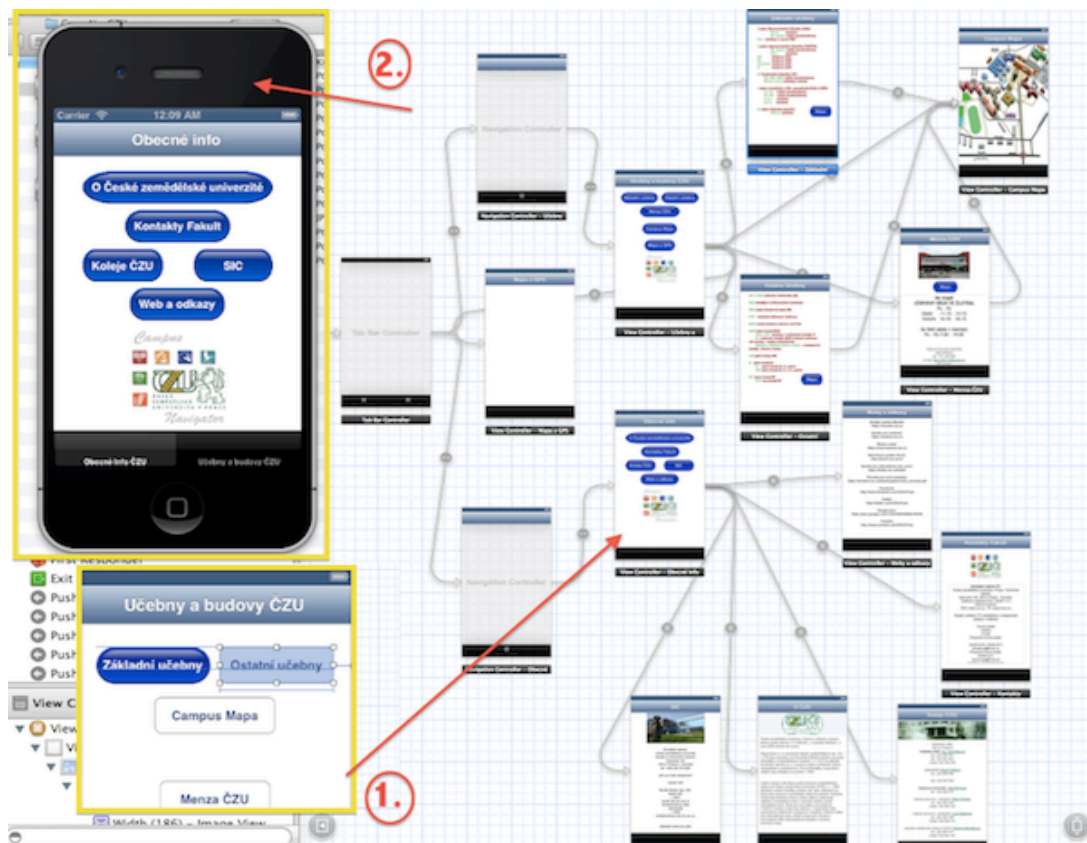
        //Určená přesnost vzdálenosti v metrech
        _locationManager.distanceFilter=500;
        self.locationManager=_locationManager;
    }

    if([CLLocationManager locationServicesEnabled]){
        [self.locationManager startUpdatingLocation];
    }

    return YES;
}
```

Naše celý projekt ve výsledném vzhledu je vidět na níže (viz. Obr. 7-10). Po dokončení posledních úprav vzhledu a tlačítek (viz. bod 1) a doděláním GPS určení polohy máme aplikaci hotovou. Spuštěná ve virtuálním emulátoru iPhone vypadá přesně jako vlevo nahoře na screenshotu (viz. bod 2). Ostatní screenshoty s ukázkou funkčnosti aplikace jsou přiloženy v tištěné Příloze č. 2 a č. 3.

Obr. 6-10: Výsledný projekt s emulací



7 Závěr

V diplomové práci jsem se snažil o to, aby po přečtení čtenář věděl o společnosti Apple zase o něco víc. Aby na vlastní oči viděl, jaké myšlenky, nestandardní řešení nástrah a hlavně využívání nových metod, jak dosáhnout svého, Steva Jobse dovedli k tomu, aby opět dopomohl k tomu, změnit nějaké odvětví od základů. Dokázal to s Macintoshem, s iPodem s iPhone a nakonec iPadem. Chtěl jsem, aby čtenář poznal i další mobilní operační systémy, konkurenci iOS. Myslím, že se mi podařilo zahrnout do práce všechny hlavní rivaly iOS a něco málo o nich informovat. Mnoho vývojářů nemá rádo programovací jazyk Objective-C, někteří ho vysloveně nesnášejí. Může to být způsobené jenom egem vývojáře, který přece už používá nejlepší jazyk a v něm taky vyvíjí. Nebo například neznalost a nechť učít se něčemu novému, co není univerzální a co využije jen na platformě Apple. Ať už je důvod jakýkoliv, jak je v práci vidět, neznámá to, že bez Objective-C nelze pro iOS vyvíjet aplikace. Uvedl jsem pár možností jak je možné obejít jazyk Objective-C. Vývojář možná nebude moci využít výhody a všechny vlastnosti iOS, ale určitě bude schopný bezproblému programovat, a ve svém známém programovacím jazyku. Pokusil jsem se o polopatické vysvětlení postupu při tvorbě aplikace, kde jsem využil mnou upravených screenshotů obrazovky, do kterých jsem zasahoval anotací a doplňoval číselné body a směry postupu. Doufám, že se mi to podařilo. Jako naprostý laik, který se vše začal učít z, převážně, anglických zdrojů, jsem se naučil hodně moc zajímavých věcí, už jen tím, že jsem hledal v nápovědách Xcode a zkoušel si vytvářet cokoli nového. Moc bych si přál, aby toto intuitivní a uživatelsky přívětivé vývojové prostředí poznalo více lidí.

Kdyby se v budoucnu třeba začalo programování pro iOS platformu na naší univerzitě vyučovat, byl by to velký krok s dobou, přece jen tato platforma dnes patří k těm nejvíc využívaným, spolu s Androidem společnosti Google. Kdyby se změnila struktura vyučovaných programovacích jazyků a zrušily se ty, které v dnešní době už nemají moc velkou budoucnost a místo nich se začalo vyučovat Objective-C. Hodně by se tím zvedla úroveň.

Samotný nápad okolo této aplikace mě velice zaujal. Vize toho, mít aplikaci, která by studentům dovolovala všechny základní věci týkající se školy řešit z jednoho místa je výborná. Je to i směr, kterým bych se chtěl vydat v možném navazujícím doktorandském studiu, kde by bylo teoreticky možné aplikaci propojit i se školními servery a pomoci tak studentům mít všechno co potřebují na jednom místě, v jedné malé aplikaci v kapse. Všechny přihlašování na předměty, na termíny zkoušek by byly možné z této aplikace. Díky jednotnému přihlašovacímu jménu a heslu by tak bylo možné se přihlásit na všechny potřebné servery. Například si moci stáhnout textový soubor se zadáním z Moodle přímo do iPhone v hodině, nebo jakékoliv další možná interakce se školními

servery by byla na dosah prstu. Toto komplexní řešení a propojení systémů by asi nebylo úplně jednoduché, přeci jen máme na univerzitě velké množství rozdílných systémů a platform, ale takový nápad mě velice zaujal a leží mi v hlavě. Vytvořit něco, co by opravdu bylo používané v praxi a někomu mohlo pomoci, to musí být příjemný pocit.

.

8 Citovaná literatura

1. **ZANDL, Patrick.** *Apple: cesta k mobilům*. 1. vyd. Praha: Mladá fronta, 2012, 269 s. ISBN 978-80-204-2641-3.
2. **ISAACSON, Walter.** *Steve Jobs*. 1st pub. United States: Simon & Schuster, 2011. ISBN 978-80-7252-352-8.
3. *Apple History* [online]. 2013 [cit. 2013-03-12]. Dostupné z: <http://www.apple-history.com/ipod>
4. *iPhone OS* [online]. [cit. 2013-03-20]. Dostupné z: <http://idaret.cz/slovník/iPhone-OS/>
5. iOS Versions. *Wikipedia.org* [online]. 2011 [cit. 2013-03-20]. Dostupné z: http://en.wikipedia.org/wiki/iOS_version_history
6. *About iPhone/iPad* [online]. 2009 [cit. 2013-03-20]. Dostupné z: http://ipod.about.com/od/iphonesoftwareterms/a/firmw_history.htm
7. iOS Jailbreaking [online]. 2011 [cit. 2013-03-21]. Dostupné z: http://en.wikipedia.org/wiki/iOS_jailbreaking
8. *Industry Leaders Announce Open Platform for Mobile Devices* [online]. Open Handset Alliance, 2007-11-05, [cit. 2011-05-02]. Dostupné z: http://www.openhandsetalliance.com/press_110507.html
9. **CLYMO, Rob a Rosemary HATTERSLEY.** *The complete guide to Android*. London: IDG Communications, 2011. ISBN 978-095-6539-045.
10. **COLLINS, Charlie, Michael D GALPIN a Matthias KAPPLER.** *Android in practice*. Shelter Island, NY.: Manning, c2012, xxii, 623 p. ISBN 19-351-8292-7.
11. *Microsoft: It was never 'Metro,' it was always 'Modern UI'*. *The Register* [online]. 2012, 8-2012 [cit. 2013-03-26]. Dostupné z: http://www.theregister.co.uk/2012/08/10/metro_is_modern_ui_now/
12. Overview and review of Windows Phone 8. *Windows Phone Central* [online]. 2012 [cit. 2013-03-26]. Dostupné z: <http://www.wpcentral.com/overview-and-review-windows-phone-8>
13. *Windows Phone*. *Microsoft Windows Phone* [online]. [cit. 2013-04-03]. Dostupné z: <http://www.windowsphone.com/cs-cz>

14. BlackBerry. *BlackBerry* [online]. 2013 [cit. 2013-03-26]. Dostupné z:
<http://global.blackberry.com/>
15. BlackBerry 10. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-03]. Dostupné z: http://en.wikipedia.org/wiki/BlackBerry_10
16. BlackBerry Reinvented. *BlackBerry* [online]. [cit. 2013-03-26]. Dostupné z:
<http://press.blackberry.com/press/2013/blackberry-10-re-designed-re-engineered-and-re-invented.html>.
17. **SWEENY**, Alastair. *BlackBerry planet: the story of Research in Motion and the little device that took the world by storm*. Mississauga, Ont.: John Wiley, c2009, xx, 284 p. ISBN 978-047-0159-408.
18. WebOS. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2013-04-03]. Dostupné z: <http://cs.wikipedia.org/wiki/WebOS>.
19. *Ubuntu* [online]. 2013 [cit. 2013-04-03]. Dostupné z: <http://www.ubuntu.com>
20. *Developing C# apps for iPhone and iPad using MonoTouch: iOS apps development for .NET developers*. New York: Apress, 2011. ISBN 978-1-4302-3174-5.
21. *Xamarin Developer Centre* [online]. 2013 [cit. 2013-04-04]. Dostupné z:
<http://docs.xamarin.com>
22. *Building iPhone apps with HTML, CSS, and JavaScript: making App Store apps without Objective-C or Cocoa*. Sebastopol: O'Reilly, 2010. ISBN 978-0-596-80578-4.
23. *HTML5: designing rich Internet applications*. Boston: Focal Press, c2010, xiv, 285 p. ISBN 978-024-0813-288.
24. *Cocoa / Richard Wentk: Developer Reference*. Indianapolis: Hoboken, 2010. ISBN 978-0-470-49589-6.
25. **MARK**, Dave a Jeff **LAMARCHE**. *iPhone SDK: průvodce vývojem aplikací pro iPhone a iPod touch*. Vyd. 1. Brno: Computer Press, 2010, 480 s. ISBN 978-80-251-2820-6; 88025128206.

9 Přehled obrázků

OBR. 3-1: iMAC	OBR. 3-2: iMAC ROZHRANÍ	6
OBR. 3-3: iTUNES 1.0 A SOUNDJAM	OBR. 3-4: iTUNES TIMELINE	8
OBR. 3-5: PŮVODNÍ IPOD	OBR. 3-6: IPOD VERZE 2.....	10
OBR. 3-7: PŘEDSTAVENÍ IPHONU 2G	OBR. 3-8: IPHONE 2G VS. IPHONE 5	13
OBR. 3-9: IPHONE OS 1	OBR. 3-10: IPHONE OS 3.2	15
OBR. 3-11: IOS FOLDERS	OBR. 3-12: IOS NOTIFICATION CENTRE	16
OBR. 4-1: IOS, ANDROID & WINDOWS PHONE		19
OBR. 4-2: BLACKBERRY HUB	OBR. 4-3: UBUNTU TOUCH	21
OBR. 5-1: NOVĚ VYTVOŘENÉ ODKAZY NA OBJEKTY V MONOTOUCH		23
OBR. 5-2: WEB APP CODE FOR BACK "HELLO" BUTTON.....		27
OBR. 6-1: XCODE - VYTVOŘENÍ PROJEKTU.....		31
OBR. 6-2: ÚVODNÍ NASTAVENÍ PROJEKTU.....		32
OBR. 6-3: IMPORT SOUBORŮ DO PROJEKTU		33
OBR. 6-4: NAVIGATION CONTROLLER		34
OBR. 6-5: PŘIDÁNÍ VIEW CONTROLLERS.....		35
OBR. 6-6: VIEW CONTROLLER A TLAČÍTKA.....		36
OBR. 6-7: OBRÁZKY A NAPLNĚNÍ INFORMACEMI		37
OBR. 6-8: ÚPRAVA VZHLEDU TLAČÍTEK.....		38
OBR. 6-9: DODATEČNÁ TLAČÍTKA.....		38
OBR. 6-10: VÝLEDNÝ PROJEKT S EMULACÍ		41

10 Seznam zkratek

Zkratky: SDK, MVC, iOS, UI, ROKR, ARC, ČZU, GPS, 3G