

Aplikace pro přenos dat z mobilních zařízení

An Application for Transferring Data from Mobile Devices

Bc. Jiří Kašpárek

Diplomová práce
2012



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jiří KAŠPÁREK**
Osobní číslo: **A10290**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Informační technologie**

Téma práce: **Aplikace pro přenos dat z mobilních zařízení**

Zásady pro vypracování:

- 1. Vytvořte literární rešerši na téma tvorby aplikací pro mobilní zařízení iPhone se zaměřením se na operační systém iOS 5.**
- 2. Navrhněte vzorovou databázi použitelnou jako jednoduchý systém CRM (Customer Relationship Management).**
- 3. Vytvořte serverovou službu zprostředkovávající komunikaci mezi klientskou aplikací a databází.**
- 4. Vlastní komunikaci mezi aplikací a serverovou službou realizujte pomocí webových služeb nebo WCF.**
- 5. Popište proces debuggování na reálném zařízení iPhone.**
- 6. Popište proces vystavení aplikace na App Store u společnosti Apple.**

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. DUNCAN, Andy. Objective-C: pocket reference [online]. O'Reilly, 2002 [cit. 2012-02-02]. ISBN 978-0-596-00423-1. Dostupné z: http://aleph.utb.cz/F/?func=service&doc_library=UTB01&doc_number=000034649&line_number=BRIEF&service_type=MEDIA
2. PIPER, Ian. Learn Xcode tools: for Mac OS X and iPhone development [online]. Apress, 2009 [cit. 2012-02-02]. ISBN 978-1-4302-7220-5. Dostupné z: <http://www.springerlink.com/content/rh6557/?p=e5e26e7ffff44e259ea934b6423d1973&pi=6>
3. ROBINSON, Simon. C : programujeme profesionálně. Vyd. 1. Brno : Computer Press, 2003. 1130 s. ISBN 80-251-0085-5.
4. SNELL, James. Programming Web services with SOAP [online]. Beijing: O'Reilly, 2002, 244 s. [cit. 2012-02-02]. ISBN 05-960-0095-2. Dostupné z: http://aleph.utb.cz/F/?func=service&doc_library=UTB01&doc_number=000034712&line_number=BRIEF&service_type=MEDIA
5. HUMMEL, Joel. LINQ: The future of data access in C 3.0 [online]. Sebastopol, Calif.: O'Reilly, 2006 [cit. 2012-02-02]. ISBN 978-059-6528-416. Dostupné z: http://aleph.utb.cz/F/?func=service&doc_library=UTB01&doc_number=000035100&line_number=BRIEF&service_type=MEDIA
6. AITCHISON, Alastair a Adam MACHANIC. Expert SQL server 2008 development [online]. New York: Apress, 2009, 433 s. [cit. 2012-02-02]. ISBN 978-143-0272-120. Dostupné z: <http://www.springerlink.com/content/u86n18/?p=b72a5606fba74f39970535542c9f2655&pi=6>

Vedoucí diplomové práce:

Ing. Jiří Pálka, Ph.D.

Ústav elektroniky a měření

Datum zadání diplomové práce:

24. února 2012

Termín odevzdání diplomové práce:

21. května 2012

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Roman Jašek, Ph.D.
ředitel ústavu

ABSTRAKT

Hlavním cílem této diplomové práce je vytvořit mobilní aplikaci pro operační systém iOS 5. Aplikace je vícevrstvá a pracuje na principu klient-server. Zdrojová data aplikace jsou uložena v relační databázi na Microsoft SQL Serveru. Struktura databáze odpovídá jednoduchému CRM systému. Komunikaci mezi klientskou aplikací a databázovým serverem zprostředkovává webová služba. Služba je napsaná pomocí technologie WCF a programovacího jazyka C Sharp. Klientská aplikace má za úkol zpřístupnit důležitá data CRM systému v moderním a přehledném uživatelském rozhraní. Aplikace je napsaná v jazyce Objective-C a je určena pro chytré telefony iPhone s operačním systémem iOS 5.0 a vyšším. Součástí diplomové práce je také popis procesů debuggování vytvořené aplikace na reálném iPhone a následné vystavení aplikace na App Store.

Klíčová slova: Mobilní aplikace, webová služba, CRM, iPhone

ABSTRACT

The main objective of this thesis is to create mobile application for operating system iOS 5. The application is multi-layer and uses client-server principle. Source data of application is stored in Microsoft SQL Server relational database. The database structure corresponds to a simple CRM system. Communication between client application and database server provides a web service. The service is written using WCF technology and C Sharp programming language. The client application is responsible for access to important data in CRM system and uses modern and clear user interface. The application is written in Objective-C programming language and designed for iPhone smartphones with operational system iOS 5.0 and higher. The thesis also describes processes of debugging application on real iPhone and subsequent publishing of application to App Store.

Keywords: Mobile application, web service, CRM, iPhone

Na tomto místě bych rád poděkoval svému vedoucímu diplomové práce Ing. Jiřímu Pálkovi, Ph.D. za odbornou pomoc, cenné rady a čas, které mi věnoval. Také bych chtěl poděkovat své rodině za podporu, tvorbu potřebného zázemí a trpělivost při mém studiu na této škole.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČÁST	11
1 IPHONE	12
1.1 HISTORIE	12
1.2 HARDWAROVÉ VYBAVENÍ	15
1.3 SOFTWAREOVÉ VYBAVENÍ.....	15
2 POUŽITÉ TECHNOLOGIE A SOFTWARE.....	17
2.1 OBJECTIVE-C.....	17
2.2 COCOA	17
2.3 MODEL-VIEW-CONTROLLER	18
2.4 XCODE	19
2.4.1 Interface Builder.....	20
2.4.2 LLVM Compiler a LLDB Debugger	20
2.4.3 iOS Simulator.....	21
2.5 WSDL2OBJC	21
2.6 iTunes CONNECT	22
2.7 MICROSOFT .NET FRAMEWORK.....	22
2.7.1 Historie	23
2.7.2 Common Language Infrastructure.....	24
2.7.3 Správa paměti.....	25
2.7.4 Zabezpečení.....	26
2.8 C SHARP.....	26
2.8.1 LINQ	27
2.9 MICROSOFT VISUAL STUDIO.....	27
2.9.1 Editor kódu.....	28
2.9.2 Designer	28
2.9.3 Debugger	29
2.10 WINDOWS COMMUNICATION FOUNDATION	30
2.11 SOAP.....	31
2.12 INTERNETOVÁ INFORMAČNÍ SLUŽBA	31
2.13 SQL	33
2.13.1 Transact-SQL	34
2.14 MICROSOFT SQL SERVER.....	34
2.14.1 Relační databáze.....	35
2.14.2 Microsoft SQL Server Management Studio	35

2.15	DIGITÁLNÍ CERTIFIKÁT	36
2.16	SECURE HASH ALGORITHM	37
2.17	CUSTOMER RELATIONSHIP MANAGEMENT	37
II	PRAKTICKÁ ČÁST	39
3	RELAČNÍ DATABÁZE	40
3.1	TABULKY	40
3.1.1	AuditTrail	41
3.1.2	Resources	41
3.1.3	Users	42
3.1.4	Content	43
3.1.5	Parties	43
3.1.6	PartyAddresses	44
3.1.7	PartyBankAccounts	44
3.1.8	PartyContacts	45
3.1.9	Clients	45
3.1.10	Contracts	46
3.1.11	Files	47
3.1.12	Relations	47
3.1.13	Vendors	48
3.1.14	Tickets	49
3.2	PROCEDURY	49
4	WEBOVÁ SLUŽBA	52
4.1	VEŘEJNĚ DOSTUPNÉ FUNKCE	52
4.2	ZABEZPEČENÍ	53
4.3	LOKALIZACE A CACHE	54
4.4	DEMO UŽIVATEL	55
4.5	NASTAVENÍ SLUŽBY	55
5	MOBILNÍ APLIKACE	57
5.1	STRUKTURA	57
5.2	KOMUNIKACE S WEBOVOU SLUŽBOU	57
5.3	ZABEZPEČENÍ	58
5.4	GRAFICKÉ UŽIVATELSKÉ ROZHRANÍ	59
5.4.1	Přihlašovací obrazovka	61
5.4.2	Přehled	61
5.4.3	Provozní informace	62
5.4.4	Zprávy	63
5.4.5	Detail zprávy	63
5.4.6	CRM	64
5.4.7	Prodejci	64
5.4.8	Detail prodejce	65
5.4.9	Klienti	66
5.4.10	Detail klienta	67

5.5	LOKALIZACE.....	67
5.6	NASTAVENÍ APLIKACE	68
6	DEBUGGOVÁNÍ MOBILNÍ APLIKACE	70
7	PUBLIKOVÁNÍ MOBILNÍ APLIKACE	74
	ZÁVĚR	78
	ZÁVĚR V ANGLIČTINĚ.....	79
	SEZNAM POUŽITÉ LITERATURY.....	80
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	84
	SEZNAM OBRÁZKŮ	87
	SEZNAM TABULEK.....	89

ÚVOD

Chytré telefony jsou v současné době na vzestupu a pro mnoho lidí jsou stejně důležitým pracovním nástrojem jako osobní počítač. Dokáží totiž nejen volat a posílat textové zprávy, ale i posílat e-maily, plánovat čas, navigovat, ovládat na dálku firemní budovy a mnohé další užitečné činnosti. Proto také stále více roste poptávka po tvorbě nových mobilních aplikací, které například doplní stávající systém o nové možnosti práce s ním. Příkladem mohou být české banky, které jedna po druhé postupně začínají nabízet mobilní aplikace pro ovládání jejich bankovních účtů. Proto jsem se při výběru tématu své diplomové práce rozhodl pro vytvoření mobilní aplikace, která bude využívat jednu z nejpoužívanějších mobilních platforem.

Aplikaci jsem pojal tak, jako bych vytvářel novou mobilní aplikaci k již fungujícímu systému. Ze stávajícího systému pak aplikace využívá jeho relační databázi a informace, které jsou v ní uloženy. K databázi přidává nové procedury, ale strukturu tabulek nijak nemění. V případě této práce má databáze podobu CRM systému, který by byl součástí mnohem větší databáze.

Architektura aplikace je vícevrstvá a skládá se celkem z 3 vrstev. První vrstvou je již zmíněná relační databáze, která uchovává informace uvnitř systému. Prostřední vrstvou je webová služba, která zprostředkovává komunikaci mezi databází a mobilním klientem. Služba je samostatná a nemá žádné vazby na ostatní služby v systému. V budoucnu ji proto bude možné využít i pro ostatní mobilní klienty, kteří poběží na jiných operačních systémech jako například Android nebo Windows Phone. Poslední vrstvou je samotná mobilní aplikace, která je vytvořena v API Cocoa, které je určené pro vývoj aplikací pro iPhone.

I. TEORETICKÁ ČÁST

1 IPHONE

iPhone je řada chytrých telefonů vytvořená společností Apple Inc. Tyto chytré telefony v sobě spojují funkce mobilního telefonu s digitálním fotoaparátem, multimediálního přehrávače iPod a zařízení pro mobilní komunikaci s Internetem. iPhone přinesl zcela nové uživatelské rozhraní, založené na dotykovém displeji s technologií multi-touch, ve spojení s inovativním operačním systémem, jenž umožňuje uživatelům ovládat telefon pouhým pohybem prstů na obrazovce. Díky své inovativnosti a také díky skvělému marketingu společnosti Apple dokázal iPhone v posledních 5 letech často určovat směr vývoje v oblasti chytrých telefonů.

1.1 Historie

Vývoj iPhone začal v roce 2005, když inženýři Apple začali zkoumat dotykové displeje. Velmi brzy pak byl představen první iPhone 2G, který se začal prodávat již 29. června 2007. Jeho vývoj trval přibližně 30 měsíců a stál okolo 150 milionů amerických dolarů. iPhone 2G změnil mobilní svět a od té doby se výrobci zaměřili na dotykové displeje ovládané prsty. iOS se stal symbolem pro operační systém přizpůsobený pro ovládání prsty. Největší konkurent se objevil až v roce 2008 a tím byl Google Android. V době svého uvedení na trh iPhone neuměl spoustu věcí a stal se terčem kritiky. V té době neexistovalo nic jako App Store, což je obchod s aplikacemi pro chytré telefony iPhone, a jediný způsob jak dostat aplikace do iPhone bylo pomocí složité cesty nebo internetového prohlížeče Safari. iPhone se první dva měsíce prodával ve třech variantách, které se lišily pouze velikostí paměti.



Obrázek 1: iPhone 2G [15]

9. června 2008 byla na celosvětové konferenci vývojářů WWDC 2008 představena nová verze iPhone s označením iPhone 3G. Mezi její nové vlastnosti patří podpora UMTS sítí, GPS navigace a drobné změny v hardwaru. Zvýšila se také výdrž akumulátoru. Na této akci byla rovněž předvedena finální verze SDK a firmwaru 2.0, který je určen jak pro iPhone 3G, tak pro první model telefonu a také mezitím uvedený multimediální přehrávač iPod Touch. Seznam zemí, kde se iPhone prodává, se rozšířil na 70 a je mezi nimi i Česko.

Nový iPhone 3GS byl při vnějším pohledu totožný s modelem 3G a oproti svému předchůdci dostal pouze několik hardwarových vylepšení. Všechna jsou ale vítanými položkami ve specifikaci telefonu, který musí čelit stále početnější konkurenci mobilních telefonů s dotykovým displejem. Zásadní proměnou prošel fotoaparát, který konečně dokáže pořizovat videonahrávky. Přibyl také kompas, který pomůže při hledání správné cesty a nechybí ani hlasové ovládání, které se hodí například při řízení. Vylepšena byla výdrž a iPhone 3GS se prodával i s větší vestavěnou pamětí 32GB. Evolucí prošla i podpora HSDPA, která nyní nabízí rychlejší datové přenosy. Na druhou stranu je novinka dražší než původní model 3G.



Obrázek 2: iPhone 3G(S) [16]

iPhone 4 byl již zcela nový model jak designem, tak i hardwarem a začal se prodávat v březnu roku 2011. Přináší novou verzi operačního systému iOS 4, který nově podporuje multitasking. iOS 4 je dostupný i pro starší model 3G, ale v jeho případě musí majitelé oželeť některé funkce. Telefon dostává zcela nový „Retina“ displej, který má dvojnásobné rozlišení oproti předchozím modelům. Fotoaparát je opět vylepšen a přibyl také gyroskop nebo LED blesk k fotoaparátu.

Aktuálně poslední iPhone 4S má stejný vzhled jako iPhone 4, ale především jiný hardware uvnitř telefonu. Mezi hlavní novinky patří nový procesor A5, který je 2x rychlejší než u iPhone 4 a má 7x rychlejší grafický procesor. Další zajímavou novinkou je vylepšený 8 megapixelový fotoaparát, který má kvalitní čočku pro fotografování v noci a umí nahrávat videa ve vysokém rozlišení. Další novinky jsou softwarové a to AirPlay, který umožňuje bezdrátové streamování multimediálního obsahu, nová verze operačního systému iOS 5 a největší novinkou je aplikace SIRI, která plní funkci hlasové asistentky.

[1]



Obrázek 3: iPhone 4(S) [17]

1.2 Hardwarové vybavení

Poslední model iPhone 4S má plastové tělo, jehož okraje tvoří 2 nerezové antény, mezi kterými je možno přepínat. Po obvodu těla telefonu jsou rozmístěny vstupní konektory a ovládací tlačítka. Přední stranu telefonu tvoří velký displej o velikosti 3,5 palce a rozlišení 960x640 pixelů. Displej je typu IPS TFT LCD a je podsvícený LED diodami. Součástí displeje je také snímací plocha, která dokáže snímat několik dotyků současně. Pod displejem je pouze hlavní ovládací tlačítko a nad ním je umístěná přední kamera a reproduktor. Na zadní straně je umístěn fotoaparát s rozlišením 8 megapixelů a jeho přisvětlovací dioda. Fotoaparátem je také možné pořizovat video až v rozlišení 1080 řádků a to při 30 snímcích za sekundu. Uvnitř telefonu je výkonný dvoujádrový procesor Apple A5 pracující na 800 MHz. S ním spolupracuje grafický koprocessor PowerVR SGX543MP2. K dispozici je 512 MB RAM a 16 až 64 GB úložné paměti. Kromě klasických telefonních sítí má telefon zabudovanou podporu pro Bluetooth, GPS a Wi-Fi. Zabudované má senzory přiblížení a okolního osvětlení, tříosý gyroskop a akcelerometr. O napájení celého telefonu se stará lithium-iontová baterie. [2]

1.3 Softwarové vybavení

iPhone 4S aktuálně běží na operačním systému iOS 5, který je plně optimalizovaný pro dotykové ovládání prsty. Kromě něj jsou v základu přeinstalované i další důležité aplikace. Samozřejmě jsou to všechny důležité aplikace jako volání, správa kontaktů a posílání SMS, které musí obsahovat každý telefon. Kromě nich ale také posílání mailů, Internetový prohlížeč Safari, diktafon, poznámky nebo kalkulačka. Z multimediálních aplikací je možné zmínit fotoaparát a videokamera, galerie fotografií a videí, hudební přehrávač nebo mapy.

Jednou z novinek iOS 5 je aplikace iMessage. Ta umožňuje posílat nejen obyčejné textové zprávy, ale také fotografie nebo videa. iMessage je dostupné na všech zařízeních s iOS a také se samo synchronizuje mezi všemi zařízeními daného uživatele.

Další novinkou je aplikace Siri (Obr. 4), která plní funkci inteligentní osobní asistentky, se kterou uživatel komunikuje v přirozeném jazyce anglicky, německy, francouzsky nebo japonsky. Siri dokáže posílat textové zprávy, volat, plánovat schůzky, vyhledat a navigovat k požadovanému místu a mnoho dalšího. Bohužel i přesto, že byla

Siri vydána společně s iPhone 4S, je stále ve fázi beta verze a proto ji aktuálně nelze považovat za plně funkční.



Obrázek 4: Aplikace Siri [18]

Ostatní aplikace, které nejsou předinstalované, je možné velmi jednoduše doinstalovat přes App Store (Obr. 5). Ten slouží jako hlavní distribuční kanál všech aplikací pro iOS. Po zadání nezbytných údajů je možné přes App Store nakupovat i placené aplikace. Aktuálně obsahuje přes půl milionu placených i neplacených aplikací všeho druhu. Všechny aplikace, které si uživatel stáhne, jsou navíc uloženy v cloudovém úložišti iCloud společnosti Apple, takže jsou uživateli vždy k dispozici. [2]



Obrázek 5: App Store [19]

2 POUŽITÉ TECHNOLOGIE A SOFTWARE

2.1 Objective-C

Objective-C je objektově orientovaný programovací jazyk, který je postaven na programovacím jazyce C. Jeho rozšíření nad C jsou většinou založeny na Smalltalku, jednom z prvních objektově orientovaných programovacích jazyků. Objective-C je navržen tak, aby jednoduchým a přímočarým způsobem přidal k C objektově orientované programovací schopnosti. Vzhledem k tomu, že je Objective-C nadmnožinou jazyka C, je možné přeložit libovolný program jazyka C kompilátorem Objective-C. Kompilátor tohoto jazyka je součástí kolekce kompilátorů GCC.

Primárně se využívá k vývoji pro počítače Mac s operačním systémem OS X a zařízení jako iPhone nebo iPad s operačním systémem iOS. Původně byl hlavním programovacím jazykem pro operační systém NeXTSTEP, který společnost Apple koupila a použila jako základ dnešního OS X, což vysvětluje, proč se dnes primárně využívá právě k vývoji pro operační systémy společnosti Apple.

Model objektově orientovaného programování v Objective-C je založen na zasílání zpráv instanci objektu. Tím se výrazně liší od běžně rozšířeného jazyka C++, kde se volají metody třídy. Rozdíl mezi těmito dvěma koncepcemi je v tom, jak se prováděný kód odkazuje na metody a zprávy. V jazyce C++ je název metody ve většině případů vázáný na částí kódu v cílové třídě kompilátorem. V Objective-C se cíl zprávy řeší za běhu a cílový objekt si zprávu sám interpretuje. Metoda odpovídající zprávě je identifikována selektorem, což je textový řetězec, který určuje její jméno. Důsledkem toho je, že zpráva procházející systémem nemá žádnou kontrolu typů. Objekt, který přijímá zprávu, tak nemůže zaručit, že na zprávu odpoví a v takovém případě pouze vyvolá výjimku. [3]

2.2 Cocoa

Cocoa je hlavní objektově orientované rozhraní pro programování aplikací společnosti Apple. Programují se pomocí něj aplikace pro operační systém OS X. Související API Cocoa Touch obsahuje navíc rozpoznávání gest, animace a jiné knihovny uživatelského rozhraní, a je určeno pro aplikace operačního systému iOS, který se používá v zařízeních jako je iPhone, iPod Touch a iPad.

Cocoa aplikace jsou typicky vyvinuty v jazyce Objective-C s využitím vývojových nástrojů poskytovaných společností Apple, konkrétně Xcode a Interface Builder. Nicméně je možné k API Cocoa přistupovat i pomocí dalších nástrojů jako jsou LispWorks, Object Pascal, Python, Perl nebo Ruby.

API Cocoa se skládá z knihoven Objective-C, některých knihoven jazyka C a 3 hlavních frameworků Foundation Kit, Application Kit a Core Data frameworks.

Foundation Kit je obecná objektově orientovaná knihovna poskytující řetězce, kontejnery a iterátory, distribuované výpočetní techniky, běhové smyčky a další funkce, které nejsou přímo vázány na grafické uživatelské rozhraní. Předpona „NS“, která se používá pro všechny třídy a konstanty frameworku, pochází API OpenStep, které je předkem Cocoa.

Application Kit pochází přímo z původního NeXTSTEP a obsahuje třídy, pomocí kterých se vytváří grafické uživatelské rozhraní aplikací. Application Kit je postaven nad Foundation Kit a používá stejnou předponu „NS“. V Cocoa Touch má svůj ekvivalent pod jménem UIKit.

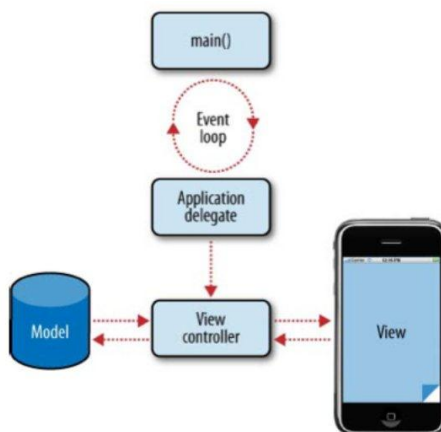
Core Data frameworks vytváří vysokoúrovňový model dat, který popisuje entity a jejich vztahy. Data umožňuje serializovat do XML, binárních souborů nebo SQLite databází. S daty lze manipulovat pomocí objektů vyšší úrovně, zastupující entity a jejich vztahy. Programátor se tak vůbec nemusí starat o to, jak jsou data načítána nebo ukládána. Tyto objekty lze také přímo navázat na prvky uživatelského rozhraní bez nutnosti psaní jakéhokoli kódu v pozadí a tím je zpřístupnit uživatelům. [4]

2.3 Model-View-Controller

Model-View-Controller je softwarová architektura, která odděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé z nich má jen minimální vliv na ostatní. Model reprezentuje informace, s nimiž aplikace pracuje. Jsou to například soubory na pevném disku, tabulky v operační paměti nebo data v relační databázi. View převádí data reprezentovaná modelem do podoby vhodné k interaktivní prezentaci uživateli a proto reprezentuje grafické uživatelské rozhraní aplikace. Controller reaguje na události, které typicky pocházející od uživatele, a

zajišťuje změny jak v Modelu, tak ve View. Je tedy jakousi ústřední výkonnou jednotkou, která se stará o provázání funkčnosti celé aplikace.

Na architektuře MVC je založeno kromě jiných také API Cocoa a design všech aplikací pro OS X i iOS se touto architekturou striktně řídí. [4]



Obrázek 6: Architektura MVC [20]

2.4 Xcode

Xcode je integrované vývojové prostředí, které obsahuje všechny nástroje potřebné k tvorbě a spravování OS X a iOS projektů a zdrojových souborů, spouštění a ladění zdrojových kódů a sestavování spustitelných aplikací ze zdrojových kódů. Obsahuje pokročilý textový editor, který podporuje doplňování zdrojového kódu, obarvování syntaxe, skládání zdrojového kódu, které slouží k dočasnému skrytí bloků kódu, řádkové anotace chyb, varování a poznámky. [5]



Obrázek 7: Xcode [21]

2.4.1 Interface Builder

Interface Builder je nástroj pro vizuální tvorbu uživatelského rozhraní a je kompletně integrovaný do Xcode. Pomocí Interface Builderu sestavuje programátor aplikaci přetahováním již hotových komponent na jejich místo v okně. Komponenty zahrnují standardní systémové ovládací prvky, jako jsou přepínače, textová pole, tlačítka a mnohé další.

Výběrem souboru Interface Builderu v projektu se otevře editor uvnitř Xcode, který na levé straně zobrazuje vytvářené view aplikace a v pravém panelu jsou inspektory rozhraní, stejně jako knihovna ovládacích prvků uživatelského rozhraní a objektů. Pokud je editor přepnutý do „split view“, je možné přetáhnout propojení přímo z uživatelského rozhraní designu do zdrojového kódu. Xcode také vytvoří nový outlet nebo akci pokud je propojení přetaženo na prázdné místo ve zdrojovém souboru. [5]

2.4.2 LLVM Compiler a LLDB Debugger

LLVM je kompilátor nové generace založený na open source projektu LLVM.org. Je velmi rychlý a dokáže zkompilevat kód dvakrát rychleji než původní GCC kompilátor. Zároveň produkuje aplikace, které jsou lépe optimalizované a běží rychleji. Kompilátor podporuje jazyky C, Objective-C a C++.

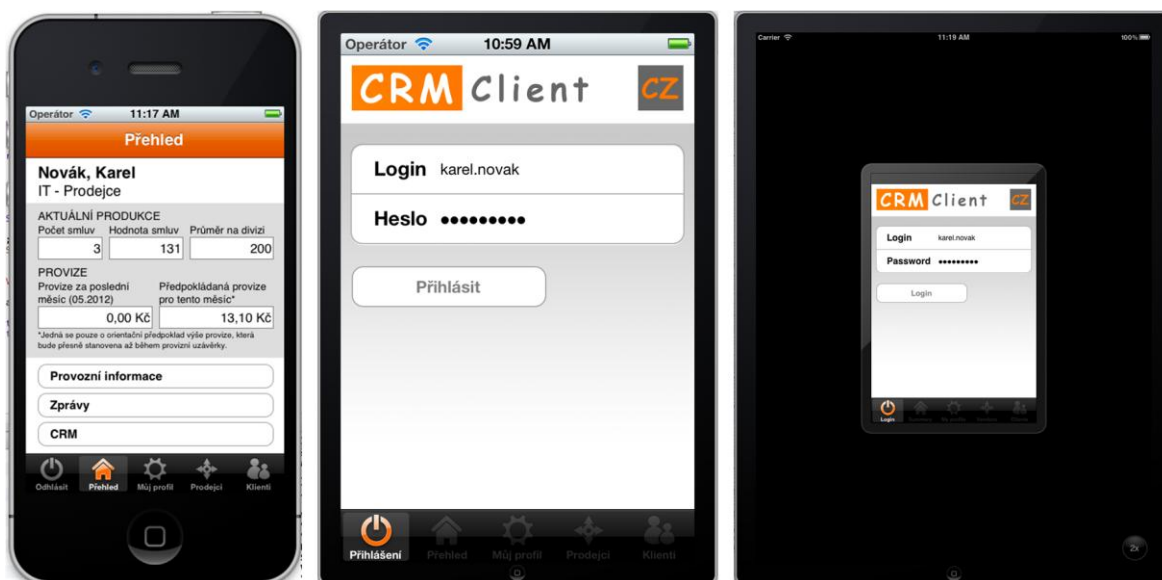
Zároveň s novým kompilátorem přišel také nový LLDB Debugger. Ten byl také od základů přepracován tak, aby spotřeboval méně paměti a běžel rychleji. Za běhu aplikace zobrazuje výpis zásobníku a přímo ve zdrojovém kódu hodnoty jednotlivých proměnných. Debugger je uzpůsoben také pro debuggování vícevláknových aplikací. Umožňuje například uzamknout se za běhu aplikace do jednoho vlákna a poté sledovat průběh jeho vykonávání.

Zvýrazňování syntaxe, doplňování kódu a všechny ostatní funkce založené na indexování zdrojového kódu jsou zpracovány LLVM parserem. Ten na pozadí nepřetržitě zpracovává zdrojový kód již při editaci, a tak mu Xcode rozumí stejně dobře jako při vytváření aplikace. Díky tomu editor hned po napsání kódu upozorňuje na překlepy, syntaktické chyby i běžné omyly v kódu. Kromě toho také Xcode rovnou nabízí automatickou opravu některých nalezených chyb a omylů. Statický analyzátor zároveň

prochází tisíce možných cest ve zdrojovém kódu a hledá místa, kde by se i platný kód choval neočekávaně, jako například u chyb přidělení paměti. [5]

2.4.3 iOS Simulator

iOS Simulator je aplikace simulující činnost reálného iPhone nebo iPad v okně počítače. S aplikací je možno komunikovat pomocí klávesnice nebo myši a dokáže simulovat dotyky prstu, gesta, zmáčknutí tlačítek daného zařízení, rotace zařízení, varování a různé další akce. Dokáže také simulovat různé verze iOS i různé rozlišení zařízení. Debuggování v Xcode se provádí právě tímto simulátorem vždy v případě, pokud není zvoleno reálné zařízení připojené k počítači.



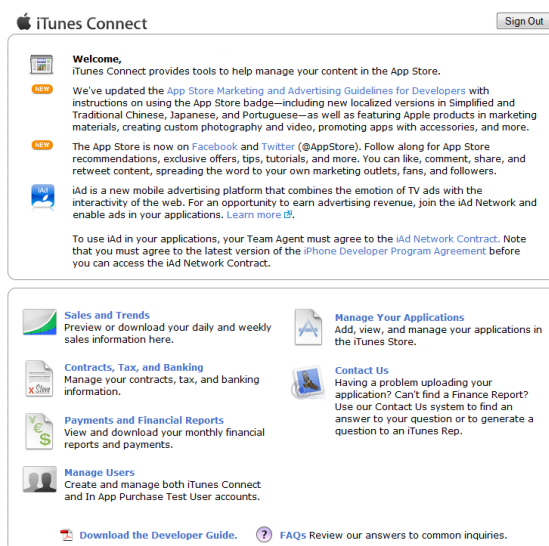
Obrázek 8: iPhone simulátor v nízkém a vysokém rozlišení a iPad simulátor

2.5 wsdl2objc

wsdl2objc je generátor zdrojových kódů pro komunikaci s webovou službou. Generátor je vytvořený v API Cocoa a jím vygenerovaný kód je v jazyce Objective-C. Podporuje webové služby, které komunikují pomocí protokolu SOAP a ke generování využívá WSDL soubory dané webové služby. Vygenerovaný kód obsahuje všechny výstupní objekty webové služby, objekty vytvářející SOAP zprávy a zajišťující komunikaci s webovou službou a parsery, které z odpovědí webové služby vytvoří instance výstupních objektů. Poslední verze wsdl2objc, která je dostupná pouze ve formě zdrojových, podporuje i nejnovější verzi iOS 5.1 včetně ARC.

2.6 iTunes Connect

iTunes Connect je sada webových nástrojů, vytvořených společností Apple, pro vývojáře, kteří vytváří aplikace pro jejich operační systémy. Aplikace umožňuje nahrávat a spravovat aplikace určené k prodeji na App Store. iTunes Connect je rozdělen do modulů, které tvoří jednotlivé funkce aplikace a přístup k nim může hlavní vývojář omezit pomocí přístupových práv. Pomocí těchto modulů pak může uživatel vytvářet nové aplikace a spravovat jejich metadata a binární data, prohlížet si stavy aplikací, spravovat prodej aplikací a prohlížet si reporty o jejich prodeji, prohlížet si nahlášené pády aplikace a různé další úkony spojené s prodejem aplikací.

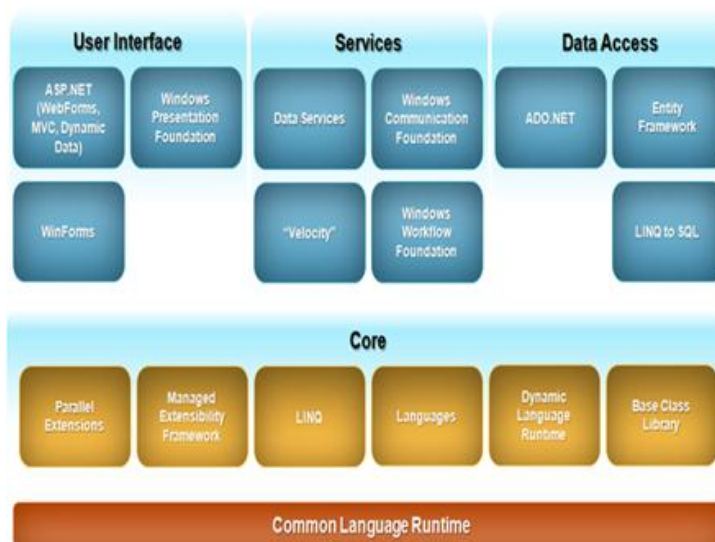


Obrázek 9: iTunes Connect

2.7 Microsoft .NET Framework

Microsoft .NET Framework je softwarový balík, který byl vyvinut společností Microsoft a je určen pro počítače s operačním systémem Microsoft Windows. Obsahuje velké množství knihoven včetně hotových řešení pro běžné programátorské problémy a také prostředí pro běh aplikací vytvořených za pomoci Microsoft .NET Frameworku. Framework nabízí širokou škálu funkcí určených k vytváření uživatelského rozhraní, přístup k datům, připojení k relačním databázím, šifrování dat, vývoji webových aplikací, numerické algoritmy, vývoj vícevláknových aplikací, zjednodušení komunikace v síti a mnoho dalších funkcí. Kromě běžné verze obsahuje rodina .NET Framework také verzi .NET Compact Framework určenou pro vývoj aplikací pro mobilní telefony a embedded

zařízení a verzi .NET Micro Framework určenou pro zařízení se značně omezenými zdroji. Nejpoužívanější programovací jazyky pro vývoj .NET aplikací jsou C Sharp, Visual Basic .NET a Delphi. K dispozici je nicméně řada dalších programovacích jazyků. [6]



Obrázek 10: Struktura .NET Framework [22]

2.7.1 Historie

Vývoj .NET Frameworku začal během 90.let 20.století ve společnosti Microsoft jako projekt s názvem Next Generation Windows Services. První beta verze však byla vydána až během roku 2000 a to již pod názvem .NET Framework 1.0. Poté již následovalo pravidelné vydávání finálních verzí .NET Framework:

- ❖ 1.0 byla uvedena v roce 2002, přináší vývojové prostředí Visual Studio .net a programovací jazyk C Sharp 1.0.
- ❖ 1.1 byla uvedena v roce 2003 a přináší vývojové prostředí Visual Studio 2003.
- ❖ 2.0 přinesla zásadní změny a byla uvedena v roce 2005, přináší nové verze jazyků C Sharp 2.0 a Visual Basic .NET 8.0 a vývojové prostředí Visual Studio 2005.
- ❖ 3.0 je pouze nadstavbou nad verzí 2.0, byla uvedena v roce 2007 a přináší WCF, WPF, WF a CardSpace.
- ❖ 3.5 byla také uvedena v roce 2007 a je opět pouze nadstavbou nad verzí 2.0, přináší však nové verze jazyků C Sharp 3.0 a Visual Basic .NET 9.0 a vývojové prostředí Visual Studio 2008.

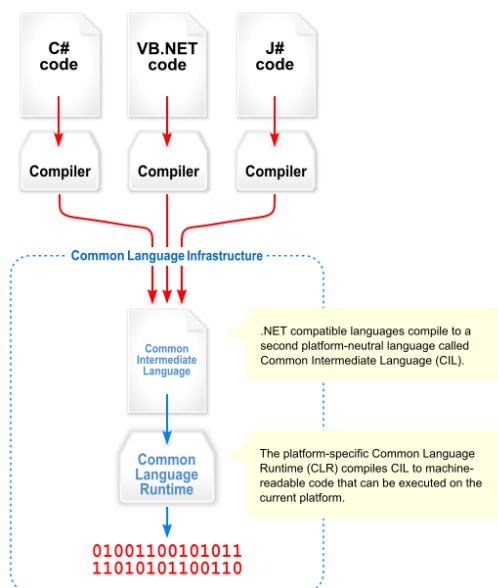
- ❖ 4.0 je již plnohodnotná nová verze a byla uvedena v roce 2010, přináší nové verze jazyků C Sharp 4.0, Visual Basic .NET 10.0, vývojové prostředí Visual Studio 2010 a zaměřuje se hlavně na vícejádrové systémy a s nimi spojené vícevláknové programování.
- ❖ 4.5 byla uvedena v roce 2011 a aktuálně je stále ve fázi beta verze, přináší novou verzi jazyka C Sharp 5.0, vývojové prostředí Visual Studio 11 a její hlavní novinkou je podpora vývoje aplikací v novém Metro stylu.

2.7.2 Common Language Infrastructure

Common Language Infrastructure je jazykově neutrální platforma pro vývoj aplikací v prostředí .NET Framework. CLI se stará o vykonávání zdrojového kódu, zpracování výjimek, úklid operační paměti a zabezpečení.

Všechny podporované jazyky jsou v tomto prostředí překládány do univerzálního jazyka Common Intermediate Language. CIL patří na nejnižší vrstvu člověkem čitelných programovacích jazyků. Je to objektově orientovaný jazyk založený pouze na práci se zásobníkem. CIL instrukce jsou pomocí Just-In-Time kompilátoru překládány do instrukcí specifických pro danou platformu. JIT kompilace překládá část kódu aplikace do nativního strojového kódu až za běhu aplikace, čímž se zrychluje provádění kódu aplikace.

Další součástí CLI je Common Language Runtime. To je softwarové prostředí, ve kterém běží všechny aplikace napsané pro .NET Framework. CLR poskytuje aplikaci abstrakci virtuálního stroje, díky kterému programátor nemusí brát v úvahu vlastnosti konkrétního systému, na kterém bude program spouštěn. CLR také poskytuje další důležité služby jako je zabezpečení, správa paměti nebo zpracování výjimek. [6]



Obrázek 11: Schéma CLI [23]

2.7.3 Správa paměti

Správa paměti v .NET Frameworku je automatická a sama si paměť hlídá a určuje, kdy může být uvolněna. Tím programátorům dovoluje se o správu paměti vůbec nestarat. Paměť se alokuje nad spravovanou haldou z .NET objektů a u každého objektu si správa paměti hlídá, kolik je na ni přímých odkazů. Pokud neexistuje žádný odkaz na objekt, a tím pádem je objekt nedosažitelný, přichází na řadu Garbage collector. Ten běží v samostatném vlákne a pravidelně prochází dostupnou paměť a uvolňuje nedosažitelnou paměť zpět k použití. To kdy bude daná paměť uvolněna, však nelze předvídat, protože se Garbage collector spouští až po použití určitého množství paměti nebo při nedostatku paměti v systému.

Garbage collector se také chová generačně. Všechny objekty jsou rozřazeny do 3 generací. Nově vytvořené objekty patří do generace 0. Objekty, které přežijí odstranění nepotřebné paměti, jsou posunuty vždy o jednu generaci výše. Garbage collector pak kontroluje objekty s vyšší generací méně často než objekty s nižší generací. To pomáhá zvýšit efektivitu úklidu paměti, protože starší objekty mívají delší životnost než novější objekty. [6]

2.7.4 Zabezpečení

Bezpečnostní mechanismus v .NET Frameworku využívá 2 vlastnosti k zajištění bezpečnosti. První je validace a verifikace zdrojového kódu a druhou je Code Access Security. Ta má za úkol zabránit nedůvěryhodnému kódu vykonat privilegovanou akci. Zabezpečení přístupu kódu je založeno na důkazech, který jsou spojeny s konkrétní sestavou. Zabezpečení používá tyto důkazy pro stanovení oprávnění udělených danému kódu. Kód, který provádí privilegovanou akci, musí vždy provést dotaz na oprávnění. [6]

2.8 C Sharp

C Sharp je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý společností Microsoft společně s platformou Microsoft .NET Framework. Lze využít k tvorbě webových aplikací, webových služeb, formulářových aplikací ve Windows, databázových aplikací, aplikací pro mobilní zařízení a různých dalších aplikací.

Jazyk částečně vychází z jazyků C++ a Java. C Sharp je silně typový, citlivý na velká a malá písmena a neexistují v něm vícenásobná dědičnost ani globální proměnné a metody. Protože není důležité, v jakém pořadí deklarujeme proměnné a metody, neobsahuje dopřednou deklaraci. Je také typově bezpečnější, protože jediné výchozí implicitní typové konverze jsou takové, které jsou považovány za bezpečné. Jsou to například rozšiřování Integerů z 32b na 64b nebo konverze z odvozeného typu na typ rodičovský. Neexistuje však implicitní konverze z typu Integer na Boolean.

Jazyk je založen na Common Language Infrastructure a také používá unifikovaný typový systém zvaný Common Type System, který je nutný kvůli kompatibilitě .NET Frameworku s různými programovacími jazyky. Všechny jeho typy, včetně primitivních datových typů jako je Integer, jsou potomky třídy System.Object a dědí od něj i všechny jeho metody jako je například ToString. Datové typy rozděluje na hodnotové, které jsou z výkonnostních důvodů uloženy přímo na zásobníku a referenční, které jsou uloženy na haldě a neuchovávají hodnotu samotnou, ale pouze odkaz na místo v paměti, kde je požadovaná instance uložena. [6]

2.8.1 LINQ

LINQ je integrovaný dotazovací jazyk, který byl představen spolu s jazykem C Sharp 3.0. Přináší jednoduchý způsob pro dotazování nad jakýmkoliv daty a usnadňuje tak jejich tvorbu, třídění, propojování i vyhledávání v nich. Dotazy lze psát buď ve formátu, který je velmi podobný SQL dotazům, nebo v podobě Lambda výrazů. Klíčové slova dotazů, které odpovídají názvům metod u Lambda výrazů, také vychází z SQL dotazů, a proto jimi jsou select, from, where, join, orderby a mnohé další.

```
var kratkaSlova = from c in slova
                  where c.Length < 5
                  orderby c.Length
                  select c;
```

Obrázek 12: LINQ dotaz ve stylu SQL

```
var kratkaSlova = slova.Where(c => c.Length < 5).OrderBy(c => c.Length);
```

Obrázek 13: LINQ dotaz pomocí Lambda výrazů

LINQ přináší několik základních implementací, které by měly pokrýt běžné potřeby programátorů pro tvorbu aplikací pracujících s daty:

- ❖ LINQ to Objects je implementace pro standardní kolekce nacházející se v paměti.
- ❖ LINQ to SQL je implementace pro Microsoft SQL Server 2000 a vyšší.
- ❖ LINQ to XML je implementace pro práci s XML daty.
- ❖ LINQ to DataSet je implementace pro práci s DataSety v ADO .NET. [7]

2.9 Microsoft Visual Studio

Microsoft Visual Studio je integrované vývojové prostředí od společnosti Microsoft. Může být použito k vývoji konzolových aplikací, Windows Forms aplikací, webových stránek, webových aplikací a webových služeb. Aplikace je možno psát jak ve strojovém kódu, tak ve spravovaném kódu. Visual Studio podporuje platformy Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework a Microsoft Silverlight. Mezi podporované jazyky patří C a C++, Visual Basic .NET a C Sharp. Podpora dalších jazyků jako Chrome, F Sharp, Python a Ruby spolu s ostatními může být přidána jazykovými službami, které se dají nainstalovat zvlášť. Existují i verze

Visual Studio jen pro určitý programovací jazyk, které uživatelé poskytují omezenější jazykové služby. [8]

2.9.1 Editor kódu

Součástí Visual Studio je i editor kódu, který podporuje IntelliSense a refaktorování. IntelliSense je nástroj pro automatické dokončování nejen pro proměnné, funkce a metody, ale také pro cykly nebo dotazy. Slouží jako dokumentace a zároveň jako rozcestník pro názvy proměnných, funkcí a metod. Při psaní kódu se návrhy automatického dokončování zobrazují ve vyskakovacím seznamu přímo u psaného kódu. IntelliSense je dostupný pro všechny podporované jazyky v .NET Framework.

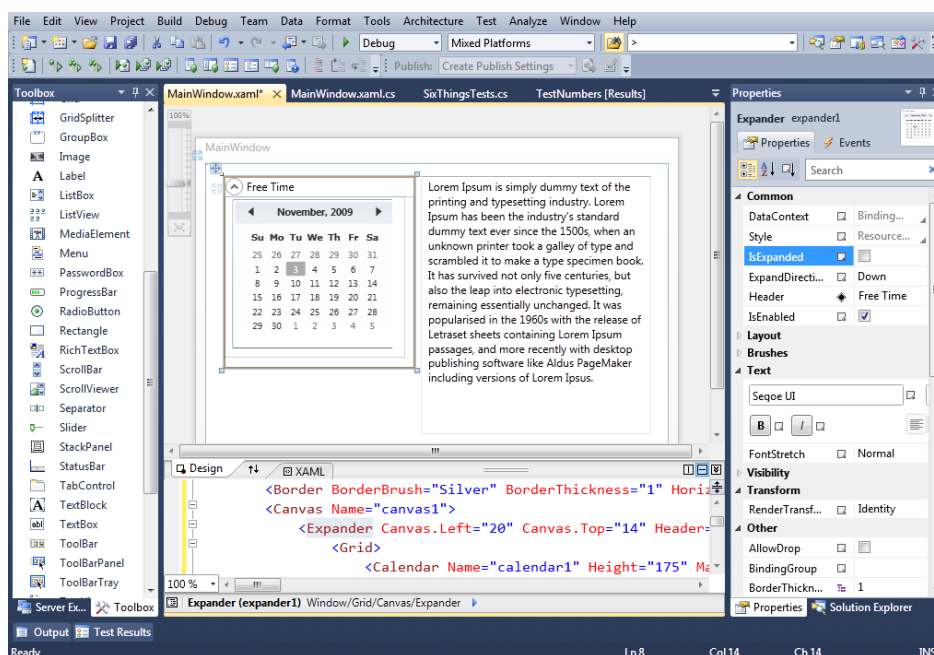
Editor kódu podporuje sbalování bloků kódu, nastavování záložek pro rychlou navigaci v kódu a přírůstkové i normální hledání s podporou regulárních výrazů. Obsahuje také vícepoložkovou schránku a seznam úkolů. Podporuje snippety, což jsou uložené šablony opakujícího se kódu, které mohou být do kódu vloženy a přizpůsobeny aktuálnímu projektu. Všechny tyto nástroje mají podobu plovoucích oken, která mohou být ukotvena na ploše nebo schována při nepoužívání. [8]

2.9.2 Designer

Visual Studio zahrnuje také množství grafických designérů, které mají za úkol usnadnit vývoj aplikací. Jedním z nich je WinForms Designer, který je určený k návrhu grafického uživatelského rozhraní aplikací. Obsahuje paletu všech očekávaných ovládacích prvků, jako jsou tlačítka, textové pole, popisky, tabulky a jiné další prvky. GUI, které vytváří, je řízeno událostmi. Dalším designérem určeným k vytváření GUI aplikací je WPF Designer. Ten obsahuje podobnou sadu ovládacích prvků. Vytváří XAML kód, který je kompatibilní s Microsoft Expression Designem a se zdrojovým kódem je spojen code-behind modelem. Kromě designérů pro Windows aplikace obsahuje Visual Studio také designér webových stránek. Ten se používá pro vývoj ASP.NET aplikací a podporuje HTML, CSS a JavaScript.

Designer tříd se používá pro vytváření a úpravu tříd aplikace s použitím UML modelu a může vygenerovat diagramy z ručně psaných tříd. Součástí Visual Studio jsou také designéry dat, které se používají například k úpravě schémat databáze, a také

designéry mapování mezi databázovými schématy a třídami, které zabalují jejich data. To je například designér pro LINQ to SQL. [8]



Obrázek 14: WPF Designer [24]

2.9.3 Debugger

Debugger integrovaný ve Visual Studiu pracuje jak na úrovni spravovaného kódu, tak na úrovni strojového kódu. Může být použit pro debuggování aplikací napsaných ve kterémkoliv jazyce podporovaném .NET Frameworkem. Debugger podporuje breakpointy, které umožňují zastavit běh programu na určité pozici a také watch, který sleduje hodnoty proměnných za běhu procesu. Breakpointy mohou být podmíněné tak, že se spustí až při splnění dané podmínky. Také dovoluje upravovat zdrojový kód přímo za běhu aplikace a poté pokračovat v jejím debuggování. Dokáže vstupovat do funkcí a debuggovat je uvnitř nebo je přeskakovat. Debugger rovněž zobrazuje obsahy proměnných, když přes ně ve zdrojovém kódu přejede kurzor. [8]

2.10 Windows Communication Foundation

Windows Communication Foundation je softwarový balík pro vytváření aplikací, které jsou orientované na komunikaci s webovými službami. WCF je součástí .NET Framework od verze 3.0. Pomocí WCF je můžete odesílat data jako asynchronní zprávy z jednoho koncového bodu služby do druhého. Koncový bod služby může být součástí trvale dostupné služby, kterou hostí IIS, nebo to může být služba, která je součástí aplikace. Koncový bod může být také klientem služby, který požaduje data ze služby. Zprávy mohou obsahovat jeden znak či data ve formátu XML, nebo mohou být komplexní jako proud binárních dat.

Zprávy si mezi sebou můžou koncové body vyměňovat podle několika různých schémat. Nejčastější schéma je, že jeden koncový bod odešle žádost a druhý mu na ni odešle odpověď. Existují i jiné schémata, jako jsou jednosměrné zprávy, ve kterých jeden koncový bod pošle zprávu a neočekává na ni odpověď nebo složitější duplexní výměna, kde oba koncové body naváží spojení a poté si mezi sebou posílají data tam i zpět podobně jako u instantní komunikace.

WCF podporuje publikování metadat služby ve formátech průmyslových standardů jako je WSDL, XML schéma a WS-Policy. Tato metadata mohou být použita pro automatické generování a nastavování klientů pro přístup k WCF službám. Metadata mohou být publikována přes HTTP, zabezpečeně přes HTTPS nebo pomocí Web Service Metadata Exchange standardu.

Zprávy odesílané WCF službou jsou odolné proti přerušení. Zprávy se ukládají do databáze, a pokud dojde k přerušení komunikace, databáze umožňuje obnovit výměnu zpráv po obnovení připojení. Zprávy mohou být šifrovány kvůli ochraně jejich obsahu a je také možné vynutit, aby se uživatel přihlásil před tím, než bude moci zprávy přijímat. Bezpečnost zajišťují známé standardy jako je SSL nebo WS-SecureConversation. [9]

2.11 SOAP

SOAP je síťový protokol pro výměnu zpráv založených na formátu XML. Tvoří základní vrstvu komunikace mezi webovými službami a poskytuje prostředí pro tvorbu složitější komunikace. Existuje několik různých druhů šablon pro komunikaci na tomto protokolu. Nejznámější z nich je RPC šablona, kde jeden z účastníků komunikace je klient a druhý server, který ihned odpovídá na požadavky klienta. SOAP zpráva se skládá z 3 hlavních částí, kterými jsou obálka, hlavička a tělo zprávy.

Aplikační vrstvu protokolu SOAP můžou tvořit protokoly HTTP i SMTP. Protokol HTTP ale dosáhl daleko většího použití. Je to především proto, že HTTP je praktický základ pro dnešní Internetovou infrastrukturu. Díky tomu může SOAP jednoduše procházet přes firewally, což je hlavní výhoda oproti ostatním protokolům.

Formát XML byl zvolen jako standard pro přenos SOAP zpráv pro jeho rozšířenost a dostupnost bezplatných vývojových nástrojů. Zdlouhavá syntaxe XML má však své výhody i nevýhody. Je jednoduše čitelná pro člověka a jednoduše se takové zprávy vytváří, ale počítač ji musí složitě parsovat, což stojí hodně procesorového času a operační paměti. Má také velké nároky na množství přenesených dat a zvláště u zpráv s velkým množstvím přenášených dat můžou z větší části nepotřebné tagy, obalující přenášená data, zabírat značnou část objemu zprávy. Proto je pak nutné takové zprávy před přenosem po síti ještě zkomprimovat. [10]

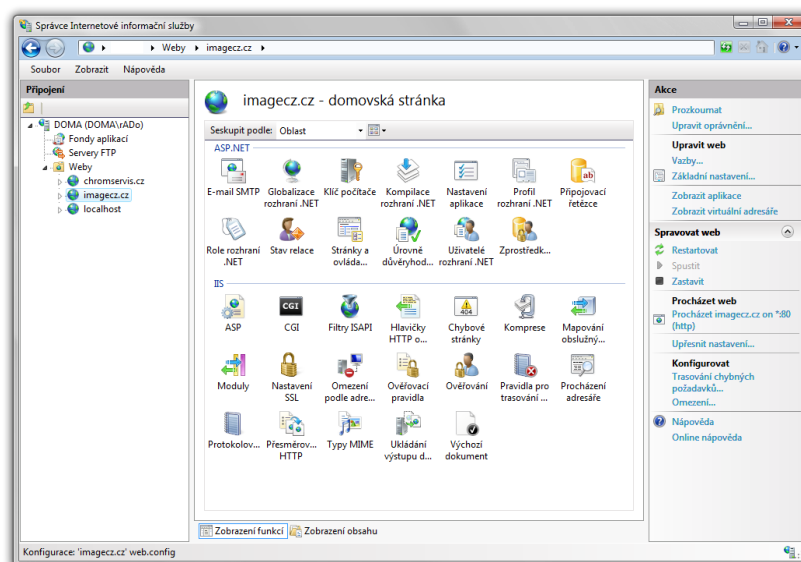
2.12 Internetová Informační Služba

Internetová Informační Služba je webový server a sada dalších rozšiřujících modulů pro hostování webových služeb a webových aplikací. IIS byla vyvinuta firmou Microsoft a lze ji provozovat na operačních systémech Windows. Je postavena na modulárním modelu, což umožňuje, aby na serveru běžely pouze ty moduly, které jsou třeba. IIS obsahuje například moduly:

- ❖ HTTP modul – Používá se k obsluze úkolů spojených s HTTP požadavky.
- ❖ Bezpečnostní modul – Stará se o zabezpečení serveru, například specifikuje autentizační schémata, provádí URL autorizaci a filtruje požadavky.

- ❖ Content modul – Zpracovává požadavky na statické soubory, vrací výchozí stránku, pokud je třeba a vypisuje obsah adresářů.
- ❖ Kompresní modul – Provádí kompresi odpovědí, aplikuje Gzip kompresi převodu kódování odpovědí a kompresi statického obsahu.
- ❖ Caching modul – Ukládá zpracovávané informace v paměti serveru a z vyrovnávací paměti poskytuje data následných žádostí o stejný zdroj.
- ❖ Logovací a diagnostický modul – Provádí diagnostiku serveru a zaznamenává různé události probíhající na serveru a ukládá je do logů.

Internetová Informační Služba podporuje komunikační protokoly jako FTP, FTPS, SMTP, NNTP, HTTP a HTTPS. Je možné ji spravovat i pomocí příkazové řádky PowerShell. Nejnovější verzí IIS je v současnosti verze 7.5 a IIS je druhým nejpoužívanějším webovým serverem na světě. Dostupná je také odlehčená verze IIS Express, kterou je možné nainstalovat zdarma na operační systémy Windows XP s aktualizací Service Pack 3 a novější. IIS Express podporuje pouze protokoly HTTP a HTTPS. [11]



Obrázek 15: Internetová Informační Služba [25]

2.13 SQL

SQL je standardizovaný dotazovací jazyk používaný pro práci s daty v relačních databázích. Poskytuje příkazy pro manipulaci s daty, definici dat, řízení přístupových práv, řízení transakcí a další speciální příkazy.

SQL byl vyvinut na počátku 70. let 20. století firmou IBM pod názvem SEQUEL. V roce 1981 pak společnost IBM uvedla novou verzi jazyka a jazyk byl přejmenován na SQL. V dalších letech pak byly vydány standardy SQL-86, SQL-92 a SQL-99, které byly pojmenovány vždy podle roku jejich přijetí. Jazyk byl také původně založený na relační algebře.

SQL patří mezi deklarativní programovací jazyky, což znamená, že kód nepíšeme v žádném samostatném programu, ale vkládáme jej do jiného programovacího jazyka, který je již procedurální. Se samotným jazykem SQL je možné pracovat pouze v případě, že se terminálem připojíme na SQL server a do příkazového řádku bychom zadávali příkazy jazyka SQL.

Jazyk SQL je rozdělen do několika jazykových prvků:

- ❖ Doložky, které nejsou v některých případech povinné a tvoří základ dotazů.
- ❖ Výrazy, které mohou produkovat buď skalární hodnoty, nebo tabulky skládající se ze sloupců a řádků dat.
- ❖ Predikáty, které vytvářejí podmínky a používají se k omezení dotazů.
- ❖ Dotazy, načítají data na základě specifických kritérií.
- ❖ Prohlášení, které ovlivňují schémata, transakce, tok programu, spojení nebo diagnostiku. Měly by obsahovat ";" jako ukončovač, ale není to vyžadováno na všech platformách.

SQL se skládá z několika částí, které jsou určeny pro administrátory a návrháře databázových systémů nebo pro koncové uživatele a programátory. První částí jazyka SQL je jazyk Data Definition Language. Jedná se o jazyk pro vytváření databázových schémat a katalogů. Způsob ukládání tabulek definuje jazyk Storage Definition Language. Třetí částí pro návrháře a správce je View Definition Language, určující vytváření pohledů. Poslední částí je Data Manipulation Language, který obsahuje základní příkazy jako je INSERT,

UPDATE, DELETE a také nejpoužívanější příkaz SELECT. S jazykem DML pak nejvíce pracují koncoví uživatelé a programátoři databázových aplikací. [12]

2.13.1 Transact-SQL

Transact-SQL je proprietární rozšíření pro jazyk SQL od společností Microsoft a Sybase. Všechny aplikace komunikují s instancí Microsoft SQL Serveru právě pomocí jazyka T-SQL bez ohledu na uživatelské rozhraní aplikace. T-SQL rozšiřuje jazyk SQL některými doplňkovými funkcemi:

- ❖ Přidává lokální proměnné.
- ❖ Přidává různé podpůrné funkce pro zpracování řetězců, datumů, matematických funkcí a mnohých dalších.
- ❖ Rozšiřuje standard SQL jazyka o možnosti procedurálního programování.
- ❖ Pozměňuje příkazy DELETE a UPDATE. [12]

2.14 Microsoft SQL Server

Microsoft SQL Server je relační databázový systém vytvořený firmou Microsoft. Hlavní funkcí Microsoft SQL Serveru je skladování a následné poskytování uskladněných dat ostatním aplikacím buď na stejném serveru, nebo přes síť. Jeho hlavními dotazovacími jazyky jsou SQL a T-SQL. Server je dostupný v několika odlišných edicích, které mají různou skladbu funkcí podle zaměření na různé uživatele.

Microsoft SQL Server zahrnuje následující nástroje:

- ❖ Relační databázový stroj s vysokým výkonem a podporou strukturovaných a nestrukturovaných dat jako například XML.
- ❖ Služba Replication Services, která zajišťuje replikaci dat pro aplikace zpracovávající distribuovaná či mobilní data.
- ❖ Služba Notification Services, která zajišťuje zasílání upozornění pro vývoj a nasazení škálovatelných aplikací, které mohou na připojená i mobilní zařízení zasílat individuální požadavky.
- ❖ Služba Integration Services, která zajišťuje funkce extrakce, transformace a načítání dat pro datové sklady a integraci dat.

- ❖ Služba Analysis Services, která zajišťuje OLAP pro rychlou a pokročilou analýzu velkých a složitých datových sad s využitím vícedimenzionálních úložišť.
- ❖ Služba Reporting Services, která zajišťuje vytváření, správu a zasílání tradičních papírových i interaktivních webových sestav.
- ❖ Integrované nástroje pro pokročilou správu a ladění databází a umožňující úzkou integraci s nástroji, jako jsou například Microsoft Operations Manager a Microsoft Systems Management Server.
- ❖ Nástroje pro vývojáře, které nabízí integrované nástroje pro databázový stroj, extrakci, transformaci a načítání dat, dolování dat, funkce OLAP a vytváření sestav, které jsou úzce integrovány s integrovaným vývojovým prostředím Microsoft Visual Studio a poskytují komplexní funkce pro vývoj aplikací. Jednotlivé součásti Microsoft SQL Serveru jsou pak dodávány s vlastními objektovými modely a sadou aplikačních programovacích rozhraní. [12]

2.14.1 Relační databáze

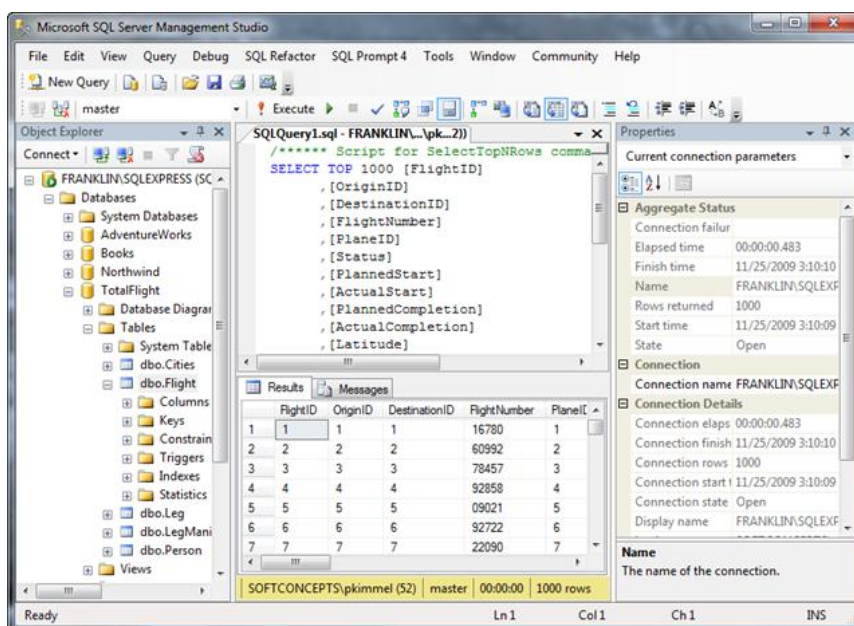
Relační databáze je databáze založená na relačním modelu a tabulkách. Databázi si lze představit jako soubor dat, který slouží pro popis reálného světa jako je například evidence studentů. Entita je prvek reálného světa, například student, který je popsán svými vlastnostmi. Ty se většinou považují za atribut jako například jméno a příjmení. Mezi entitami můžou existovat 3 typy vazeb, kterými jsou 1:1, 1:N a M:N.

Relaci si lze představit jako tabulku, která se skládá ze sloupců a řádků. Sloupce odpovídají jednotlivým vlastnostem (atributům) entity. Údaje v jednom řádku tabulky pak zobrazují aktuální stav světa a představují prvky relace. [12]

2.14.2 Microsoft SQL Server Management Studio

Microsoft SQL Server Management Studio je integrované prostředí pro přístup, nastavení, správu, administraci a vývoj na webovém serveru Microsoft SQL Server. Bylo vydáno společně s Microsoft SQL Server 2005 a kombinuje do jednoho společného prostředí funkce Enterprise Manager, Query Analyzer a Analysis Manager, které byly součástí předchozích verzí Microsoft SQL Serveru. Kromě nich funguje i u všech ostatních

součástí SQL Serveru jako jsou například Reporting Services nebo Integration Services. [12]



Obrázek 16: Microsoft SQL Server Management Studio [26]

2.15 Digitální certifikát

Digitální certifikát je soubor dat ve stanoveném formátu, který identifikuje osobu nebo server a může během elektronické komunikace mezi dvěma subjekty zajistit šifrování přenášených dat, ověření jedné nebo druhé strany, rozpoznání neoprávněné modifikace dat a s tím související digitální podpis. Pro certifikáty se používá mezinárodní norma X.509, která jednoznačně popisuje strukturu certifikátu.

Celý systém digitálních certifikátů je postaven na matematické podstatě, ze které vycházejí algoritmy používané k šifrování. K tomu je nutné, aby v počítači uživatele nebo na serveru existoval takzvaný soukromý nebo též privátní klíč. Tento soukromý klíč vygeneruje uživatel ve svém počítači. Jelikož se z hlediska bezpečnosti jedná o velice kritickou součást, má uživatel za povinnost chránit soukromý klíč na bezpečném úložišti, zabránit jeho ztrátě a zároveň jeho zneužití jinými subjekty. Pokud to neomezují organizační nebo technické požadavky, měl by být chráněn heslem, které je známo pouze uživateli. Následně uživatel vygeneruje žádost o vystavení digitálního certifikátu, do níž uvede své identifikační údaje, které si přeje do certifikátu uvést (jméno osoby, název firmy, název serveru, email atd.). Do žádosti se navíc automaticky přidá matematický protějšek

soukromého klíče, takzvaný veřejný klíč. Vystavitel certifikátu, kterým je certifikační autorita, provede ověření žadatele dle certifikační politiky pro daný typ certifikátu, případně upraví obsah certifikátu, a následně vystaví certifikát, což je soubor obsahující informace vztažené k danému soukromému klíči. Certifikační autorita digitálně podepíše informace uvedené v certifikátu, čímž se zaručuje za jejich správnost. Z matematické podstaty digitálních certifikátů je pak samozřejmě možné ověřit, že certifikát podepsala právě uvedená certifikační autorita. [13]

2.16 Secure Hash Algorithm

Secure Hash Algorithm patří do rodiny hashovacích funkcí, které vytváří ze vstupních dat výstup fixní délky (otisk). SHA vytváří výstupní otisk o délce minimálně 160bitů, ale doporučená délka je minimálně 256bitů. Jeho hlavní vlastností je, že malá změna na vstupu vede k velké změně na výstupu a k vytvoření zásadně odlišného otisku. Nevýhodou SHA je, že z otisku nelze zpět získat původní data. Používá se pro kontrolu integrity souborů, ukládání hesel a u různých protokolů jako jsou SSL nebo IPsec.

2.17 Customer Relationship Management

CRM je zkratka z anglického Customer Relationship Management a označují se tak systémy pro řízení vztahů se zákazníky. Jsou to programy, které umožňují shromažďovat, třídit a zpracovávat údaje o zákaznících, především jejich kontakty, probíhající obchodní procesy a dosahované tržby. CRM systémy tak pomáhají sledovat a vyhodnocovat veškeré obchodní aktivity v rámci celé společnosti. I z toho důvodu bývají součástí CRM systémů nejrůznější statistiky.

Cílem CRM je především zlepšit cílení služeb, lépe porozumět zákazníkům a identifikovat jejich konkrétní potřeby. To umožňuje budovat dlouhodobě prospěšné vztahy se zákazníky a tím vytěžit z jednoho zákazníka větší zisk. Protože stávající zákazníci jsou pro firmu nejhodnotnější, vyplatí se pomocí CRM systémů zajistit si jejich věrnost a důkladně o ně pečovat.

Z výše uvedeného je zřejmé, že CRM systém není pouhým software, jde o dlouhodobou strategii na poli komunikace, marketingu, obchodu a servisu, navíc průběžně

přizpůsobovanou aktuálním podmínkám a požadavkům zákazníků. CRM tak zasahuje do všech vnějších úrovní fungování společnosti.

CRM systémy se ve firmách objevují od počátku devadesátých let, kdy k jejich rozvoji přispěly nové možnosti v oblasti e-business, růst nákladů na získávání nových zákazníků a snižování průměrného zisků u většiny běžných produktů. [14]

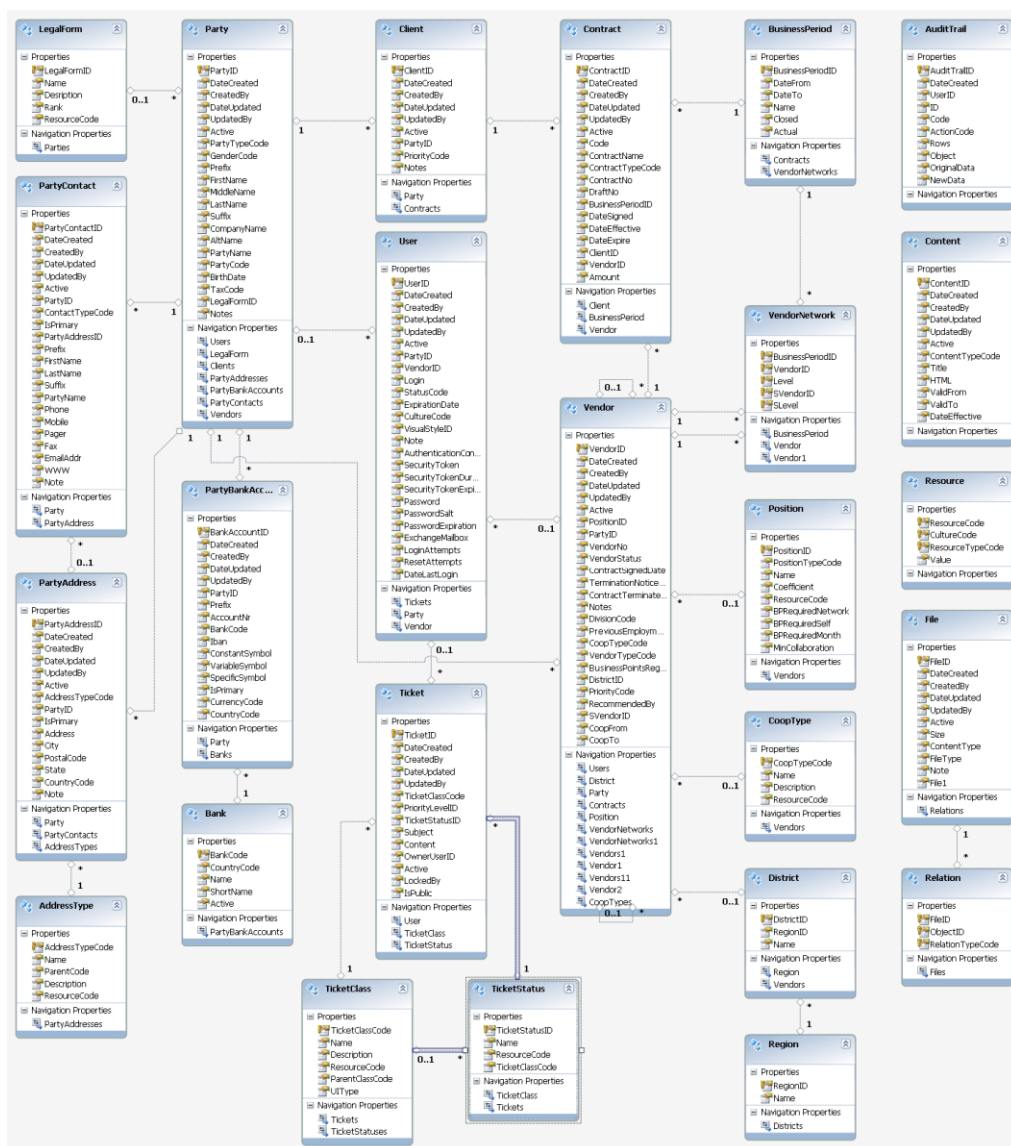
II. PRAKTICKÁ ČÁST

3 RELAČNÍ DATABÁZE

Zdrojem všech dat pro aplikaci je relační databáze. Ta běží na Microsoft SQL Serveru. Řešení od firmy Microsoft jsem zvolil proto, aby se na databázi dalo dobře připojit ze systémů postavených na .NET Frameworku. Databáze je navržena, jako by byla součástí větší databáze, která neslouží pouze pro mobilního klienta, ale je součástí většího systému, který obsahuje webovou aplikaci nebo desktopového klienta.

3.1 Tabulky


Tabulky určují strukturu dat, které jsou uloženy v systému, a tyto data v sobě uchovávají. Databáze obsahuje celkem 25 tabulek, a část z nich je dále podrobně popsána.



Obrázek 17: Struktura relační databáze

3.1.1 AuditTrail

Tabulka uchovává záznamy o všech voláních uložených procedur. U procedur, které mění data v databázi, zaznamenává stav dat před a po úpravě. U dotazů na data zaznamenává parametry, podle kterých byly data vybírány. To umožňuje správci systému kontrolovat činnost uživatelů v systému, a také vrátit data do původního stavu, pokud byly poškozeny.

	Column Name	Data Type	Allow Nulls
	AuditTrailID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	UserID	int	<input type="checkbox"/>
	ID	int	<input checked="" type="checkbox"/>
	Code	varchar(50)	<input checked="" type="checkbox"/>
	ActionCode	varchar(2)	<input type="checkbox"/>
	Rows	int	<input type="checkbox"/>
	Object	varchar(100)	<input type="checkbox"/>
	OriginalData	xml	<input checked="" type="checkbox"/>
	NewData	xml	<input checked="" type="checkbox"/>

Obrázek 18: Struktura tabulky AuditTrail

3.1.2 Resources


Tabulka uchovává překlady textů a číselníků. Mohla by také uchovávat překlad celé aplikace, například webové, která se by mohla automaticky lokalizovat překlady v databázi. Překlady jsou v tabulce identifikované svým kódem a také kódem jazyka.

	Column Name	Data Type	Allow Nulls
	ResourceCode	varchar(150)	<input type="checkbox"/>
	CultureCode	varchar(10)	<input type="checkbox"/>
	ResourceTypeCode	varchar(10)	<input type="checkbox"/>
	Value	nvarchar(1024)	<input checked="" type="checkbox"/>

Obrázek 19: Struktura tabulky Resources

3.1.3 Users


Tabulka uchovává uživatelské účty a základní informace o nich jako je uživatelské jméno, heslo, stav účtu nebo preferovaný jazyk. Každý záznam tabulky může být navázaný na záznam tabulek Parties nebo Vendors. Hesla jsou v tabulce uloženy ve formě hashe, proto není možné zjistit hesla uživatelů ani pro správce systému. Použitý hash je SHA256, který je natolik bezpečný, že je v současné době prakticky nemožné hesla rozluštit i v případě, že by byla data odcizena.

	Column Name	Data Type	Allow Nulls
	UserID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	PartyID	int	<input checked="" type="checkbox"/>
	VendorID	int	<input checked="" type="checkbox"/>
	Login	varchar(50)	<input type="checkbox"/>
	StatusCode	varchar(2)	<input checked="" type="checkbox"/>
	ExpirationDate	date	<input checked="" type="checkbox"/>
	CultureCode	varchar(10)	<input checked="" type="checkbox"/>
	VisualStyleID	int	<input checked="" type="checkbox"/>
	Note	nvarchar(250)	<input checked="" type="checkbox"/>
	AuthenticationContact	varchar(50)	<input checked="" type="checkbox"/>
	SecurityToken	varchar(10)	<input checked="" type="checkbox"/>
	SecurityTokenDuration	int	<input checked="" type="checkbox"/>
	SecurityTokenExpiration	datetime	<input checked="" type="checkbox"/>
	Password	nvarchar(512)	<input checked="" type="checkbox"/>
	PasswordSalt	nvarchar(512)	<input checked="" type="checkbox"/>
	PasswordExpiration	datetime	<input checked="" type="checkbox"/>
	ExchangeMailbox	tinyint	<input checked="" type="checkbox"/>
	LoginAttempts	int	<input type="checkbox"/>
	ResetAttempts	int	<input type="checkbox"/>
	DateLastLogin	datetime	<input checked="" type="checkbox"/>

Obrázek 20: Struktura tabulky Users

3.1.4 Content


Tabulka uchovává různý textový obsah, kterým můžou být texty novinek, oznámení nebo obsah nápovědy. V případě této aplikace obsahuje tabulka novinky ve formátu HTML.

	Column Name	Data Type	Allow Nulls
	ContentID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	ContentTypeCode	varchar(10)	<input checked="" type="checkbox"/>
	Title	nvarchar(250)	<input type="checkbox"/>
	HTML	nvarchar(MAX)	<input checked="" type="checkbox"/>
	ValidFrom	datetime	<input checked="" type="checkbox"/>
	ValidTo	datetime	<input checked="" type="checkbox"/>
	DateEffective	datetime	<input checked="" type="checkbox"/>

Obrázek 21: Struktura tabulky Content

3.1.5 Parties


Záznamy této tabulky představují rozšiřující informace o prodejcích, klientech, firmách nebo uživatelích. U klienta jsou to například informace jako celé jméno, rodné číslo nebo pohlaví. Přes záznamy této tabulky jsou dále navázané další rozšiřující informace jako kontaktní údaje, adresa nebo bankovní účet. Tabulka obsahuje vazbu na tabulku LegalForms.

	Column Name	Data Type	Allow Nulls
	PartyID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	PartyTypeCode	varchar(2)	<input type="checkbox"/>
	GenderCode	varchar(2)	<input checked="" type="checkbox"/>
	Prefix	nvarchar(20)	<input checked="" type="checkbox"/>
	FirstName	nvarchar(50)	<input checked="" type="checkbox"/>
	MiddleName	nvarchar(50)	<input checked="" type="checkbox"/>
	LastName	nvarchar(150)	<input checked="" type="checkbox"/>
	Suffix	nvarchar(20)	<input checked="" type="checkbox"/>
	CompanyName	nvarchar(200)	<input checked="" type="checkbox"/>
	AltName	nvarchar(100)	<input checked="" type="checkbox"/>
	PartyName	nvarchar(202)	<input checked="" type="checkbox"/>
	PartyCode	varchar(50)	<input checked="" type="checkbox"/>
	BirthDate	date	<input checked="" type="checkbox"/>
	TaxCode	varchar(50)	<input checked="" type="checkbox"/>
	LegalFormID	int	<input checked="" type="checkbox"/>
	Notes	varchar(250)	<input checked="" type="checkbox"/>

Obrázek 22: Struktura tabulky Parties

3.1.6 PartyAddresses


Tabula uchovává adresy pro záznamy z tabulky Parties a obsahuje vazby na tabulky Parties a AddressTypes.

	Column Name	Data Type	Allow Nulls
	PartyAddressID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	AddressTypeCode	varchar(2)	<input type="checkbox"/>
	PartyID	int	<input type="checkbox"/>
	IsPrimary	tinyint	<input type="checkbox"/>
	Address	nvarchar(150)	<input type="checkbox"/>
	City	nvarchar(100)	<input checked="" type="checkbox"/>
	PostalCode	varchar(20)	<input checked="" type="checkbox"/>
	State	nvarchar(150)	<input checked="" type="checkbox"/>
	CountryCode	varchar(2)	<input type="checkbox"/>
	Note	nvarchar(250)	<input checked="" type="checkbox"/>

Obrázek 23: Struktura tabulky PartyAddresses

3.1.7 PartyBankAccounts

Tabula uchovává bankovní účty pro záznamy z tabulky Parties a obsahuje vazby na tabulky Parties a Banks.

	Column Name	Data Type	Allow Nulls
	BankAccountID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	PartyID	int	<input type="checkbox"/>
	Prefix	varchar(10)	<input checked="" type="checkbox"/>
	AccountNr	varchar(10)	<input type="checkbox"/>
	BankCode	varchar(10)	<input type="checkbox"/>
	Iban	varchar(31)	<input checked="" type="checkbox"/>
	ConstantSymbol	varchar(10)	<input checked="" type="checkbox"/>
	VariableSymbol	varchar(10)	<input checked="" type="checkbox"/>
	SpecificSymbol	varchar(10)	<input checked="" type="checkbox"/>
	IsPrimary	tinyint	<input type="checkbox"/>
	CurrencyCode	char(3)	<input type="checkbox"/>
	CountryCode	varchar(2)	<input type="checkbox"/>

Obrázek 24: Struktura tabulky PartyBankAccounts

3.1.8 PartyContacts


Tabulka uchovává kontaktní údaje pro záznamy z tabulky Parties a obsahuje vazby na tabulky Parties a PartyAddresses.

	Column Name	Data Type	Allow Nulls
	PartyContactID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	PartyID	int	<input type="checkbox"/>
	ContactTypeCode	varchar(5)	<input type="checkbox"/>
	IsPrimary	tinyint	<input type="checkbox"/>
	PartyAddressID	int	<input checked="" type="checkbox"/>
	Prefix	nvarchar(20)	<input checked="" type="checkbox"/>
	FirstName	nvarchar(50)	<input checked="" type="checkbox"/>
	LastName	nvarchar(150)	<input checked="" type="checkbox"/>
	Suffix	nvarchar(20)	<input checked="" type="checkbox"/>
	PartyName	nvarchar(202)	<input type="checkbox"/>
	Phone	varchar(50)	<input checked="" type="checkbox"/>
	Mobile	varchar(50)	<input checked="" type="checkbox"/>
	Pager	varchar(50)	<input checked="" type="checkbox"/>
	Fax	varchar(50)	<input checked="" type="checkbox"/>
	EmailAddr	varchar(150)	<input checked="" type="checkbox"/>
	WWW	varchar(250)	<input checked="" type="checkbox"/>
	Note	nvarchar(250)	<input checked="" type="checkbox"/>

Obrázek 25: Struktura tabulky PartyContacts

3.1.9 Clients


Tabula uchovává seznam všech klientů v systému. Neobsahuje žádné podrobnosti o klientech, ale pouze vazbu na tabulku Parties, která uchovává další podrobnosti o klientech.

	Column Name	Data Type	Allow Nulls
	ClientID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	int	<input type="checkbox"/>
	PartyID	int	<input type="checkbox"/>
	PriorityCode	varchar(5)	<input checked="" type="checkbox"/>
	Notes	nvarchar(500)	<input checked="" type="checkbox"/>

Obrázek 26: Struktura tabulky Clients

3.1.10 Contracts


Tabula uchovává všechny smlouvy, které prodejci uzavřeli přes tento systém. Obsahuje podrobnosti o smlouvě jako její částka nebo datum uzavření, ze kterých se pak dále vytvářejí statistiky o produkci prodejců, divizí nebo všech prodejců v systému. Záznamy také obsahují vazby na klienta a prodejce, kteří smlouvu uzavřeli, a zúčtovací období, do kterého smlouva patří. Ty odpovídají vazbám na tabulky Clients, Vendors a BusinessPeriods.

	Column Name	Data Type	Allow Nulls
	ContractID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	Code	varchar(20)	<input type="checkbox"/>
	ContractName	nvarchar(100)	<input type="checkbox"/>
	ContractTypeCode	varchar(2)	<input type="checkbox"/>
	ContractNo	varchar(50)	<input checked="" type="checkbox"/>
	DraftNo	varchar(50)	<input checked="" type="checkbox"/>
	BusinessPeriodID	int	<input type="checkbox"/>
	DateSigned	date	<input type="checkbox"/>
	DateEffective	date	<input type="checkbox"/>
	DateExpire	date	<input checked="" type="checkbox"/>
	ClientID	int	<input type="checkbox"/>
	VendorID	int	<input type="checkbox"/>
	Amount	money	<input type="checkbox"/>

Obrázek 27: Struktura tabulky Contracts

3.1.11 Files

Tabula slouží jako úložiště pro všechny soubory, které je třeba v systému uchovat. Mohou to být textové dokumenty, obrázky nebo binární soubory. Data souborů se uchovávají v binární podobě, a aby bylo možné soubor správně obnovit z databáze, uchovává se i původní typ souboru a jeho velikost.

	Column Name	Data Type	Allow Nulls
	FileID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	Size	bigint	<input type="checkbox"/>
	ContentType	varchar(255)	<input type="checkbox"/>
	FileType	varchar(10)	<input type="checkbox"/>
	Note	nvarchar(50)	<input checked="" type="checkbox"/>
	[File]	varbinary(MAX)	<input type="checkbox"/>

Obrázek 28: Struktura tabulky Files

3.1.12 Relations


Tabula uchovává vazby mezi soubory, které jsou uloženy v tabulce Files a záznamy z jiných tabulek v databázi. Jsou to například vazby mezi naskenovaným obrázkem se smlouvou nebo jinými dokumenty ke smlouvě a příslušnou smlouvou nebo vazba mezi prodejcem a jeho fotkou, která se má zobrazovat v systému.

	Column Name	Data Type	Allow Nulls
	FileID	int	<input type="checkbox"/>
	ObjectID	int	<input type="checkbox"/>
	RelationTypeCode	varchar(5)	<input type="checkbox"/>

Obrázek 29: Struktura tabulky Relations

3.1.13 Vendors


Tabula uchovává seznam prodejců. Obsahuje všechny potřebné informace o prodejci jako je jeho pozice, osobní číslo nebo k jaké patří divizi. Další informace o prodejci jsou navázány přes vazbu na tabulku Parties. Tabulka dále obsahuje vazby na číselníky pozic prodejců a typů spolupráce v tabulkách Positions a CoopTypes. Informace obsahují i vazbu na okres v tabulce Districts, ke kterému prodejce patří.

	Column Name	Data Type	Allow Nulls
	VendorID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	Active	tinyint	<input type="checkbox"/>
	PositionID	smallint	<input checked="" type="checkbox"/>
	PartyID	int	<input type="checkbox"/>
	VendorNo	varchar(20)	<input checked="" type="checkbox"/>
	VendorStatus	char(1)	<input checked="" type="checkbox"/>
	ContractSignedDate	date	<input checked="" type="checkbox"/>
	TerminationNoticeDate	date	<input checked="" type="checkbox"/>
	ContractTerminatedDate	date	<input checked="" type="checkbox"/>
	Notes	nvarchar(MAX)	<input checked="" type="checkbox"/>
	DivisionCode	varchar(5)	<input checked="" type="checkbox"/>
	PreviousEmployment	nvarchar(500)	<input checked="" type="checkbox"/>
	CoopTypeCode	char(2)	<input checked="" type="checkbox"/>
	VendorTypeCode	char(1)	<input checked="" type="checkbox"/>
	BusinessPointsRegionID	int	<input type="checkbox"/>
	DistrictID	int	<input checked="" type="checkbox"/>
	PriorityCode	char(2)	<input checked="" type="checkbox"/>
	RecommendedBy	int	<input checked="" type="checkbox"/>
	SVendorID	int	<input checked="" type="checkbox"/>
	CoopFrom	datetime	<input checked="" type="checkbox"/>
	CoopTo	datetime	<input checked="" type="checkbox"/>

Obrázek 30: Struktura tabulky Vendors

3.1.14 Tickets

Tabula uchovává systémové tikety, které můžou patřit různým činnostem, jako jsou například reklamace, dotazy od klientů nebo zpracovávání smlouvy. Každý tiket je navázaný na svého vlastníka v tabulce Users. Jak tikety prochází systémem a jsou zpracovávány, tak se postupně mění i jejich stav. Ten je navázaný na svůj číselník v tabulce TicketStatuses. Tikety jsou také rozdělené na různé třídy, které jsou navázané na číselník v tabulce TicketClasses.

	Column Name	Data Type	Allow Nulls
	TicketID	int	<input type="checkbox"/>
	DateCreated	datetime	<input type="checkbox"/>
	CreatedBy	int	<input type="checkbox"/>
	DateUpdated	datetime	<input type="checkbox"/>
	UpdatedBy	int	<input type="checkbox"/>
	TicketClassCode	varchar(10)	<input type="checkbox"/>
	PriorityLevelID	int	<input type="checkbox"/>
	TicketStatusID	int	<input type="checkbox"/>
	Subject	nvarchar(200)	<input type="checkbox"/>
	[Content]	nvarchar(MAX)	<input checked="" type="checkbox"/>
	OwnerUserID	int	<input checked="" type="checkbox"/>
	Active	bit	<input type="checkbox"/>
	LockedBy	int	<input checked="" type="checkbox"/>
	IsPublic	tinyint	<input type="checkbox"/>

Obrázek 31: Struktura tabulky Tickets

3.2 Procedury

Aplikace využívá i procedur uložených přímo v databázi. V nich jsou definovány všechny požadavky na data, které může mobilní aplikace z databáze vyžadovat. To je sice pracnější, než napsat dotazy na data přímo ve zdrojových kódech webové služby, ale na druhou stranu je možné se na procedury odkázat i na několika místech současně a také se při jejich opravě nemusí zasahovat do zdrojového kódu webové služby. Zároveň je tak možné využít všech nástrojů Microsoft SQL Server Management Studio a vytvářené procedury si díky němu správně odladit.

Tato databáze obsahuje celkem 18 procedur, které jsou dále podrobně popsány v následující tabulce. Všechny procedury kromě popsané činnosti také zapisují záznam o své činnosti do tabulky AuditTrail.

Název procedury	Vstupní parametry	Popis
ProcAddressTypes	CultureCode varchar(10)	Vrací lokalizovaný číselník s typy adres.
ProcClientInfo	UserID int ClientID int	Vrací podrobné informace o zadaném klientovi včetně jeho adresy, 2 kontaktů a bankovního účtu.
ProcClients	UserID int VendorID int Search nvarchar(200)	Vrací prvních 100 klientů zadaného prodejce. Zároveň také vyhledává podle celého jména klienta. Vyhledávání není citlivé na háčky ani čárky.
ProcClientsBirthday	UserID int VendorID int DateFrom date DateTo date	Vrací prvních 50 klientů, kteří mají v daném časovém rozmezí narozeniny a jsou klienty zadaného prodejce.
ProcContentNews	UserID int	Vrací 20 nejnovějších globálních systémových zpráv.
ProcCoopTypes	CultureCode varchar(10)	Vrací lokalizovaný číselník s typy spolupráce prodejců.
ProcCurrentProduction	UserID int VendorID int DivisionCode char(2)	Vrací informace o aktuální produkci pro zadaného prodejce, která je vypočítaná z jeho uzavřených smluv.
ProcExpiredContracts	UserID int VendorID int DateFrom date DateTo date	Vrací informace o prvních 50 smlouvách, které vyprší v zadaném časovém období, a byly uzavřené zadaným prodejcem.
ProcLegalForms	CultureCode varchar(10)	Vrací lokalizovaný číselník s typy právních forem.
ProcLogin	Login varchar(50)	Vrací základní informace o uživatelském účtu podle zadaného uživatelského jména. Uživatel musí mít přiřazeného prodejce a musí být aktivní.
ProcPositions	CultureCode varchar(10)	Vrací lokalizovaný číselník s pozicemi prodejců.
ProcResources	CultureCode varchar(10)	Vrací vybrané lokalizované texty.
ProcTickets	UserID int UIType nchar(1)	Vrací 20 nejdůležitějších a nejnovějších systémových tiketů, u kterých je zadaný uživatel vlastníkem tiketu.

ProcVendorInfo	UserID int VendorID int	Vrací podrobné informace o zadaném prodejci včetně jeho fotografie, adresy, 2 kontaktů a bankovního účtu.
ProcVendorPhoto	UserID int VendorID int	Vrací obrázek s fotkou daného prodejce.
ProcVendors	UserID int VendorID int Search nvarchar(200)	Vrací prvních 100 prodejců, kteří mají vazbu na zadaného prodejce. Zároveň také vyhledává podle celého jména prodejce. Vyhledávání není citlivé na háčky ani čárky.
ProcVendorsAnniversary	UserID int VendorID int DateFrom date DateTo date	Vrací prvních 50 prodejců, kteří mají v daném časovém rozmezí výročí spolupráce a mají vazbu na zadaného prodejce.
ProcVendorsBirthday	UserID int VendorID int DateFrom date DateTo date	Vrací prvních 50 prodejců, kteří mají v daném časovém rozmezí narozeniny a mají vazbu na zadaného prodejce.

Tabulka 1: Procedury relační databáze

4 WEBOVÁ SLUŽBA

Webová služba tvoří prostředníka pro komunikaci mezi relační databází a mobilním klientem. Služba je vytvořena v programovacím jazyce C Sharp a s využitím technologie WCF. S klienty služba komunikuje pomocí protokolu SOAP, což přináší její univerzálnost a je možné ji v budoucnu použít bez jakýchkoliv úprav i pro mobilní klienty, určené pro ostatní mobilní operační systémy.

Nejdůležitější vlastností webové služby je, že převádí výstupní data definovaná v procedurách, které jsou uloženy v relační databázi, na objektový model. To klientské aplikaci velmi ulehčuje práci, protože může pracovat s pevně definovanými a předem vytvořenými objekty, které přesně odpovídají výstupním datům z relační databáze. Tento objektový model je vytvořený jako datový model pomocí technologie ADO.NET.

4.1 Veřejně dostupné funkce

Název	Vstupní parametry	Typ výstupu	Popis
Login	string login string cultureCode	User	Vrací základní informace o uživateli podle jeho uživatelského jména
CurrentProduction	int userID int vendorID string divisionCode	CurrentProduction	Vrací aktuální produkci daného prodejce
Tickets	int userID	List<Ticket>	Vrací různé provozní informace ze systému za nedávnou dobu pro daného uživatele
CRM	int userID int vendorID string cultureCode	List<CRMItem>	Vrací upozornění pro daného prodejce na události, které jsou spojené s klienty a ostatními prodejci (například narozeniny nebo výročí spolupráce)
ContentNews	int userID	List<ContentNews>	Vrací globální zprávy, které jsou určené všem uživatelům systému
VendorInfo	int userID int vendorID string cultureCode	VendorInfo	Vrací podrobné informace o daném prodejci (například jméno, osobní číslo, fotku, adresu, kontakty, atd.)

Vendors	int userID int vendorID string search string cultureCode	List<Vendor>	Vrací seznam prodejců, se kterými je daný prodejce ve spojení, umožňuje vyhledávat podle jména prodejce
Clients	int userID int vendorID string search	List<Client>	Vrací seznam klientů daného prodejce, umožňuje vyhledávat podle jména klienta
ClientInfo	int userID int clientID string cultureCode	ClientInfo	Vrací podrobné informace o daném klientovi (například jméno, adresu, kontakty, atd.)

Tabulka 2: Veřejně dostupné funkce webové služby

4.2 Zabezpečení

Komunikace mezi databázovým serverem a webovou službou je zabezpečena pomocí uživatelského jména a hesla. Zároveň je také nutné, aby webová služba i databázový server běžely na jenom fyzickém stroji nebo jejich vzájemná komunikace probíhala výhradně po důvěryhodné vnitřní podnikové síti. To dohromady zajistí dostatečné zabezpečení jejich vzájemné komunikace.

Komunikace webové služby a mobilního klienta probíhá běžnou cestou přes Internet, a proto bylo nutné zvolit robustnější zabezpečení než v případě komunikace s databázovým serverem. Využil jsem proto možností .NET Frameworku a použil zabezpečení typu `TransportWithMessageCredentials`. To využívá digitálního certifikátu, který je nainstalovaný na serveru. Pomocí něj nejprve vytvoří bezpečný tunel pro komunikaci, který běží na protokolu HTTPS a je šifrovaný pomocí SSL. Poté se skrze tento tunel posílají SOAP zprávy, které obsahují další zabezpečení ve své hlavičce. Toto zabezpečení je nutné k tomu, aby služba mohla ověřit, jestli má daný uživatel ke službě přístup. Zabezpečení v hlavičce má formát takzvaného `UsernameToken`, který v sobě obsahuje uživatelské jméno, heslo a některé další důležité časové údaje jako je například expirace.

```
<?xml version="1.0"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:xsi="http://
www.w3.org/2001/XMLSchema-instance" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:iPhoneCRMSERVICE="http://tempuri.org/"
xmlns:ns1="http://tempuri.org/Imports" xmlns:tns1="http://schemas.datacontract.org/2004/07/iPhoneCRMSERVICE" xmlns:tns2="http://
schemas.datacontract.org/2004/07/System.Data.Objects.DataClasses" xmlns:tns4="http://schemas.microsoft.com/2003/10/Serialization/"
xmlns:tns3="http://schemas.datacontract.org/2004/07/iPhoneCRM" xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-
wssecurity-utility-1.0.xsd" xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xsl:version="1.0">
  <soap:Header>
    <o:Security>
      <u:Timestamp>
        <u:Created>2012-05-11T08:17:32.159Z</u:Created>
        <u:Expires>2012-05-11T10:22:32.159Z</u:Expires>
      </u:Timestamp>
      <o:UsernameToken>
        <o:Username>karel.novak</o:Username>
        <o:Password>0FzRzII+vGoVBuQ/Twa2Kns5gb17KcfZDR6lcHVuZwE=</o:Password>
      </o:UsernameToken>
    </o:Security>
  </soap:Header>
  <soap:Body>
    <iPhoneCRMSERVICE:Login>
      <iPhoneCRMSERVICE:Login>karel.novak</iPhoneCRMSERVICE:Login>
      <iPhoneCRMSERVICE:cultureCode>cs</iPhoneCRMSERVICE:cultureCode>
    </iPhoneCRMSERVICE:Login>
  </soap:Body>
</soap:Envelope>
```

Obrázek 32: SOAP zpráva

O ověření správnosti uživatelského jména a hesla se ve službě stará třída LoginValidator. Ta přetěžuje důležitou funkci Validate, kterou služba volá vždy, když chce uživatele ověřit. V této funkci se načte uživatel s příslušným uživatelským jménem z databáze a porovná se jeho heslo s heslem, které přišlo do služby. Toto porovnání má na starost další třída Authentization, které dokáže pracovat s použitým formátem hesla, který se používá pro komunikaci mezi klientem a službou. Heslo se pro komunikaci vždy nejprve zakóduje pomocí SHA256 a poté převede na formát Base64.

4.3 Lokalizace a cache

Webová služba lokalizuje všechny data, která skrz ni prochází, a jsou závislá na zvoleném jazyku. To jsou všechny použité číselníky jako například typ prodejce a také texty, které se vytváří uvnitř služby. Proto všechny funkce služby, které vrací takováto lokalizovaná data, mají jako jeden ze vstupních parametrů i cultureCode, který určuje, v jakém jazyce budou výstupní data. Číselníky se z databáze načítají již lokalizované ve zvoleném jazyce.

Všechny data, které se ve službě často používají a zároveň jsou v čase takřka neměnná, spravuje cache. Ta data nejen načítá z databáze, ale také si je po načtení ukládá pro další použití. To značně snižuje zátěž na databázi, kterou by generovaly opakované dotazy na neměnná data v databázi. Cache spravuje všechny číselníky a také texty určené pro lokalizaci.

4.4 Demo uživatel

Třída `DemoUser` představuje demo uživatele, který není uložený v databázi, ale existuje pouze virtuálně uvnitř webové služby. Všechna jeho data jsou pevně zadána ve zdrojovém kódu třídy `DemoUser`. Z pohledu klientské aplikace je ale tento uživatel totožný s jakýmkoliv jiným uživatelem, který je uložený přímo v databázi. Uživatel se jmenuje „demo.user“ a jeho heslo je „DemoPass“. Lze jej použít pro testování webové služby i bez napojení na databázový server a pro rychlé odhalení nefunkčnosti databázového serveru. Také je vhodný jako ukázkový uživatel, díky kterému si mohou aplikaci prohlédnout i neznámí uživatelé nebo pracovníci firmy Apple při ověřování funkcí aplikace během procesu jejího vystavování na App Store.

4.5 Nastavení služby

Všechny atributy webové služby se nastavují v jediném souboru `web.config`. Ten definuje připojení k relační databázi a hlavně koncový bod webové služby, včetně jeho typu a chování. Jméno koncového bodu služby je „iPhoneCRMBinding“. Typ koncového bodu je nastavený na `basicHttpBinding` a jeho zabezpečení na již zmiňované `TransportWithMessageCredential`. Důležitou součástí chování koncového bodu je `serviceCredentials`, kde se nastavuje hodnota digitálního certifikátu, který musí být na serveru nainstalovaný, a také validátor, který ověřuje správnost přihlašovacích jmen a hesel. Kromě těchto atributů definuje soubor i množství různých časových limitů pro komunikaci s webovou službou.

Při vystavení webové služby na server je možné použít stávající `web.config` po 3 menších úpravách:

- ❖ Opravit `baseAddresses` podle adresy nového serveru.
- ❖ Nahradit hodnotu digitálního certifikátu v `serviceCertificate`.
- ❖ Opravit adresu databázového serveru, název databáze a přihlašovací jméno a heslo podle zvolené databáze v `connectionStrings`.

```

<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <system.web>
    <compilation debug="true" targetFramework="4.0">
      <assemblies>
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral, PublicKeyToken=b77a5c561934e089" />
      </assemblies>
    </compilation>
  </system.web>
  <system.serviceModel>
    <services>
      <service behaviorConfiguration="ServiceBehavior" name="iPhoneCRM.iPhoneCRMService">
        <endpoint address="" name="iPhoneCRMBinding" binding="basicHttpBinding" bindingConfiguration="basicHttpBinding" contract="iPhoneCRM.IiPhoneCRMService" />
        <host>
          <baseAddresses>
            <add baseAddress="https://localhost/" />
          </baseAddresses>
          <timeouts closeTimeout="00:00:20" openTimeout="01:00:00" />
        </host>
      </service>
    </services>
    <behaviors>
      <serviceBehaviors>
        <behavior name="ServiceBehavior">
          <serviceMetadata httpGetEnabled="true" />
          <serviceDebug includeExceptionDetailInFaults="true" />
          <dataContractSerializer maxItemsInObjectGraph="2147483647" />
          <serviceTimeouts transactionTimeout="01:00:00" />
          <serviceThrottling maxConcurrentCalls="512" maxConcurrentSessions="512" maxConcurrentInstances="2147483647" />
          <serviceCredentials>
            <serviceCertificate findValue="15 7e 23 7d da 97 6d a3 a2 1a dd 5d 08 75 af 25 67 d4 f4 75" storeLocation="LocalMachine" storeName="My" x509FindType="FindByThumbprint" />
            <userNameAuthentication userNamePasswordValidationMode="Custom" customUserNamePasswordValidatorType="iPhoneCRM.IiPhoneCRMService+LoginValidator, iPhoneCRMService"
              cachedLogonTokenLifetime="01:00:00" />
          </serviceCredentials>
        </behavior>
      </serviceBehaviors>
    </behaviors>
    <serviceHostingEnvironment multipleSiteBindingsEnabled="true" />
  </system.serviceModel>
  <bindings>
    <binding name="basicHttpBinding" closeTimeout="00:01:00" openTimeout="00:01:00" receiveTimeout="00:10:00" sendTimeout="00:01:00" allowCookies="false"
      bypassProxyOnLocal="false" hostNameComparisonMode="StrongWildcard" maxBufferSize="65536" maxBufferPoolSize="524288" maxReceivedMessageSize="65536"
      messageEncoding="Text" textEncoding="utf-8" transferMode="Buffered" useDefaultWebProxy="true">
      <readerQuotas maxDepth="32" maxStringContentLength="8192" maxArrayLength="16384" maxBytesPerRead="4096" maxNameTableCharCount="16384" />
      <security mode="TransportWithMessageCredential">
        <transport clientCredentialType="None" proxyCredentialType="None" realm="" />
        <message clientCredentialType="UserName" algorithmSuite="Default" />
      </security>
    </binding>
  </bindings>
</configuration>

```

Obrázek 33: Web.config

5 MOBILNÍ APLIKACE

Nejrozsáhlejší a z pohledu koncového uživatele i nejdůležitější součástí celého systému je mobilní aplikace. Tu jsem vytvořil ve vývojovém prostředí Xcode s využitím API Cocoa.

5.1 Struktura

Struktura aplikace je postavená na MVC modelu, který je základním stavebním kamenem API Cocoa. Grafické uživatelské rozhraní je tvořeno pouze jedním souborem, který tvoří takzvaný storyboard. Všechny controllery, které tvoří logiku aplikace, jsou uloženy ve složce Controllers. Model je zde tvořen třídami pro práci s webovou službou a je uložený ve složce Service. Obrázky, které aplikace používá, jsou umístěny do složky Images. Kopie storyboardu a souboru s lokalizovanými texty jsou uloženy ve 2 složkách cs.lproj a en.lproj, které tvoří jednotlivé jazykové verze aplikace. Vstupní bod aplikace tvoří soubor main.m. Ten spouští delegáta aplikace, který je tvořen třídou AppDelegate.

5.2 Komunikace s webovou službou

Vývojové prostředí firmy Apple nenabízí žádný nástroj, který by umožňovat vygenerovat třídy pro práci s webovou službou, tak jako to nabízí například .NET Framework. Proto jsem hledal nástroj, který by byl schopný takovýto kód v jazyce Objective-C vygenerovat. Po vyzkoušení několika nástrojů, z nichž většina nebyla funkční, jsem nakonec zvolil wsdl2objc, který fungoval ze všech nejlépe. Použitím tohoto generátoru jsem značně urychlil vytvoření aplikace i následné úpravy nutné při změnách ve webové službě i databázi, a to díky tomu, že není třeba psát velké množství opakujícího se zdrojového kódu pro obsluhu webové služby.

Pomocí wsdl2objc jsem si z wsdl souboru již hotové webové služby vygeneroval zdrojové kódy pro práci s webovou službou. Ty jsou rozděleny do 22 souborů, které jsem uložil do složky Service. Součástí těchto zdrojových kódů jsou hotové třídy, které odpovídají třídám výstupních dat z webové služby. Dále jsou to třídy obstarávající komunikaci mezi službou a aplikací, a které nabízí funkce pro synchronní i asynchronní komunikaci se všemi veřejně dostupnými funkcemi webové služby.

Komunikace mezi webovou službou a aplikací probíhá pomocí protokolu SOAP. Proto jsou součástí vygenerovaných tříd i funkce, které dokáží vytvořit validní SOAP zprávy. Další součástí jsou také parsery, které převádí data ze SOAP zpráv, které jsou ve formátu XML, na objekty příslušných výstupních dat. To programátorovi značně ulehčuje práci s webovou službou, protože z volání funkce webové služby dostává přímo objekty s daty a nemusí se starat o to, jak jsou data přenášena přes síť.

5.3 Zabezpečení

Přístup k aplikaci je zabezpečený pomocí uživatelského jména a hesla. Bez těchto údajů se uživatel nedostane přes úvodní přihlašovací obrazovku a ani bez nich není možná komunikace s webovou službou. Samotné heslo se nikdy neposílá jako obyčejný text, ale vždy se nejprve zakóduje pomocí SHA256 a poté převede na formát Base64.

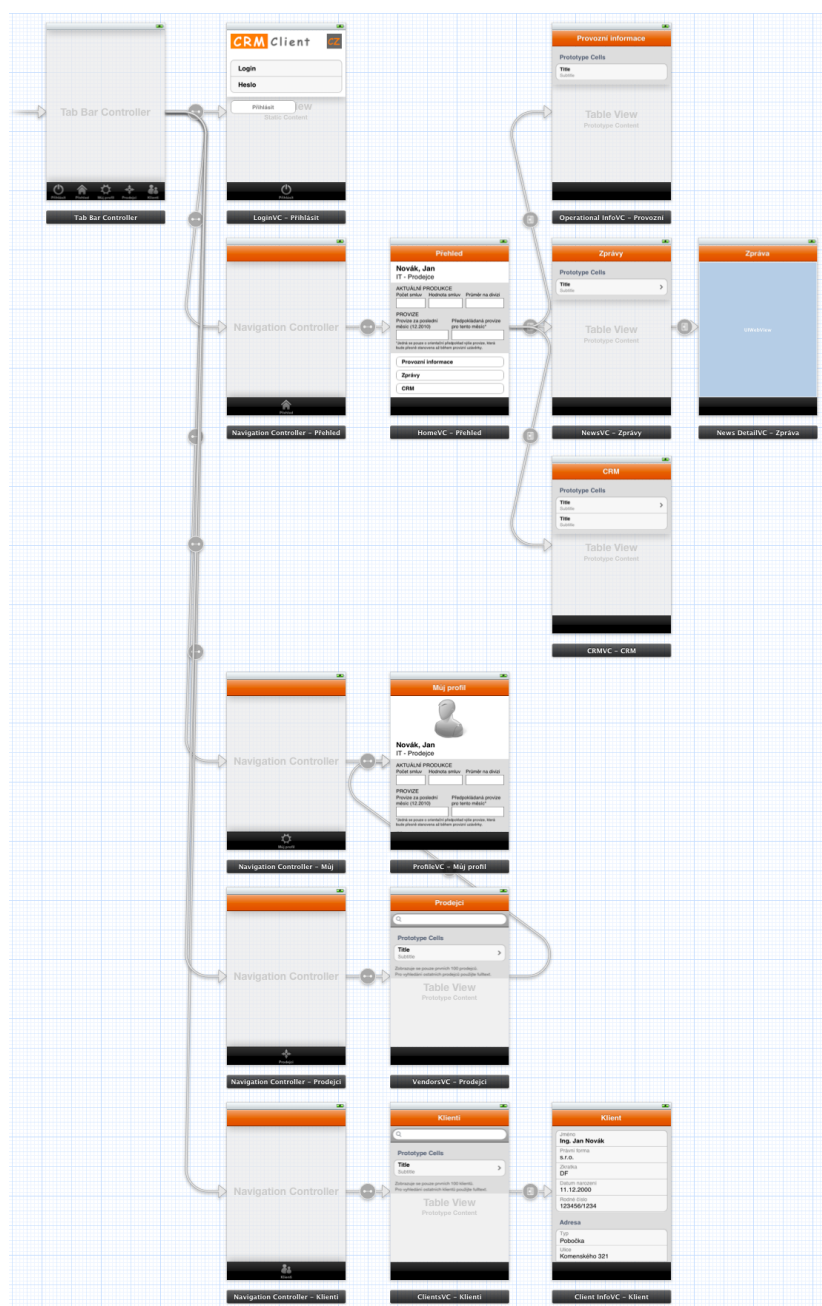
Komunikace mezi aplikací a webovou službou je zabezpečená 2 způsoby. Nejprve se vytvoří bezpečný tunel pro komunikaci, který běží na protokolu HTTPS a je šifrovaný pomocí SSL. Poté se skrze tento tunel posílají SOAP zprávy, které obsahují další zabezpečení ve své hlavičce, jak sem již popisoval výše u webové služby. Toto zabezpečení hlavičky však vygenerovaný zdrojový kód pro práci s webovou službou nepodporuje. Podpoje pouze vytvoření tunelu pomocí protokolu HTTPS. Proto jsem zabezpečení hlavičky SOAP zprávy, do vygenerovaného kódu doprogramoval.

```
<soap:Header>
  <o:Security>
    <u:Timestamp>
      <u:Created>2012-05-11T08:17:32.159Z</u:Created>
      <u:Expires>2012-05-11T10:22:32.159Z</u:Expires>
    </u:Timestamp>
    <o:UsernameToken>
      <o:Username>karel.novak</o:Username>
      <o:Password>0FzRzII+vGoVBuQ/Twa2Kns5gb17KcfZDR6lcHVuZwE=</o:Password>
    </o:UsernameToken>
  </o:Security>
</soap:Header>
```

Obrázek 34: Hlavička SOAP zprávy se zabezpečením

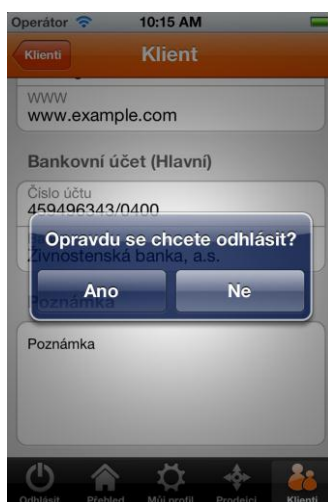
5.4 Grafické uživatelské rozhraní

Chytré telefony iPhone jsou určeny k dotykovému ovládání pomocí prstů. Proto je i grafické uživatelské rozhraní této aplikace optimalizované pro dotykové ovládání prsty. GUI je vytvořeno pomocí API Cocoa a využívá novinky v iOS 5, kterou je takzvaný storyboard. Díky němu je celé GUI součástí jediného souboru a tak má o něm programátor po celou dobu lepší přehled. Kromě prvků GUI se ve storyboardu definují i přechody mezi jednotlivými obrazovkami aplikace.



Obrázek 35: Storyboard aplikace

Kořenovým prvkem celé aplikace je Tab Bar Controller. Ten má na starost dolní ovládací pruh, který umožňuje volné přepínání mezi jednotlivými sekcemi aplikace. Sekcí je celkem 5, avšak sekce s přihlašovací obrazovkou má mezi nimi výsadní postavení. Zobrazuje se jako první obrazovka po startu aplikace a je to jediná dostupná sekce, dokud se uživatel nepřihlásí. Pokud se po přihlášení uživatel přepíná na tuto sekci, proběhne zároveň i jeho odhlášení a opětovné zakázání ostatních sekcí. Při odhlášení se také zobrazuje dialog s potvrzením odhlášení, aby nedocházelo k nechtěným odhlášením uživatele.

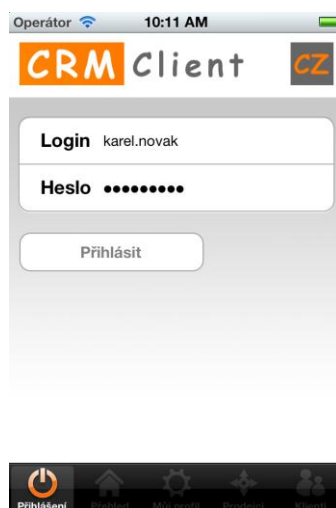


Obrázek 36: Potvrzení odhlášení uživatele

Zbylé 4 sekce obsluhuje ještě druhý Navigation Controller. Ten tvoří horní oranžový pruh, který se stará o pohyb mezi obrazovkami v dané sekci, a nebylo by bez něj možné vrátit se na předchozí obrazovku. Druhou sekci tvoří obrazovka přehledu, ze které může uživatel dál pokračovat na obrazovky provozních informací, CRM informací nebo zpráv, na které navazuje ještě obrazovka s detailem zprávy. Třetí sekce je tvořena jedinou obrazovkou s profilem přihlášeného prodejce. Čtvrtá a pátá sekce jsou velmi podobné. Obě začínají obrazovkou s vyhledáváním prodejců a v případě páté sekce klientů. Na ně pak navazují obrazovky s příslušnými detaily. U čtvrté sekce je to obrazovka s detailem zvoleného prodejce, která má navíc společný design s obrazovkou profilu přihlášeného prodejce. U páté sekce je to pak obrazovka s detailem klienta.

5.4.1 Přihlašovací obrazovka

Přihlašovací obrazovka je v horní části tvořena bílým pruhem, který v levé části obsahuje logo aplikace a v pravé části obrázek se zkratkou regionu, ke kterému dané sestavení aplikace náleží. Pod ním jsou 2 řádky pro zadání uživatelského jména a hesla. Pokud si uživatel označí jedno z polí pro zadání uživatelského jména nebo hesla, automaticky vyjede ze spodní hrany obrazovky klávesnice, která je optimalizovaná pro psaní prsty na displeji. Po zadání vstupní hodnoty je možné klávesnicí skrýt zmáčknutím tlačítka Enter nebo dotykem prstu kdekoliv mimo vstupní pole. Pod částí pro zadávání přihlašovacích údajů se nachází pouze tlačítko pro přihlášení uživatele. Během přihlašování uživatele se na obrazovce zobrazuje dialog, který indikuje probíhající přihlašování. Po úspěšném přihlášení se aplikace přepne na obrazovku přehledu. Po neúspěšném přihlášení se uživateli zobrazí upozornění, že uživatel zadal špatné uživatelské jméno nebo heslo.



Obrázek 37: Přihlašovací obrazovka

5.4.2 Přehled

Obrazovka přehledu slouží jako úvodní obrazovka po přihlášení a je rozdělena na 3 části. V horní části je jméno přihlášeného prodejce, divize, ke které patří a jeho pozice. V prostřední části jsou informace o aktuální produkci prodejce, provize za minulé období a očekávaná provize za aktuální období. V dolní části jsou 3 tlačítka, která umožňují pokračovat na navazující obrazovky s provozními informacemi, zprávami nebo CRM informacemi.

Operátor 10:11 AM

Přehled

Novák, Karel
IT - Prodejce

AKTUÁLNÍ PRODUKCE

Počet smluv	Hodnota smluv	Průměr na divizi
3	131	200

PROVIZE

Provize za poslední měsíc (05.2012)	Předpokládaná provize pro tento měsíc*
0,00 Kč	13,10 Kč

*Jedná se pouze o orientační předpoklad výše provize, která bude přesně stanovena až během provizní uzávěrky.

Provozní informace

Zprávy

CRM

Odhlásit Přehled Můj profil Prodejci Klienti

Obrázek 38: Obrazovka přehledu

5.4.3 Provozní informace

Obrazovka provozních informací je tvořena seznamem se systémovými tikety. U každé položky seznamu se zobrazuje v pravé části název tiketu a datum a čas jeho poslední úpravy a v levé části ikona, která odpovídá typu daného tiketu a také tomu, jestli je tiket otevřený nebo uzavřený.

Operátor 10:11 AM

Přehled **Provozní informace**

Reklamacie 2
24.04.2012 00:00:00

Reklamacie 1
23.04.2012 00:00:00

Dotaz 3
13.04.2012 00:00:00

Odhlásit Přehled Můj profil Prodejci Klienti

Obrázek 39: Obrazovka provozních informací

5.4.4 Zprávy

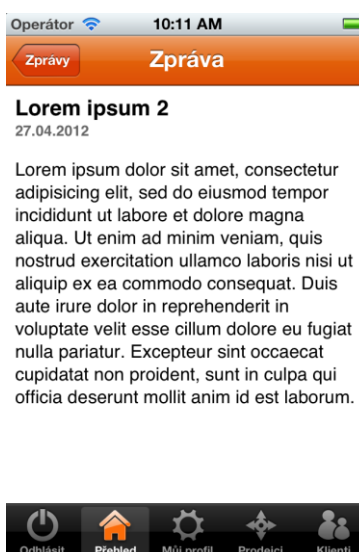
Obrazovka zpráv je tvořena seznamem s názvy globálních zpráv. Jednotlivé zprávy je možné také vybrat, a tím se přesunout na další obrazovku s detailem dané zprávy.



Obrázek 40: Obrazovka zpráv

5.4.5 Detail zprávy

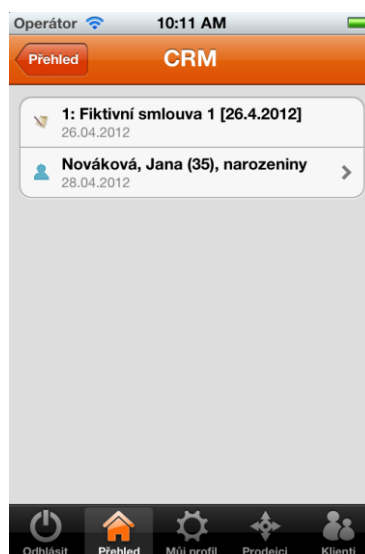
Celá obrazovka detailu zprávy je tvořena prvkem Web View, který dokáže zobrazit webový obsah. Do něj se vkládá zpráva, která je již ve formátu HTML. Před zprávou se ještě v aplikaci připojí název zprávy a její datum, které jsou také doplněny o HTML.



Obrázek 41: Obrazovka detailu zprávy

5.4.6 CRM

Obrazovka CRM informací je také tvořena jedním seznamem. Jednotlivé informace v seznamu se pak skládají z názvu, který stručně popisuje danou událost, a data, ke kterému se událost vztahuje. U každé události je také vlevo zobrazená ikona, které odpovídá typu dané události. Události, které mají vazbu na klienta nebo prodejce, je také možné vybrat a tím se přesunout na obrazovku s klienty nebo prodejci, kde je daná osoba předvyplněná ve vyhledávacím poli a položky v seznamu jsou podle něj vyhledané.



Obrázek 42: Obrazovka CRM informací

5.4.7 Prodejci

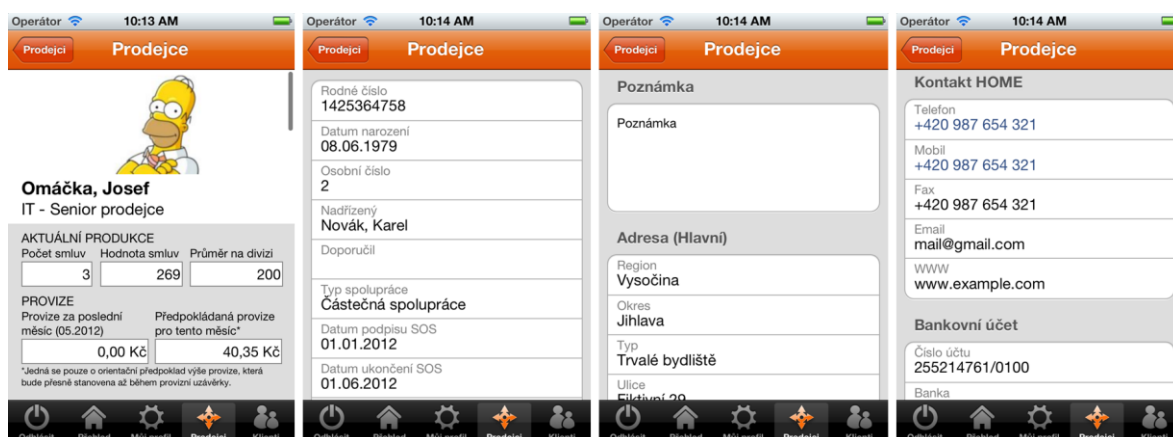
Obrazovka prodejců je v horní části tvořena pruhem pro vyhledávání. Ten je z velké části tvořen vstupním polem pro zadání jména prodejce. Po jeho označení se vysune na obrazovku klávesnice, která funguje stejně jako v případě přihlašovací obrazovky. V pravé části pruhu je křížek, který po jeho zmáčknutí vymaže obsah vstupního pole. Vyhledání se spouští tlačítkem hledat na klávesnici. Po úspěšném vyhledání se ve zbytku obrazovky zobrazí seznam s nalezenými prodejci. Položky seznamu zobrazují jméno prodejce a pod ním jeho divizi a pozici. Protože je seznam omezen na maximálně 100 zobrazených poradců, zobrazí se v případě dosažení tohoto limitu pod seznamem i oznámení o jeho dosažení a doporučení, aby uživatel zpřesnil své vyhledávání. Výběrem některého prodejce se uživatel přesune na obrazovku s detailem příslušného prodejce.



Obrázek 43: Obrazovka prodejců

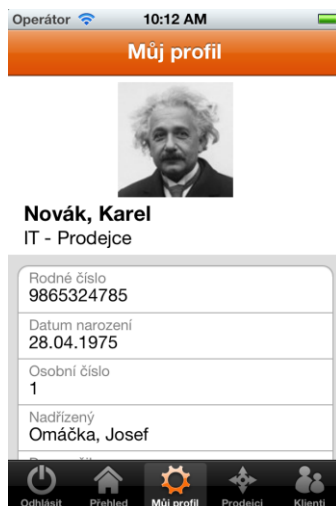
5.4.8 Detail prodejce

Detail prodejce začíná nahoře fotkou daného prodejce. Pokud prodejce žádnou fotku nemá, je místo ní zobrazený základní obrázek symbolizující portrét prodejce. Pod fotkou je zobrazeno jméno prodejce, jeho divize a pozice. Pod nimi následuje část produkce, která je totožná s částí produkce na obrazovce přehledu. Dále následuje seznam s různými dalšími informacemi o prodejci, který je rozdělený na sekce obecných informací, poznámku, adresu, 2 kontakty a bankovní účet. U telefonních čísel v sekcích kontaktů je možné výběrem daného telefonního čísla spustit jeho vytočení.



Obrázek 44: Části obrazovky detailu prodejce

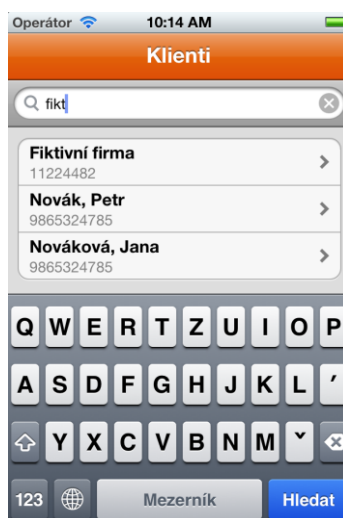
Obrazovka detailu prodejce je také sdílená s obrazovkou profilu přihlášeného prodejce. Tyto obrazovky se liší pouze tím, že u profilu přihlášeného prodejce je skrytá sekce aktuální produkce, která v tomto případě není třeba zobrazovat. Jinak jsou obě obrazovky shodné jak vizuálně tak funkčně.



Obrázek 45: Část obrazovky můj profil

5.4.9 Klienti

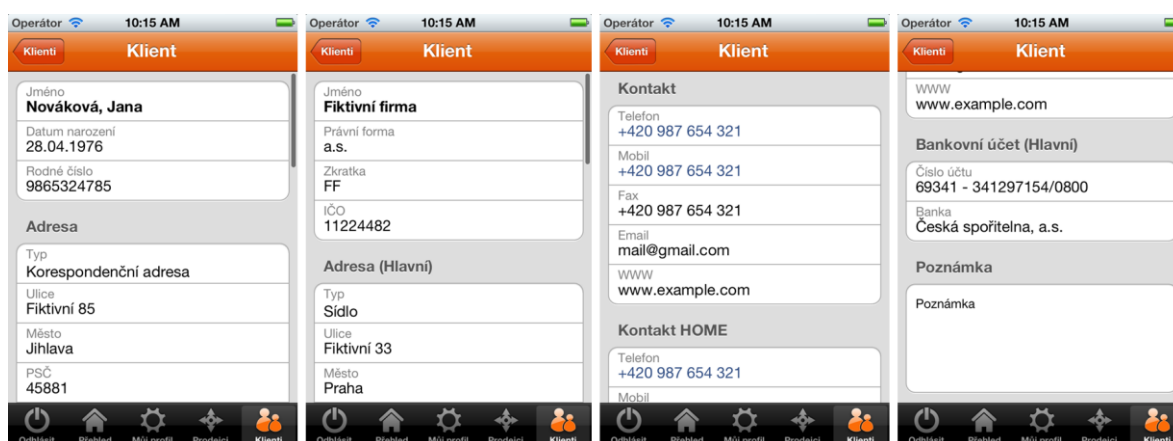
Obrazovka klientů funguje stejně a také má stejný design jako obrazovka prodejců pouze s tím rozdílem, že umožňuje vyhledávání v klientech aktuálně přihlášeného uživatele. Jednotlivé položky seznamu zobrazují jméno klienta a jeho rodné číslo, nebo v případě firmy je to její název a IČO. Výběrem některého klienta se uživatel přesune na obrazovku detailu příslušného klienta.



Obrázek 46: Obrazovka klientů se zobrazenou klávesnicí

5.4.10 Detail klienta

Detail klienta je tvořen jedním seznamem s informacemi o daném klientovi, který je rozdělený na několik sekcí. První sekce se základními informacemi se mění v závislosti na tom, jestli je klientem firma nebo soukromá osoba. Na ni pak navazují sekce adresy, 2 kontaktů, bankovního účtu a poznámky, které jsou velmi podobné nebo úplně stejné jako sekce v detailu prodejce. I zde je u telefonních čísel v sekcích kontaktů možné výběrem daného telefonního čísla spustit jeho vytočení.



Obrázek 47: Části obrazovky detailu klienta

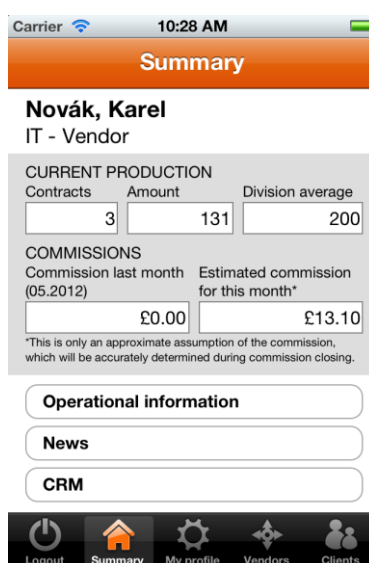
5.5 Lokalizace

Aplikace je lokalizovaná do českého a anglického jazyka. Výběr lokalizace, která bude použita, je ponechán na operačním systému. Ten vybírá tu lokalizaci, která se shoduje s jazykem, který je nastavený pro celý telefon. Pokud je nastavený jazyk, do kterého není aplikace přeložena, nastaví systém jazyk, který byl použitý při vývoji aplikace. Tento jazyk nastavuji v aplikaci podle jazyka regionu, ke kterému patří dané sestavení aplikace.

Lokalizace v API Cocoa se provádí tak, že se vytvoří duplikát lokalizovaného souboru pro každý jazyk, který chceme v aplikaci podporovat. Proto jsou součástí zdrojových kódů aplikace 2 soubory se storyboardem a 2 soubory App.strings. Soubory App.strings slouží jako zdroj lokalizovaných textů, které je nutné načíst dynamicky až za běhu aplikace. To jsou například dynamicky se měnící popisky u sekce pro přihlášení a odhlášení. Jako součást App.strings má aplikace také cultureCode, který obsahuje zkratku daného jazyka. Aplikace ho používá k určení aktuálně vybraného jazyka a také ho posílá jako parametr do webové služby, aby věděla do jakého jazyka má načítaná data lokalizovat.

Toto řešení pro zjištění aktuálního jazyka jsem zvolil, protože jsem při testování aplikace zjistil, že žádná z funkcí API, která má vracet informace o jazyku aplikace, nedokázala vždy spolehlivě vrátit aktuálně vybraný jazyk.

Toto řešení lokalizace GUI má výhodu v tom, že kromě samotných textů můžeme pro jednotlivé jazykové verze přizpůsobit i jednotlivé prvky uživatelského rozhraní. Naopak velkou nevýhodou je nutnost upravovat každou jazykovou verzi storyboardu zvlášť, což může být dost časově náročné, pokud provádíme větší úpravy GUI nebo ho rozšiřujeme a zároveň podporujeme větší počet jazyků.



Obrázek 48: Anglická jazyková verze aplikace

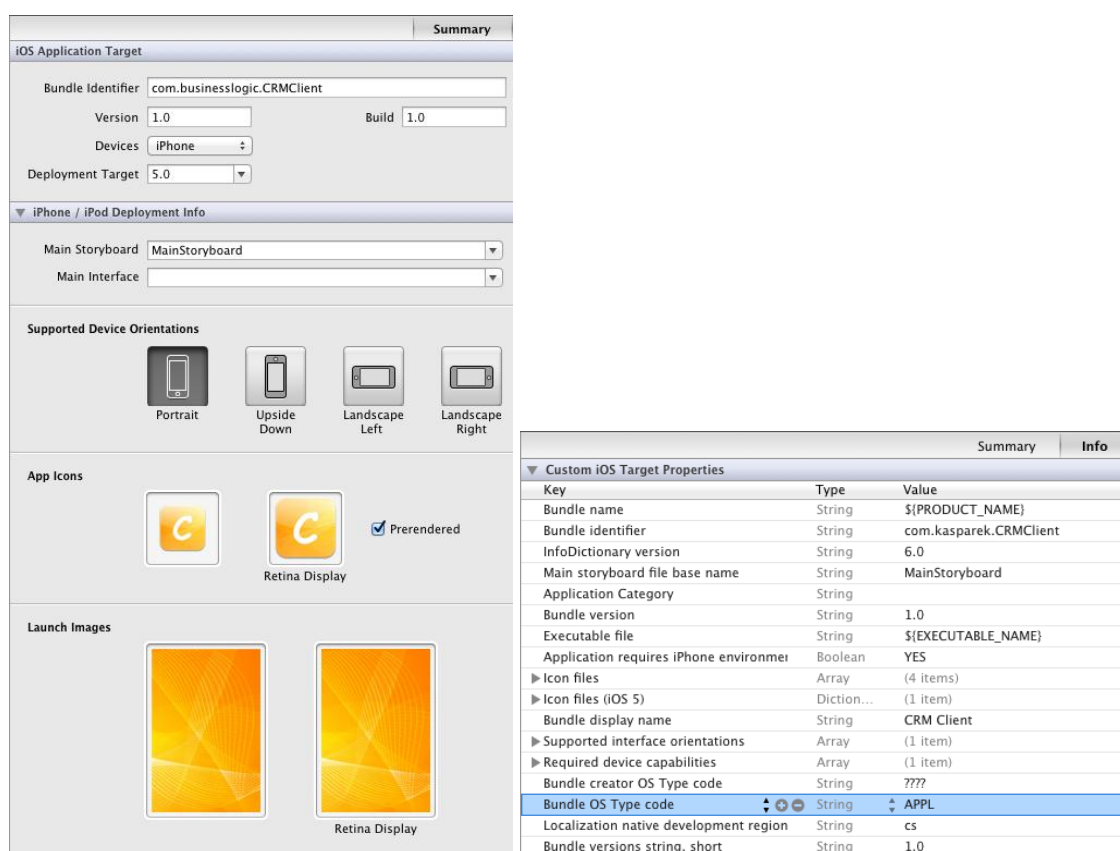
5.6 Nastavení aplikace

Aby bylo možné aplikaci debuggovat a následně i publikovat na App Store je nutné nejprve správně nastavit sestavení aplikace. Asi nejdůležitější je BundleID, které identifikuje danou aplikaci, a proto musí být unikátní. Podle konvence by mělo začínat na „com.“. Kromě identifikátoru se k bundle nastavuje i jeho jméno, zobrazované jméno a verze. V případě této aplikace je také nutné správně nastavit development region. Ten určuje kromě základního jazyka také region, ke kterému aplikace náleží.

Velmi důležitým parametrem je development target. Ten určuje minimální verzi iOS, na kterém jde aplikace spustit. V případě této aplikace je to verze 5.0, protože nižší verzi není možné použít z důvodu využití storyboardu při vývoji aplikace. Ten je podporovaný až od verze iOS 5.0.

Ve zdrojovém kódu controlleru LoginVC je třeba nastavit proměnnou ADDRESS, která určuje adresu webové služby, ke které se aplikace připojuje.

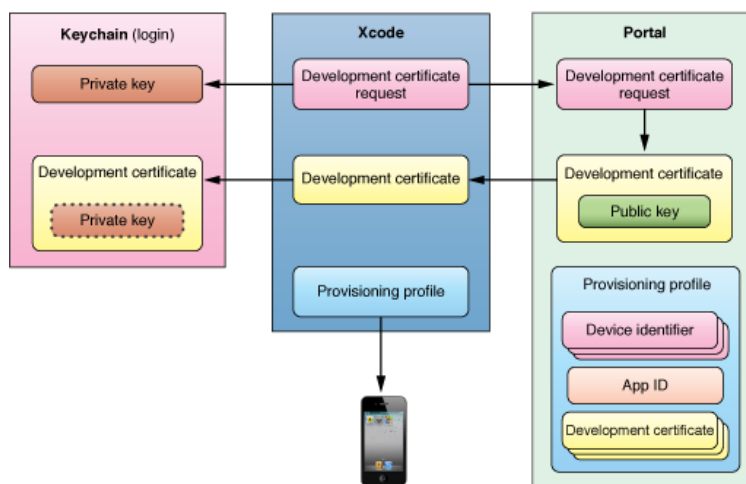
Důležitou, ale nepovinnou součástí nastavení je i ikona aplikace, díky které budou moci uživatelé lépe identifikovat aplikaci mezi ostatními v seznamu aplikací telefonu. Ikonu je třeba nahrát jak v nízkém, tak ve vysokém rozlišení kvůli podpoře všech velikostí displejů. Další nepovinnou součástí, kterou je vhodné nastavit je launch image. Ten se uživateli zobrazuje při spouštění aplikace před tím, než se zobrazí první obrazovka s přihlášením. Tento obrázek je také třeba nahrát v nízkém i vysokém rozlišení.



Obrázek 49: Ukázky nastavení sestavení aplikace

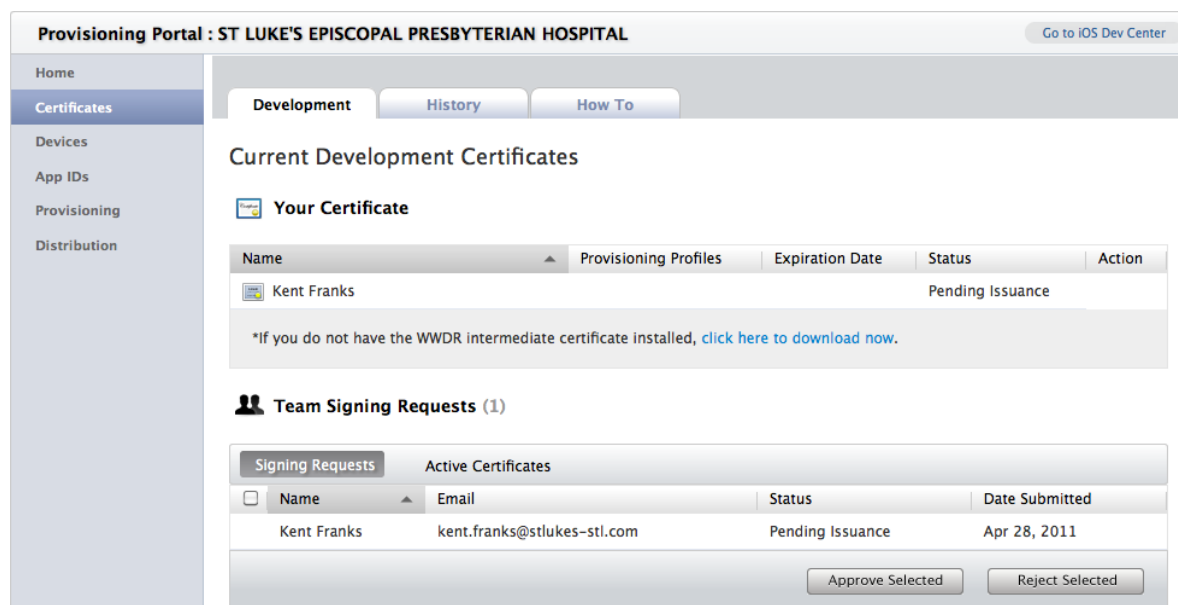
6 DEBUGGOVÁNÍ MOBILNÍ APLIKACE

Před tím, než může vývojář začít debuggovat své aplikace na reálném iPhone, musí si obstarat potřebné vývojářské certifikáty a nainstalovat je do svého počítače, který používá k vývoji aplikací. První krok k jejich získání je vytvoření vývojářského účtu u společnosti Apple. Ten je možné založit na adrese <https://developer.apple.com/programs/ios/>.



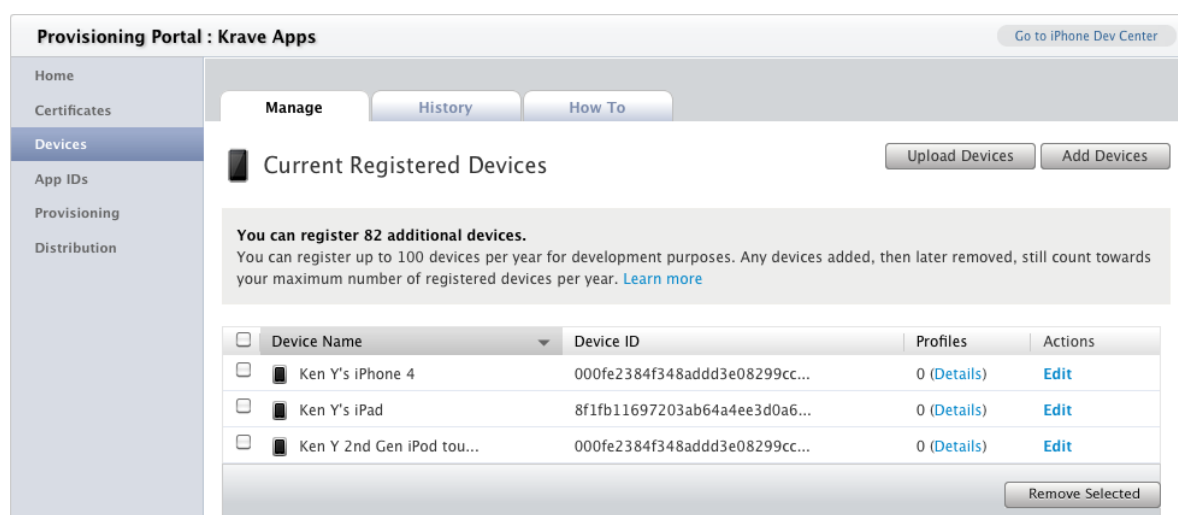
Obrázek 50: Digitální prostředky potřebné k vývoji [27]

Po vytvoření vývojářského účtu je již možné zažádat o vytvoření vývojářského certifikátu. Soubor s žádostí o certifikát si vygenerujeme pomocí aplikace Keychain, která je základní součástí operačního systému. Tuto žádost najdeme v menu Keychain Access → Certificate Assistant → Request a Certificate from a Certificate Authority. Poté se přihlásíme na Provisioning Portal na adrese <https://developer.apple.com/ios/manage/overview/index.action>, kde můžeme spravovat všechny své digitální prostředky nutné pro vývoj a publikování iOS aplikací. Zde si v sekci Certificates → Development zažádáme s pomocí vygenerované žádosti o nový vývojářský certifikát. V této chvíli musíme počkat, než naši žádost schválí takzvaný Team agent. To je některý ze členů vývojového týmu, který má na starost správu vývojového týmu, a mezi jeho pravomoci patří i schvalování certifikátů. Když Team agent schválí certifikát, objeví se nám v sekci Certificates → Development daný certifikát, který si stáhneme na svůj vývojový počítač. Poté stačí soubor s certifikátem spustit a certifikát se sám nainstaluje. To že byl certifikát správně nainstalovaný, si můžeme ověřit v aplikaci Keychain, kde bude vidět certifikát i s podrobnostmi o něm.



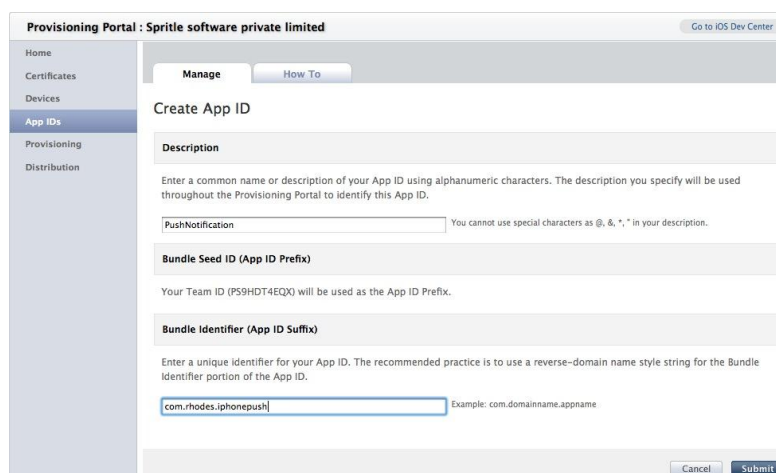
Obrázek 51: Sekce Provisioning Portalu pro vytváření vývojářského certifikátu [28]

Dalším krokem je přidání zařízení, na kterých budeme aplikace testovat, do Provisioning Portalu. K tomu slouží sekce Devices → Manage. Abychom mohli dané zařízení přidat, musíme znát jeho UDID, což je unikátní identifikátor každého zařízení, který se skládá ze 40 hexadecimálních znaků. UDID můžeme zjistit například v aplikaci iTunes po tom, co připojíme zařízení k počítači kabelem. Počet zařízení, které je možné přidat, omezuje společnost Apple na 100 zařízení ročně.



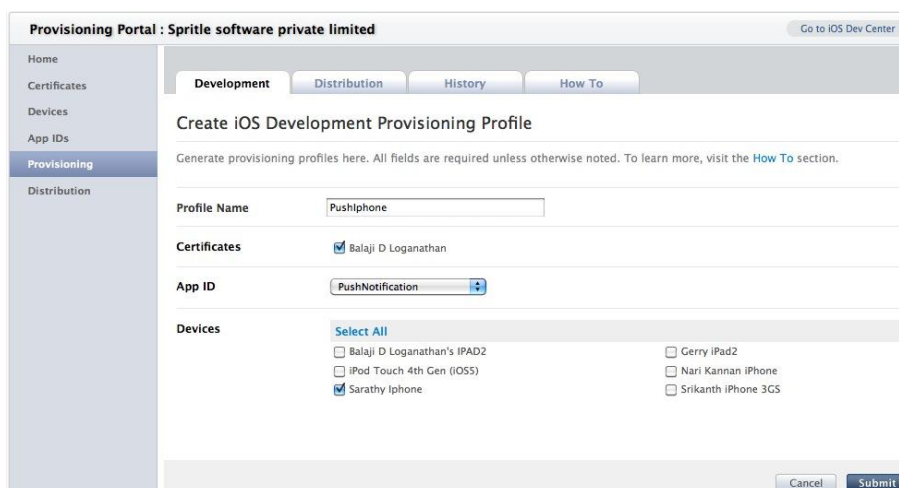
Obrázek 52: Sekce Provisioning Portalu pro přidávání zařízení [29]

Další částí, kterou je třeba vytvořit na Provisioning Portalu je App ID, což je unikátní identifikátor aplikace. Ten můžeme vytvořit buď jako přesně specifikovaný pouze pro jednu aplikaci, nebo jako prefix identifikátoru, který může sloužit pro více aplikací. App ID se vytváří v sekci AppIDs → Manage.

The screenshot shows the 'Provisioning Portal : Spritle software private limited' interface. On the left is a sidebar with links: Home, Certificates, Devices, App IDs (selected), Provisioning, and Distribution. The main area has tabs for 'Manage' and 'How To'. The 'Create App ID' form contains three sections: 'Description' with a text input 'PushNotification' and a warning about special characters; 'Bundle Seed ID (App ID Prefix)' with a note that the team ID 'P59HDT4EQX' will be used; and 'Bundle Identifier (App ID Suffix)' with a text input 'com.rhodes.iphonepush' and an example 'com.domainname.appname'. 'Cancel' and 'Submit' buttons are at the bottom right.

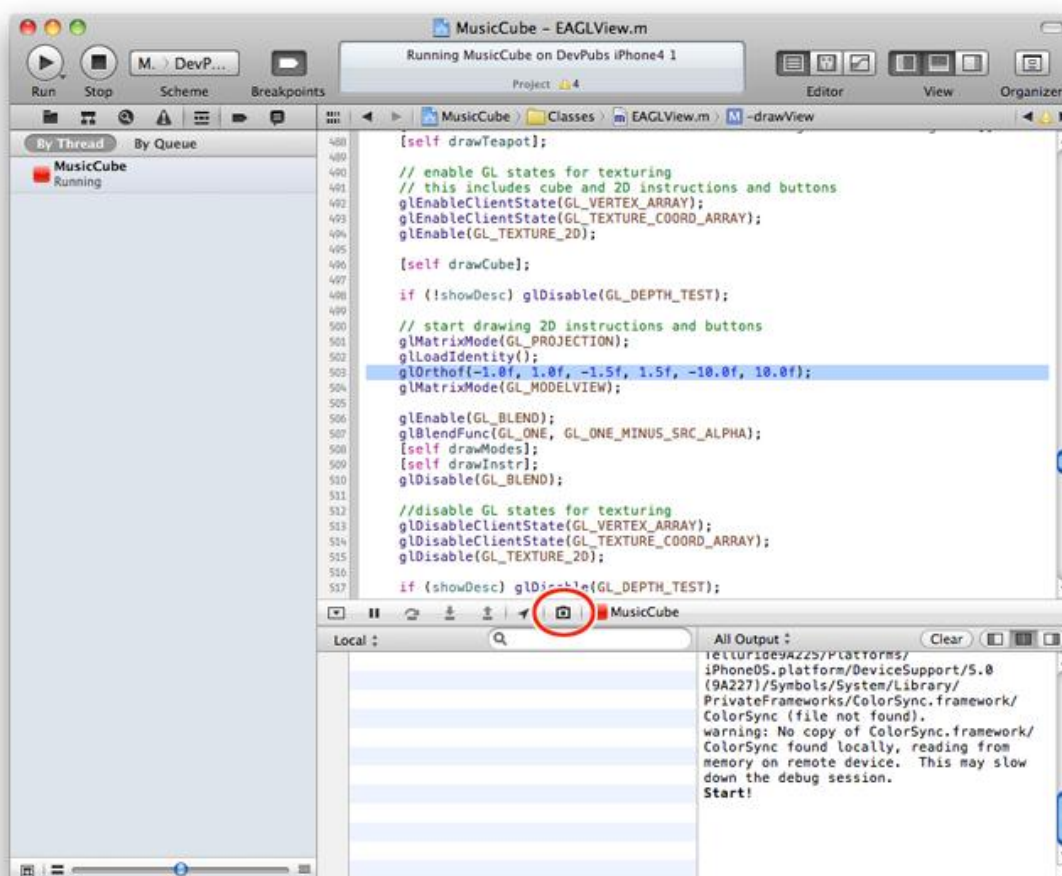
Obrázek 53: Vytváření App ID [30]

Poslední částí, kterou je třeba vytvořit na Provisioning Portalu je Development Provisioning Profile. Ten spojuje dohromady vývojářské certifikáty, App ID a zařízení, které jsme přidali do Provisioning Portalu a vytváří se v sekci Provisioning → Development. Po jeho vytvoření si ho v této sekci stáhneme na svůj vývojářský počítač. Poté spustíme IDE Xcode a následně i stažený soubor s Development Provisioning Profile, čímž se nám automaticky nainstaluje do Xcode. Následně připojíme iPhone, na kterém budeme debuggovat, k počítači kabelem a Xcode si ho automaticky spáruje s certifikátem i Development Provisioning Profilem.

The screenshot shows the 'Provisioning Portal : Spritle software private limited' interface. The sidebar is the same as in the previous image. The main area has tabs for 'Development' (selected), 'Distribution', 'History', and 'How To'. The 'Create iOS Development Provisioning Profile' form includes a note: 'Generate provisioning profiles here. All fields are required unless otherwise noted. To learn more, visit the How To section.' The form fields are: 'Profile Name' with input 'Pushiphone'; 'Certificates' with a checked checkbox for 'Balaji D Loganathan'; 'App ID' with a dropdown menu showing 'PushNotification'; and 'Devices' with a 'Select All' link and a list of checkboxes for various devices. The 'Saratthy Iphone' checkbox is checked. 'Cancel' and 'Submit' buttons are at the bottom right.

Obrázek 54: Vytváření Development Provisioning Profile [30]

V této chvíli máme Xcode připravený k debuggování aplikací na zařízení iPhone kdykoliv, když budeme mít iPhone připojený k počítači kabelem. V levém horním rohu Xcode si vybereme připojený iPhone jako zařízení, na kterém chceme aktuálně debuggovat aplikace a stiskneme tlačítko Run, které nejprve vytvoří sestavení aplikace a potom spustí aplikaci na připojeném iPhone. S aplikací teď pracujeme tak, jako by to dělal uživatel aplikace, a snažíme se tak odhalit chyby a nedostatky v aplikaci. K jejich odhalování používáme prostředky Xcode jako jsou breakpoints, které automaticky zastaví běh aplikace na určeném řádku zdrojového kódu, krokování, které nám umožňuje procházet kód řádek po řádku a sledovat, jak se aplikace chová, nebo náhledy aktuálních hodnot proměnných, které nám umožňují odhalit například chyby ve výpočtech.



Obrázek 55: Xcode se spuštěným debuggováním na zařízení iPhone [27]

7 PUBLIKOVÁNÍ MOBILNÍ APLIKACE

Aplikace pro zařízení iPhone je možné publikovat dvěma způsoby. První je způsob Ad Hoc, který umožňuje přímo vytvořit instalační soubor. Jeho použití je však omezené pouze na zařízení, které jsou součástí Distribution Provisioning Profilu. Těchto zařízení může být ale maximálně 100, a proto se tento způsob používá pouze pro distribuci aplikace k testerům. Druhý způsob je publikování přímo na App Store. Tento způsob není nijak omezený a oba způsoby publikování jsou se většinou kroků takřka totožné, a proto budu dále popisovat pouze publikování na App Store.

I u publikování aplikace je nejprve nutné si zajistit vývojářský účet a poté postupně vytvořit a nainstalovat Distribution Certificate, který je obdobou Development Certificate, Distribution Provisioning Profile, který je obdobou Development Provisioning Profile a vytvořit App ID aplikace. Tyto kroky jsou takřka totožné s kroky, které jsem popisoval v kapitole 6, a provádí se pouze v sousedních sekcích Distribution, nebo už byly provedeny při přípravě na debuggování aplikace.

The screenshot shows the 'Provisioning Portal : Krave Apps' interface. On the left is a sidebar with links: Home, Certificates, Devices, App IDs, Provisioning (selected), and Distribution. The main area has tabs for Development, Distribution (selected), History, and How To. The title is 'Modify iPhone Distribution Provisioning Profile'. The form contains the following fields:

- Distribution Method:** Radio buttons for App Store and Ad Hoc (selected).
- Profile name:** Text input field containing 'Tweeb Ad Hoc'.
- Distribution Certificate:** Text field showing 'Krave Apps (expiring on Sep 27, 2010)'.
- App ID:** Dropdown menu showing 'Tweeb'.
- Devices (optional):** A section with the instruction 'Select up to 100 devices for distributing the final application; the final application will run only on these selected devices.' It includes a 'Select All' link and a list of devices with checkboxes:
 - ☐ Ken Y 2nd Gen iPod touch
 - ☒ Ken Y's iPhone 4
 - ☐ Ken Y's iPad

At the bottom right are 'Cancel' and 'Submit' buttons.

Obrázek 56: Vytváření Distribution Provisioning Profile [29]

Když máme Xcode připravený na publikování, přichází řada na vytvoření profilu aplikace na portálu iTunes Connect, který má adresu <https://itunesconnect.apple.com/>. Zde je třeba postupně zadat všechny důležité informace související s vystavením aplikace, jako je název aplikace, její popis, klíčové slova, adresy na podporu k aplikaci, cena, za kterou se bude aplikace prodávat, datum, od kterého chceme aplikaci publikovat a rating aplikace. Důležité je napsat do poznámky k aplikaci přihlašovací jméno a heslo, které budou využívat pracovníci společnosti Apple při schvalování aplikace před jejím publikováním. Kromě těchto údajů je také třeba nahrát do profilu velkou ikonu aplikace a až 5 náhledů obrazovek aplikace. Když máme profil aplikace připravený, použijeme tlačítko Ready to Upload Binary, které se zobrazuje vpravo nahoře v profilu aplikace. Po dokončení průvodce je aplikace připravená na nahrání sestavení aplikace ke schválení pracovníky společnosti Apple.

Apple iTunes Connect Sign Out

Testovací

Enter the following information in **English**.

Version Information

Version Number: 1.0

Copyright: 2012 Kasperek

Primary Category: Catalogs

Secondary Category (Optional): Education

Review Notes (Optional): Login: demo.user
Password: DemoPass

Rating

For each content description, choose the level of frequency that best describes your app.

[App Rating Details](#)

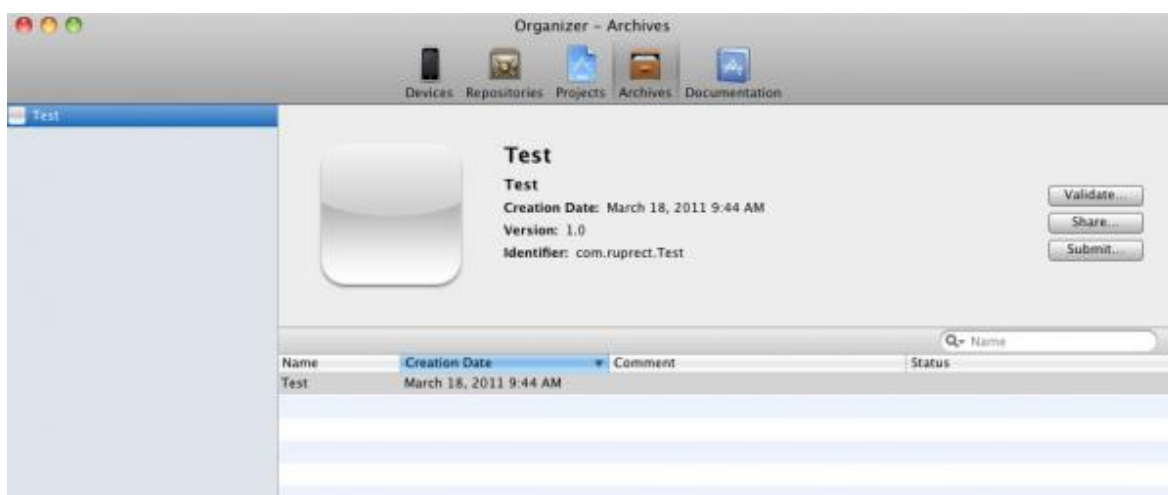
Apps must not contain any obscene, pornographic, offensive or defamatory content or materials of any kind (text, graphics, images, photographs, etc.), or other content or materials that in Apple's reasonable judgment may be found objectionable.

Apple Content Descriptions	None	Infrequent/Mild	Frequent/Intense
Cartoon or Fantasy Violence	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Realistic Violence	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Sexual Content or Nudity	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Profanity or Crude Humor	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
Alcohol, Tobacco, or Drug Use or References	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>

4+
App Rating

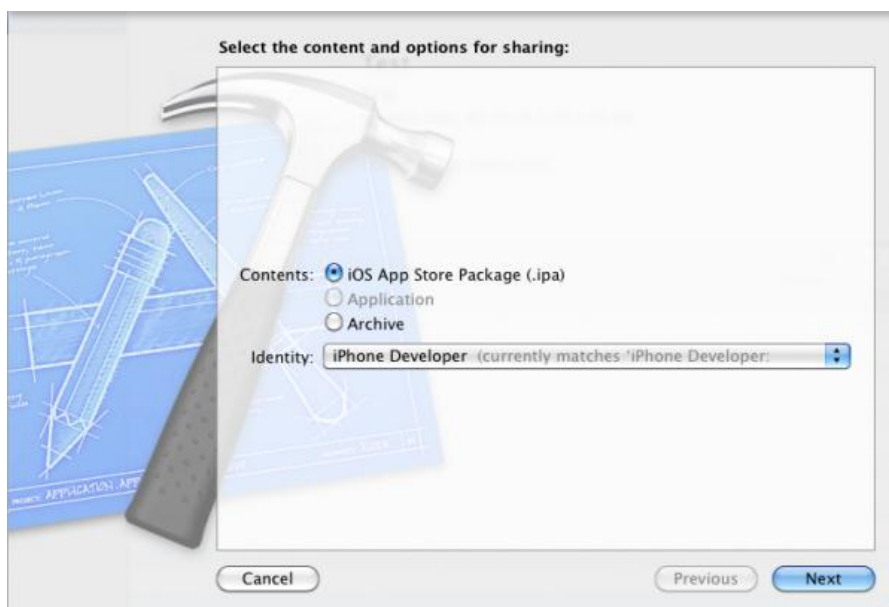
Obrázek 57: Vytváření profilu aplikace v iTunes Connect

Pro další kroky se přesuneme zpět do Xcode. V levém horním rohu vybere iOS Device nebo zařízení iPhone, které máme připojené kabelem k počítači. Poté vytvoříme sestavení aplikace pomocí položky menu Product → Archive. Následně se přesuneme do nástroje Organizer, který je součástí Xcode. V něm přejdeme do záložky Archives a vybereme sestavení, které jsme právě vytvořili.



Obrázek 58: Organizer se sestavením aplikace [31]

Sestavení aplikace nejprve ověříme pomocí tlačítka Validate. Pokud jsme aplikaci nastavili správně a dobře vytvořili profil na iTunes Connect, měl by se nám tento profil zobrazit v jenom z kroků průvodce ověřením aplikace. Pokud se nám podařilo projít celým průvodcem a následná validace nenašla žádný problém, můžeme pokračovat nahráním sestavení na App Store pomocí tlačítka Submit. Zde opět projdeme průvodcem, který je z větší části totožný s průvodcem ověření aplikace, a na jeho konci se nahraje sestavení aplikace na servery společnosti Apple, kde bude čekat na schválení. Když je aplikace schválená, tak ji systém automaticky publikuje k datu, které jsme zadali při vytváření profilu aplikace. Pokud aplikace není schválena, přejde do stavu Rejected a je třeba aplikaci opravit a proces publikování aplikace zopakovat. Abychom věděli, proč byla aplikace odmítnuta, je v profilu aplikace zanechána pracovníky společnosti Apple poznámka, která popisuje, proč nebyla aplikace schválená.



Obrázek 59: Průvodce vystavením sestavení aplikace na App Store [31]

ZÁVĚR

Cílem této diplomové práce bylo vytvořit mobilní aplikaci pro operační systém iOS 5, včetně všech vrstev její vícevrstvé architektury. Pro ukládání zdrojových dat aplikace jsem vytvořil relační databázi, která běží na Microsoft SQL Serveru. Databáze má podobu jednoduchého CRM systému, a je koncipována jako součást většího, již existujícího, systému. Komunikaci mezi klientskou aplikací a databázovým serverem zprostředkovává webová služba, která je vytvořená pomocí technologií .NET Frameworku. Hlavní část práce pak tvoří klientská aplikace, která má za úkol zpřístupnit důležitá data CRM systému v moderním a přehledném uživatelském rozhraní. Aplikace je určena pro chytré telefony iPhone s operačním systémem iOS 5.0 a vyšším, a je vytvořená pomocí API Cocoa a programovacího jazyka Objective-C. Na konci práce jsem také popsal proces debuggování vytvořené aplikace na reálném iPhone a proces vystavení aplikace na App Store.

Vytvořený systém splňuje všechny požadavky, které na něj byly kladeny zadáním práce a všechny jeho vrstvy jsou postavené na nejmodernějších technologiích od společností Microsoft a Apple. Grafické uživatelské rozhraní je vytvořené podle nejnovějších trendů, a proto je přizpůsobené pro dotykové ovládání prsty. Veškerá komunikace v rámci systému i důležitá data uložená v databázi jsou zabezpečena tak, aby nemohlo dojít k uniku jakýchkoliv údajů do nepovolaných rukou. Struktura systému je vytvořena tak, že je snadné ji rozšířit o další mobilní klientské aplikace, které poběží na jiných operačních systémech jako například Android nebo Windows Phone.

Celý systém jsem během vývoje i po jeho dokončení důkladně testoval a na základě zjištěných nedostatků během vývoje jsem aplikaci průběžně zlepšoval. Při testech výsledného systému se poté prokázala jeho spolehlivost a systém si dokázal dobře poradit i se situacemi, kdy jsem simuloval selhání databázového serveru nebo webové služby.

Jako další možnost rozšiřování aplikace v budoucnu bych viděl její těsné propojení s webovou aplikací, která by byla součástí předpokládaného většího systému. Mobilní aplikace by tak mohla na svých okrajích, kde už není schopná poskytnout podrobnější informace, plynule přecházet na webovou aplikaci, kterou by zobrazoval internetový prohlížeč Safari.

ZÁVĚR V ANGLIČTINĚ

The main objective of this thesis was to create a mobile application for iOS 5 operating system, including all layers of its multilayer architecture. Source data of application is stored in Microsoft SQL Server relational database. The database structure corresponds to a simple CRM system, and is conceived as part of a larger existing system. Communication between client application and database server provides web service that is created by .NET Framework technology. The main part of this thesis consists on client application that is designed for access to important data in CRM system and uses modern and clear user interface. The application is designed for iPhone smartphones with operational system iOS 5.0 and higher, and is created with API Cocoa and programming language Objective-C. At the end of this thesis I have also described process of debugging application on real iPhone device and process of publishing application to App Store.

Created system meets all the requirements placed on it by this thesis and all of its layers are built with the latest technologies from companies Microsoft and Apple. Graphical user interface is designed according to the latest trends and optimized for touch control. All communication within the system and all important data in database are secured to prevent any data leakage into wrong hands. The structure of the system is also designed to be easily expandable to other mobile client applications that run on other operating systems such as Android or Windows Phone.

I thoroughly tested whole system during development and after its completion and I continually improved the application during development based on founded deficiencies. Application demonstrated its reliability during the tests of the system, and the system was able to cope well with situations when I simulated the failure of the database server or web service.

As a further option of extension of application in the future I saw its tight integration with the web application, which would be anticipated part of a larger system. Mobile application could seamlessly move into the web application on its edges and web application provides more information from system in Safari web browser.

SEZNAM POUŽITÉ LITERATURY

- [1] Apple Historie - iPhone. [online]. [cit. 2012-04-02]. Dostupné z: <http://applehistorie.cz/historie-produkty/iphone/>
- [2] Apple - iPhone 4S - The most amazing iPhone yet. [online]. [cit. 2012-04-02]. Dostupné z: <http://www.apple.com/iphone/>
- [3] DUNCAN, Andy. *Objective-C: pocket reference* [online]. O'Reilly, 2002 [cit. 2012-04-02]. ISBN 978-0-596-00423-1. Dostupné z: http://aleph.utb.cz/F/?func=service&doc_library=UTB01&doc_number=000034649&line_number=0002&func_code=WEB-BRIEF&service_type=MEDIA
- [4] PIPER, Ian. *Learn Xcode tools: for Mac OS X and iPhone development* [online]. Apress, 2009 [cit. 2012-04-02]. ISBN 978-1-4302-7220-5. Dostupné z: <http://www.springerlink.com/content/rh6557/?p=e5e26e7ffff44e259ea934b6423d1973&pi=69>
- [5] Apple Developer. [online]. [cit. 2012-04-03]. Dostupné z: <https://developer.apple.com>
- [6] ROBINSON, Simon. *C Sharp: programujeme profesionálně*. Vyd. 1. Brno: Computer Press, 2003. 1130 s. ISBN 80-251-0085-5.
- [7] HUMMEL, Joel. *LINQ: The future of data access in C Sharp 3.0* [online]. Sebastopol, Calif.: O'Reilly, 2006 [cit. 2012-04-13]. ISBN 978-059-6528-416. Dostupné z: http://aleph.utb.cz/F/?func=service&doc_library=UTB01&doc_number=000035100&line_number=0002&func_code=WEB-BRIEF&service_type=MEDIA
- [8] Visual Studio Home | Microsoft Visual Studio. [online]. [cit. 2012-04-14]. Dostupné z: <http://www.microsoft.com/visualstudio/>
- [9] Windows Communication Foundation [online]. [cit. 2012-04-14]. Dostupné z: <http://msdn.microsoft.com/en-us/library/dd456779.aspx>

- [10] SNELL, James. *Programming Web services with SOAP* [online]. Beijing: O'Reilly, 2002, 244 s. [cit. 2012-04-15]. ISBN 05-960-0095-2. Dostupné z: http://aleph.utb.cz/F/?func=service&doc_library=UTB01&doc_number=000034712&line_number=0002&func_code=WEB-BRIEF&service_type=MEDIA
- [11] MACDONALD, Matthew. *ASP.NET 2.0 a C Sharp : tvorba dynamických stránek profesionálně*. Brno: Zoner press, 2006. 1376 s. ISBN 80-86815-38-2.
- [12] AITCHISON, Alastair a Adam MACHANIC. *Expert SQL server 2008 development* [online]. New York: Apress, 2009, 433 s. [cit. 2012-04-17]. ISBN 978-143-0272-120. Dostupné z: <http://www.springerlink.com/content/u86n18/?p=b72a5606fba74f39970535542c9f2655&pi=64>
- [13] Co jsou digitální certifikáty? | Bcom Certifikáty [online]. [cit. 2012-04-19]. Dostupné z: <http://www.certifikaty.com/certifikaty/co-jsou-digitalni-certifikaty>
- [14] Adaptic - Co je CRM [online]. [cit. 2012-04-19]. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/crm/>
- [15] Nakupujeme použitý iPhone. Nenechte se napálit!!! | Apple Fans.cz [online]. [cit. 2012-05-02]. Dostupné z: <http://www.applefans.cz/clanky/nakupujeme-pouzity-iphone-nenechte-se-napalit!!!-64/>
- [16] Nový Apple iPhone 3G S: rychlý a s řadou inovací - Mobilizujeme.cz [online]. [cit. 2012-05-02]. Dostupné z: <http://mobilizujeme.cz/clanky/novy-apple-iphone-3g-s-rychly-a-s-radou-inovaci/>
- [17] Apple iPhone 4S pictures, official photos [online]. [cit. 2012-05-02]. Dostupné z: http://www.gsmarena.com/apple_iphone_4s-pictures-4212.php
- [18] What is Siri? | PCMag.com [online]. [cit. 2012-05-02]. Dostupné z: <http://www.pcmag.com/article2/0,2817,2394787,00.asp>
- [19] Apple geeft App Store Games-sectie op de iPhone nieuw uiterlijk [Update] - iPhoneclub.nl [online]. [cit. 2012-05-02]. Dostupné z: <http://www.iphoneclub.nl/90407/apple-geeft-app-store-games-sectie-op-de-iphone-nieuw-uiteerlijk/>

- [20] Overview of an iPhone application | you said it and you can [online]. [cit. 2012-05-02]. Dostupné z: <http://yousaiditnow.blogspot.com/2010/11/overview-of-iphone-application.html>
- [21] Cool-O-Rama » Xcode 4 unifies design and code view, available to all for \$4.99 [online]. [cit. 2012-05-02]. Dostupné z: <http://www.coolorama.com/ars-technica/xcode-4-unifies-design-and-code-view-available-to-all-for-4-99/>
- [22] Novedades del Framework 4.0 · un espacio para net [online]. [cit. 2012-05-02]. Dostupné z: <http://unespacioparanet.appfuture.net/?p=152>
- [23] Microsoft .NET Framework Wiki [online]. [cit. 2012-05-02]. Dostupné z: <http://microsoft-net-framework4.software.informer.com/wiki/>
- [24] Six Things That Will Surprise You About .NET 4.0 | Onsite Training | Online Training | Public Course - DevelopMentor [online]. [cit. 2012-05-02]. Dostupné z: <http://www.develop.com/6thingsdotnet4>
- [25] IIS 7 s FastCGI a PHP 5 - ideální kombinace pro Windows - Webdesign, CMS - Webdesign - Radek Hulán - oficiální web - MyEgo.cz [online]. [cit. 2012-05-02]. Dostupné z: <http://myego.cz/item/iis-7-s-fastcgi-a-php-5-idealni-kombinace-pro-windows/category/webdesign-cms/group/webdesign>
- [26] Installing SSMS for SQLEXPRESS 2008 - Paul Kimmel's Blog [online]. [cit. 2012-05-02]. Dostupné z: <http://community.devexpress.com/blogs/paulk/archive/2009/11/25/installing-ssms-for-sqlexpress-2008.aspx>
- [27] Resources for Apple Developers - Apple Developer [online]. [cit. 2012-05-14]. Dostupné z: <https://developer.apple.com/resources/>
- [28] Getting an Apple Developer Certificate | The AppCode Blog [online]. [cit. 2012-05-14]. Dostupné z: <http://www.theappcodeblog.com/2011/04/28/getting-an-apple-developer-certificate/>
- [29] Making Your App Better Before the App Store - App Savvy - OFPS - O'Reilly Media [online]. [cit. 2012-05-14]. Dostupné z: <http://ofps.oreilly.com/titles/9781449389765/ch06.html>

- [30] iPhone Push Notification using Ruby « Spritle's Blog [online]. [cit. 2012-05-14]. Dostupné z: <http://www.spritle.com/blogs/2012/03/21/iphone-push-notification-using-ruby/>
- [31] Ad-Hoc App Distribution with XCode 4 « Diary of a Code Monkey [online]. [cit. 2012-05-15]. Dostupné z: <http://diaryofacodemonkey.ruprect.com/2011/03/18/ad-hoc-app-distribution-with-xcode-4/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ADO	ActiveX Data Objects.
API	Application Programming Interface.
ARC	Automatic Reference Counting.
ASP	Active Server Pages.
CIL	Common Intermediate Language.
CLI	Common Language Infrastructure.
CLR	Common Language Runtime.
CRM	Customer Relationship Management.
CSS	Cascading Style Sheets.
DML	Data Manipulation Language.
FTP	File Transfer Protocol.
FTPS	File Transfer Protocol Secure.
GCC	GNU Compiler Collection.
GNU	Projekt zaměřený na svobodný software, inspirovaný operačními systémy unixového typu.
GPS	Global Positioning System.
GSM	Globální Systém pro Mobilní komunikaci.
GUI	Graphical User Interface.
HSDPA	High-Speed Downlink Packet Access.
HTML	HyperText Markup Language.
HTTP	HyperText Transfer Protocol.
HTTPS	HyperText Transfer Protocol Secure.
IČO	Identifikační číslo osoby.
IDE	Integrated Development Environment.

IIS	Internet Information Services.
IPS	In-Plane Switching.
JIT	Just-In-Time.
LED	Light-Emitting Diode.
LINQ	Language Integrated Query.
MVC	Model-View-Controller.
NNTP	Network News Transfer Protocol.
OLAP	Online Analytical Processing.
RPC	Remote Procedure Call.
SDK	Software Development Kit.
SHA	Secure Hash Algorithm.
SMS	Short Message Service.
SMTP	Simple Mail Transfer Protocol.
SOAP	Simple Object Access Protocol.
SQL	Structured Query Language.
SSL	Secure Sockets Layer.
T-SQL	Transact Structured Query Language.
TFT LCD	Thin film transistor liquid crystal display.
UML	Unified Modeling Language.
UMTS	Universal Mobile Telecommunication System.
WCF	Windows Communication Foundation.
WPF	Windows Presentation Foundation.
WF	Windows Workflow Foundation.
WS	Web Service.
WSDL	Web Services Description Language.

WWDC Worldwide Developers Conference.

XAML Extensible Application Markup Language.

XML Extensible Markup Language.

SEZNAM OBRÁZKŮ

Obrázek 1: iPhone 2G [15]	12
Obrázek 2: iPhone 3G(S) [16]	13
Obrázek 3: iPhone 4(S) [17]	14
Obrázek 4: Aplikace Siri [18]	16
Obrázek 5: App Store [19]	16
Obrázek 6: Architektura MVC [20]	19
Obrázek 7: Xcode [21]	19
Obrázek 8: iPhone simulátor v nízkém a vysokém rozlišení a iPad simulátor	21
Obrázek 9: iTunes Connect	22
Obrázek 10: Struktura .NET Framework [22]	23
Obrázek 11: Schéma CLI [23]	25
Obrázek 12: LINQ dotaz ve stylu SQL	27
Obrázek 13: LINQ dotaz pomocí Lambda výrazů	27
Obrázek 14: WPF Designer [24]	29
Obrázek 15: Internetová Informační Služba [25]	32
Obrázek 16: Microsoft SQL Server Management Studio [26]	36
Obrázek 17: Struktura relační databáze	40
Obrázek 18: Struktura tabulky AuditTrail	41
Obrázek 19: Struktura tabulky Resources	41
Obrázek 20: Struktura tabulky Users	42
Obrázek 21: Struktura tabulky Content	43
Obrázek 22: Struktura tabulky Parties	43
Obrázek 23: Struktura tabulky PartyAddresses	44
Obrázek 24: Struktura tabulky PartyBankAccounts	44
Obrázek 25: Struktura tabulky PartyContacts	45
Obrázek 26: Struktura tabulky Clients	45
Obrázek 27: Struktura tabulky Contracts	46
Obrázek 28: Struktura tabulky Files	47
Obrázek 29: Struktura tabulky Relations	47
Obrázek 30: Struktura tabulky Vendors	48
Obrázek 31: Struktura tabulky Tickets	49

Obrázek 32: SOAP zpráva.....	54
Obrázek 33: Web.config.....	56
Obrázek 34: Hlavička SOAP zprávy se zabezpečením	58
Obrázek 35: Storyboard aplikace.....	59
Obrázek 36: Potvrzení odhlášení uživatele.....	60
Obrázek 37: Přihlašovací obrazovka	61
Obrázek 38: Obrazovka přehledu	62
Obrázek 39: Obrazovka provozních informací.....	62
Obrázek 40: Obrazovka zpráv	63
Obrázek 41: Obrazovka detailu zprávy.....	63
Obrázek 42: Obrazovka CRM informací.....	64
Obrázek 43: Obrazovka prodejců	65
Obrázek 44: Části obrazovky detailu prodejce	65
Obrázek 45: Část obrazovky můj profil.....	66
Obrázek 46: Obrazovka klientů se zobrazenou klávesnicí	66
Obrázek 47: Části obrazovky detailu klienta	67
Obrázek 48: Anglická jazyková verze aplikace	68
Obrázek 49: Ukázky nastavení sestavení aplikace	69
Obrázek 50: Digitální prostředky potřebné k vývoji [27].....	70
Obrázek 51: Sekce Provisioning Portalu pro vytváření vývojářského certifikátu [28]	71
Obrázek 52: Sekce Provisioning Portalu pro přidávání zařízení [29]	71
Obrázek 53: Vytváření App ID [30]	72
Obrázek 54: Vytváření Development Provisioning Profile [30]	72
Obrázek 55: Xcode se spuštěným debuggováním na zařízení iPhone [27]	73
Obrázek 56: Vytváření Distribution Provisioning Profile [29].....	74
Obrázek 57: Vytváření profilu aplikace v iTunes Connect.....	75
Obrázek 58: Organizer se sestavením aplikace [31].....	76
Obrázek 59: Průvodce vystavením sestavení aplikace na App Store [31].....	77

SEZNAM TABULEK

Tabulka 1: Procedury relační databáze	51
Tabulka 2: Veřejně dostupné funkce webové služby.....	53