

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INTERNETOVÁ APLIKACE NA MOBILNÍM ZAŘÍZENÍ IPHONE

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

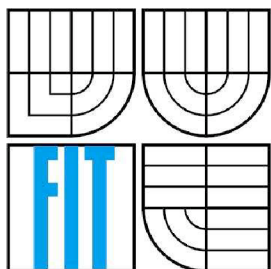
AUTHOR

ONDŘEJ FABIÁN

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INTERNETOVÁ APLIKACE NA MOBILNÍM ZAŘÍZENÍ IPHONE

INTERNET APPLICATION ON AN IPHONE MOBILE EQUIPMENT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ FABIÁN

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VLADIMÍR BARTÍK, Ph.D.

BRNO 2009

Zadání bakalářské práce

Řešitel: **Fabián Ondřej**

Obor: Informační technologie

Téma: **Internetová aplikace na mobilním zařízení iPhone**

Kategorie: Databáze

Pokyny:

1. Seznamte se s principy tvorby www aplikací na mobilním zařízení iPhone.
2. Analyzujte požadavky kladené na internetový blog pro zařízení iPhone. Data zachycená zařízením iPhone budou odesílána na server a na serveru z nich bude vygenerován HTML kód pro výslednou stránku.
3. Na základě požadavků navrhnete koncepci výsledné aplikace.
4. Navrženou aplikaci implementujte a otestujte.
5. Zhodnoťte dosažené výsledky, srovnajte s existujícími produkty a diskutujte další možnosti pokračování v tomto projektu.

Literatura:

- iPhone Dev Center: <http://developer.apple.com/iphone/>

Při obhajobě semestrální části projektu je požadováno:

- Body 1-3.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Bartík Vladimír, Ing., Ph.D., UIFS FIT VUT**

Datum zadání: 1. listopadu 2008

Datum odevzdání: 20. května 2009

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav informačních systémů
602 00 Brno, Božetěchova 2



doc. Dr. Ing. Dušan Kolář
vedoucí ústavu

Abstrakt

Tato práce se zabývá tvorbou internetové aplikace typu blog na mobilním zařízení iPhone s podporou serverového pozadí. Obsahem teoretické části je seznámení jak se zařízením samotným, tak i s principy a postupy tvorby programů pro operační systém iPhone OS. Praktická část pak popisuje návrh konkrétní aplikace, implementaci jednotlivých částí a komunikaci mezi nimi. Uvedeny jsou veškeré postupy a metody použité při vývoji takové aplikace.

Abstract

This thesis concerns the development of an internet blog application for Apple iPhone with server background. The theoretical section contains an introduction to both the device itself and the principles and procedures of the iPhone OS's program development. The practical segment describes the design of the actual application, the implementation of individual parts and the communication between them. Also mentioned are the techniques and methods applied during the application development.

Klíčová slova

Apple iPhone, iPhone aplikace, blog, uživatelské rozhraní, iPhone SDK, databáze, BSD sockets, HTML

Keywords

Apple iPhone, iPhone application, blog, user interface, iPhone SDK, database, BSD sockets, HTML

Citace

Fabián Ondřej: Internetová aplikace na mobilním zařízení iPhone, bakalářská práce, Brno, FIT VUT v Brně, 2009

Internetová aplikace na mobilním zařízení iPhone

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Vladimíra Bartíka, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ondřej Fabián
19.5.2009

Poděkování

Děkuji svému vedoucímu Ing. Vladimírovi Bartíkovi, Ph.D. za odborné vedení mé práce. Dále děkuji společnostem ALFA-HELICOPTER, s.r.o. a Golden Hour Data Systems, Inc., které mi poskytly podněty a motivaci pro tvorbu mé práce.

© Ondřej Fabián, 2009

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

Obsah

Obsah.....	1
1 Úvod.....	2
2 Historie a technologie.....	3
2.1 iPhone.....	3
2.2 Vývoj a distribuce iPhone aplikace.....	4
2.3 Alternativní přístroje.....	4
3 Teorie a potřebné znalosti.....	5
3.1 Vývojové prostředí iPhone SDK.....	5
3.2 Struktura a stavba iPhone aplikace.....	5
3.3 Síťová komunikace.....	6
4 Blog.....	8
5 Koncepce.....	9
6 Implementace.....	11
6.1 Mobilní část – iPhone aplikace.....	11
6.2 Serverové pozadí.....	16
6.3 Síťová komunikace mezi serverem a klientem.....	19
7 Možnosti dalšího rozšíření aplikace.....	21
8 Srovnání s existujícími produkty.....	23
9 Závěr.....	24

1 Úvod

Tato práce se zabývá vývojem jednoduché mobilní aplikace typu blog na zařízení iPhone se serverovým pozadím a textovým výstupem v podobě webové stránky. Nejprve bude stručně popsán přístroj, který je pro tuto práci použit, tedy mobilní telefon iPhone od společnosti Apple. Zmíněny budou jeho parametry a schopnosti, zdůrazníme zejména ty, které jsou využity pro účely této práce. Využití mobilní zařízení srovnáme s konkurenčními prostředky a uvedeme důvody zvolení právě mobilního telefonu iPhone. Dále se zmíníme o úskalích spojených s distribucí takové aplikace a stručně naznačíme i politiku firmy Apple a její postoj k vývojářům na jejich iPhone. Poté budou řešeny obecně problémy tvorby aplikace na tomto zařízení v dostupném vývojovém prostředí a jednotlivé stavební prvky použitelné pro vývoj aplikace. Další podkapitola se zaměří na síťovou komunikaci, a to jak na možnosti připojení mobilního zařízení k síti, tak na nejnižší úroveň spojení se serverem. Druhá polovina práce se bude soustředit na implementování aplikace dle zadání. Zmíníme se o návrhu celé aplikace, rozboru dílčích složek a požadavcích na ně kladených. To znamená: struktura mobilní aplikace, její jednotlivé součásti, uživatelské rozhraní a komunikační protokol mezi aplikací samotnou a serverovým programem, který bude taktéž podrobně rozebrán. Kapitola o serverovém pozadí se zaměří zejména na způsob zachování perzistence dat a generování výstupu. Na úplný závěr budou zhodnotíme dosažené výsledky a aplikace bude porovnána se současnými produkty na trhu. Rozebereme i možnosti pokračování vývoje a rozšíření aplikace.

2 Historie a technologie

Nejprve zde popíšeme přístroj, pro který je aplikace určena, tedy iPhone. Zmíněna bude i jeho stručná historie a postoj společnosti Apple vzhledem k nezávislým vývojářům. Objasněny jsou také důvody volby právě tohoto přístroje.

2.1 iPhone

Mobilní telefon iPhone byl představen světu 29. července 2007 v USA společností Apple. Jedná se o revoluční zařízení typu smartphone fungující na unixovém operačním systému iPhone OS, který je derivací systému MAC OS X, používaného na počítačích Macintosh. Současná verze systému je iPhone OS 2.2.1, popř. iPhone 3.0 beta.

Z hardwarového vybavení přístroje lze zmínit např. procesor architektury ARM pracující na frekvenci 620 MHz, dále 128 MB operační paměti a 8–16 GB interní paměti, 2 megapixelový fotoaparát a GPS přijímač. Z bezdrátových technologií, které jsou nepostradatelné pro tuto práci, je nutné uvést integrovaný Wi-Fi modul, mobilní standardy druhé generace GPRS a EDGE a od 11. července 2008, kdy byla vydána nová řada tohoto zařízení nazvaná iPhone 3G, přístroj podporuje také standardy třetí generace UMTS a HSDPA, která umožňuje přenos dat rychlostí až 14,4 MB/s. Pomocí těchto technologií je možné zabezpečit rychlý a spolehlivý datový přenos i pro vysoce náročné aplikace. Hlavní předností tohoto zařízení je však uživatelské rozhraní, které je přístupné pomocí dotykového displeje o rozměrech 320 na 480 pixelů schopného zobrazit 262144 barev. Novinkou pak byla možnost telefon ovládat pomocí tzv. *multi-touch* událostí, čili pomocí více dotyků v jediný okamžik. Další inovací, kterou společnost Apple představila jako jedna z prvních, je akcelerometr nebo měřič zrychlení. Pomocí něj je schopný iPhone určit svoji polohu vůči zemi a případně upravit uživatelské rozhraní na šířku (*landscape*) nebo na výšku (*portrait*). Akcelerometr může být však ovládacím prvkem spouštějící jakoukoliv událost, nejen úpravu uživatelského rozhraní. Text se do zařízení vkládá pomocí virtuální klávesnice zobrazené podle potřeby nejčastěji v dolní části displeje. iPhone v sobě také obsahuje podporu OpenGL a 3D grafiky, kterou zabezpečuje systém oken a grafiky zvaný *Quartz*.

Z výše uvedeného stručného výčtu některých vlastností je patrné, že iPhone poskytuje široké možnosti ve smyslu tvorby uživatelského rozhraní a komunikace. V této oblasti je také umístěné těžiště celé práce.



Obrázek 2.1 - iPhone 3G, Zdroj: [2]

2.2 Vývoj a distribuce iPhone aplikace

Téměř rok od uvedení iPhone na trh nebylo teoreticky možné pro třetí strany vyvíjet své aplikace. Prakticky to možné bylo, ne však legální cestou bez porušení autorských práv. Až 6. března 2008 uvolnila společnost Apple vývojové prostředí iPhone SDK (*software development kit*), které umožňovalo tvorbu aplikací vývojářům třetích stran. iPhone SDK v sobě obsahuje i prvky Interface Builder a iPhone Simulator. Vývojové prostředí je k dispozici zdarma.

Navzdory tomu, že je dnes možné vyvíjet aplikace dle libosti, s distribucí a testováním je tomu jinak. Distribuce probíhá přes tzv. App Store, což je v podstatě internetový obchod s aplikacemi. Aby mohl vývojář prodávat, šířit anebo jen testovat své produkty, musí se však ucházet o zpoplatněný developerský program u společnosti Apple a i pak je omezen právem této společnosti na 30% jeho výtěžku a možností distribuci aplikace kdykoliv přerušit. Bez tohoto developerského programu rovněž není možné instalovat vyvíjenou aplikaci na samotné zařízení, ale pouze do aplikace iPhone Simulator.

Přes všechna omezení je nutné říci, že se tato oblast rozvíjí obrovským tempem. Důkazem může být fakt, že bylo vytvořeno asi 35000 oficiálních aplikací a proběhlo přes miliardu nákupů od zprovoznění obchodu App Store.

2.3 Alternativní přístroje

Mobilních zařízení podobných iPhone je v současné době na trhu několik. Většina z nich však vznikla až po iPhone. Mezi tato zařízení patří např. i900 Omnia od společnosti Samsung nebo Touch Diamond2 a Magic od HTC. Prvky, které tato zařízení spojují, jsou zejména rozměrný dotykový displej pokrývající většinu plochy telefonu, operační systém, akcelerometr atd.

Omnia i800 od společnosti Samsung má např. displej o rozměrech 240 na 400 pixelů, 5Mpix fotoaparát a pracuje s operačním systémem Windows Mobile 6.1 od společnosti Microsoft. Touch Diamond2 od HTC disponuje displejem s 480 na 800 pixely, procesorem Qualcomm MSM7200A běžícím na frekvenci 528MHz a (stejně jako Omnia i900) 5Mpix fotoaparátem a Windows Mobile 6.1. HTC Magic má o něco horší parametry než jeho předchůdce, podporuje však unixový systém Android vyvinutým společností Google.

Důvodem volby iPhone nebyl ani výkon ani parametry periférií, které jsou konkurencí překonány, ale spíše dojem zařízení na uživatele jako celku. Celkový design, plynulost, čistota a ladnost celého uživatelského rozhraní byly výhradně subjektivními důvody vedoucími k volbě této platformy pro vývoj aplikace. Dále by se dala zmínit obecně známá stabilita unixových systémů, která také směřovala k použití iPhone OS před Windows Mobile.

3 Teorie a potřebné znalosti

Následující podkapitoly seznamují s prostředky použitými při vývoji a pojmy nutnými pro implementaci aplikace. Poté se podrobněji rozeberou postupy tvorby programu na tomto zařízení, a to včetně nejdůležitějších objektů použitelných při stavbě aplikace a prvků uživatelského rozhraní.

3.1 Vývojové prostředí iPhone SDK

Jak již bylo zmíněno, vývojové prostředí iPhone SDK obsahuje kromě součástí, které se dají u podobného balíku určeného pro danou platformu očekávat jako podrobná dokumentace a API také dvě důležité aplikace iPhone Simulator a Interface Builder. Nejen z důvodu nemožnosti vyzkoušet vyvíjenou aplikaci přímo na samotném zařízení, jak bylo uvedeno o kapitolu dříve, je tato součást nezbytnou složkou pro každého začínajícího i pokročilého vývojáře, nemluvě o ladění apod. Není tomu tak u nástroje zvaného Interface Builder. Tento program slouží k jednoduchému tvoření uživatelských rozhraní metodou *drag & drop* a navazování spojení mezi vstupy aplikace a objekty zpracovávajícími jejich události. I když by Interface Builder mohl pomoci k pochopení struktury iPhone aplikace, která bude uvedena níže, pro lehce znalého vývojáře už bude spíše přítěží. Tento nástroj totiž negeneruje kód, ale zapouzdřuje ho do samostatných souborů, odkud se musí k hlavnímu programu znovu ručně navazovat. Jako lepší řešení se tedy zdá vytvářet si uživatelské rozhraní ručně, programaticky.



Obrázek 3.1 - iPhone SDK

3.2 Struktura a stavba iPhone aplikace

Programovací jazyk pro tvorbu iPhone aplikací je Objective-C, který vychází z jazyků C a Smalltalk. Syntaxe pochází z C a po Smalltalku Obj-C dědí zejména systém zasílání zpráv. Vzhledem k jazyku C je plně zpětně kompatibilní. Objective-C je tedy objektový jazyk používaný primárně pro tvorbu aplikací na systémy MAC OS X a iPhone OS.

Následuje výčet vybraných tříd z programové dokumentace a jejich vlastnosti nezbytné pro tvorbu iPhone aplikace:

UIView je abstraktní třída, ze které dědí konkrétní třídy uživatelského rozhraní (např. textová pole, tlačítka, obrázky). Instance **UIView** se také používá jako plátno či kontejner, do kterého se vykreslují jednotlivé prvky rozhraní.

UIViewController poskytuje základní správu instancí objektů dědicích ze třídy **UIView**. Používá se zejména pro ruční vykreslování prvků uživatelského rozhraní na obrazovku nebo např. pro obsluhu rotace zařízení tzn. překreslování prvků do vodorovného zobrazení a zpět.

UINavigationController je podtřídou **UIViewController**. Tato třída spravuje zásobník instancí tříd **UIViewController** a vytváří tak hierarchii jednotlivých obrazovek. Objekt, který je na vrcholu zásobníku, je zároveň zobrazen na displeji přístroje.

UIDelegate definuje protokol, pomocí kterého zasílají prvky uživatelského rozhraní zprávy o interakci s uživatelem nebo jiných událostech. Delegátem se může stát téměř libovolný objekt, nejčastěji to však bývá **UIViewController**, který se stará o vykreslování a chod vždy jedné celé obrazovky.

Dále jsou tu prvky uživatelského rozhraní, které není potřeba rozebírat. Detailní popis je k nalezení v programové dokumentaci přiložené k výše zmíněnému vývojovému prostředí. Mezi tyto třídy patří např. **UIButton**, **UITextField**, **UITextView**, **UIAlertView**, **UIPageControl** atd.

Zmíněné třídy jsou pravděpodobně neodmyslitelné součástí většiny netriviálních aplikací. Pochopení jejich účelů a návazností mezi nimi je fundamentální pro vývoj grafického rozhraní iPhone aplikace.

```
- (void)loadView {
    UIView *containerView = [[UIView alloc] initWithFrame:[UIScreen mainScreen] applicationFrame];
    containerView.backgroundColor = [UIColor blackColor];
    self.view = containerView;
    [containerView release];

    attachButton = [self createAttachButton];
    bgimageView = [self createBgimageView];

    [self.view addSubview:bgimageView];
    [self.view addSubview:iView];
    [self.view addSubview:attachButton];
}
```

Obrázek 3.2 - Ukázka kódu - metoda loadView

V přiložené ukázce je metoda *loadView* třídy *UIViewController*. V této metodě je vytvořeno *UIView* a připojeno jako kontejner ke svému *UIViewController*. Do tohoto kontejneru jsou pak připojeny další prvky uživatelského rozhraní.

3.3 Síťová komunikace

Jak již bylo zmíněno v první kapitole, iPhone 3G využívá trvalé připojení k internetu pomocí podporovaných technologií GPRS, EDGE, UMTS, HSDPA a Wi-Fi. Tato podkapitola se však zaměří na síťovou komunikaci spíše na úrovni komunikačních protokolů a práce s datovými toky.

Při tvorbě síťové iPhone aplikace lze vybírat z několika úrovní síťové komunikace a zvolit ten způsob, který se pro konkrétní potřebu nejvíce hodí. Hladiny síťové komunikace jsou následující:

BSD Sockets - klasické standardní API jako u každé jiné aplikace psané v jazyce C.

CFNetwork nabízí nízkoúrovňové řešení, které umožňuje detailní kontrolu nad TCP/IP vrstvou, jelikož teoreticky i fyzicky vychází z BSD sockets. Nabízí však některé rozšiřující funkce a poskytuje objekty pro zjednodušení komunikace např. s FTP a HTTP servery.

NSURL je další nástavba nad CFNetwork. NSURL poskytuje způsob, jak zacházet s URL adresami a zdroji, na které odkazují. Standardy ohledně URL uvedené v RFC 1808, 1738 a 2732 jsou touto třídou dodržovány. NSURL je tedy třídou určenou pro komunikaci se servery podle standardních internetových protokolů. Tato třída je poměrně vysokoúrovňová, čili implementuje většinu detailů síťové komunikace sama o sobě.

Web Kit je poslední a nejvyšší nástavbou, která umožňuje např. přímo zobrazovat obsah webových stránek v oknech.

Volba úrovně síťové komunikace tedy záleží na konkrétních potřebách každé aplikace, resp. na tom, do jaké hladiny chce mít tvůrce nad síťovým spojením kontrolu a co chce nechat v režii poskytnutých knihoven.

4 Blog

Jelikož je tato práce technického charakteru, je následující teoretická informace velice stručná a podává pouze základní vědomosti nezbytné k řešení zadané úlohy.

Slovo *blog* vzniká upravením anglického slovního spojení *web log*, což lze přeložit jako internetový deník. Vznik tohoto slovního spojení se připisuje Američanovi Jornu Bargerovi, který jej použil poprvé v roce 1997. O dva roky později jako první v tomto sousloví posunul mezeru o jedno písmeno dopředu Peter Merholz a dal vzniku sousloví *we blog*. Slovo *blog* je od té doby používáno jako podstatné jméno i sloveso (česky blogovat). Tvůrce blogu se nazývá *blogger*. Tento odstavec čerpá z [8].

Blog je tedy typ webové stránky, která zobrazuje záznamy, komentáře a postřehy jednotlivce, či skupiny lidí. Záznamy se zpravidla hromadí v inverzním chronologickém pořadí, tedy novější před staršími. Kromě samotného textového obsahu bývají většinou doplněny minimálně o datum příspěvku, ale i o další podklady jako např. obrázky, videa, odkazy aj. Podle úhlu pohledu lze blogy dělit do několika skupin.

Kritérium dělení	Příklad
Autor	Osobní, firemní.
Typ záznamu	Text, video, odkaz, malba, obraz, smíšený.
Zařízení použité k tvorbě	Mobilní zařízení (PDA), počítač, kamera.
Žánr	Politika, cestování, domácnost, móda, vzdělávání, hudba, umění...

Tabulka 4.1 - Typy blogů

Další specifickou skupinou blogů jsou takzvané mikroblogy. Jedná se o typy blogů, kde se délka příspěvků pohybuje zhruba kolem 180 znaků. Jsou to tedy velice krátké vzkazy, kterými autor většinou odpovídá na otázku "Co právě dělám?" Ze zahraničních je nejznámějším mikroblohem pravděpodobně Twitter a z tuzemských Mikroblog.cz.

Závěrem lze podotknout, že tvorba a údržba blogů je v současné době poměrně oblíbená a rozvíjející se činnost. Komunita lidí věnujících se této aktivitě se nazývá blogosféra.

5 Koncepce

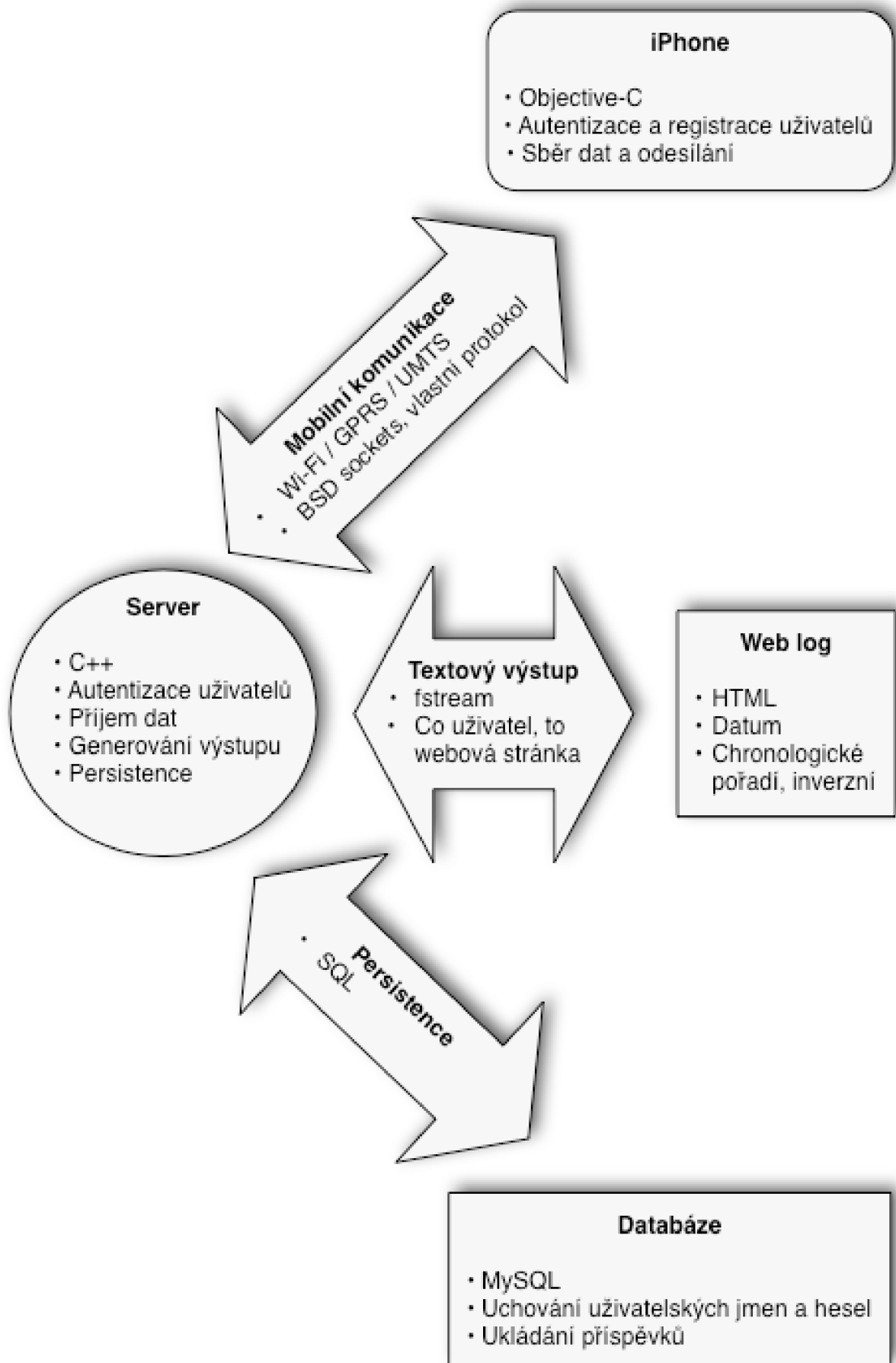
Pro splnění zadání bylo nutné analyzovat několik problémů. Jako upřesnění cíle je možné aplikovat předchozí kapitolu na konkrétní případ. Úkolem bylo vytvořit osobní textový blog prakticky libovolného žánru generovaný na mobilním zařízení.

Jelikož se jedná o aplikaci typu klient-server, bude nutné vytvořit dvě samostatné části programu. Způsob implementace klienta – iPhone aplikace – byl předem daný. Vývojové prostředí iPhone SDK představuje zřejmě jedinou oficiální možnost. Implementačním jazykem klienta je tedy Objective-C. U serverové části aplikace tomu bylo jinak. Zde bylo možné vybrat téměř libovolný způsob implementace. Byl zvolen jazyk C++. Ze zásadnějších problémů byl dále u této části aplikace řešen způsob zachování persistence dat a výstupu – výsledné HTML stránky. V další fázi bylo nutné vyřešit způsob komunikace mezi klientem a serverem. V podkapitole 3.3 bylo zmíněno několik možností, kterými iPhone OS API disponuje. Zbývalo se tedy zamyslet, zda využít nějaký ze stávajících internetových protokolů, či nikoliv. Jelikož mobilní aplikace nemusí mít ideální podmínky připojení k síti, požadavky pro síťovou komunikaci budou hlavně jednoduchost, minimální nároky na objem přenesených dat a obsluha na nízké úrovni.

Požadavků na klientskou část bylo několik. V první řadě musí osahovat vstupní bod, přes který se uživatel dostane jen se správnými přihlašovacími údaji. Z tohoto místa by také měla být umožněna nová registrace uživatelů. Po překonání vstupní brány a ověření přihlašovacích údajů by mělo následovat menu, ze kterého by uživatel měl mít možnost ovládat všechny implementované funkce aplikace. V první verzi to bude nejspíš pouze vytvoření nového příspěvku a případně prohlížení výsledné stránky generované serverovou částí aplikace. V budoucnu se tu mohou objevit např. odkaz na úpravu stylu výsledné stránky a jiné libovolné funkce. Obrazovka pro nové příspěvky by měla osahovat aspoň dvě textová pole pro titulek příspěvku a samotný obsah. Textové pole pro obsah příspěvku by mělo být větších rozměrů. Dále by obrazovka pro nové příspěvky měla obsahovat možnosti pro připojení obrázku ke článku. Obrázek by mělo být možné přiblížit a případně oříznout. Prohlížení výsledné webové stránky by bylo možné dvěma způsoby. Jednou možností je prohlížet stránku přímo v aplikaci pomocí třídy *UIWebView*, nebo stránku otevírat v nativním prohlížeči přístroje iPhone, kterým je Safari. Pro korektní zobrazení bude pravděpodobně lepší zvolit druhou možnost.

Serverové pozadí má především funkční význam. Musí být schopné přijímat klientské zprávy, korektně je analyzovat, provádět požadované operace a odesílat odpovídající zprávy zpět klientovi. Vše by mělo být prováděno konkurentně. Jeden požadavek klienta by neměl čekat na zpracování druhého, měly by být zpracovávány současně a paralelně. Informace o uživateli musejí být persistentní, čili musejí přetrvat ukončení a opětovné spuštění aplikace. K tomuto účelu bude pravděpodobně vhodné využít databázi, a to i vzhledem k možným budoucím rozšířením aplikace. V databázi budou pro další vývoj uchovávány i jednotlivé příspěvky uživatelů. Osobní webová stránka každého uživatele bude vytvořena při registraci a jednotlivé příspěvky budou zapisovány na začátek souboru, aby tak bylo zajištěno požadované inverzní chronologické pořadí výsledného blogu. Vzhled webové stránky bude upraven externím kaskádovým stylem.

Komunikační zprávy by neměly obsahovat redundantní data, měly by být jednoduché, přímočaré a dále by se měly dát jednoduše zpracovat. V ideálním případě by zprávy měly vycházet z určitého vzorce pro snadnou reprodukovatelnost a rozšiřitelnost.



Obrázek 5.1 - Koncept aplikace

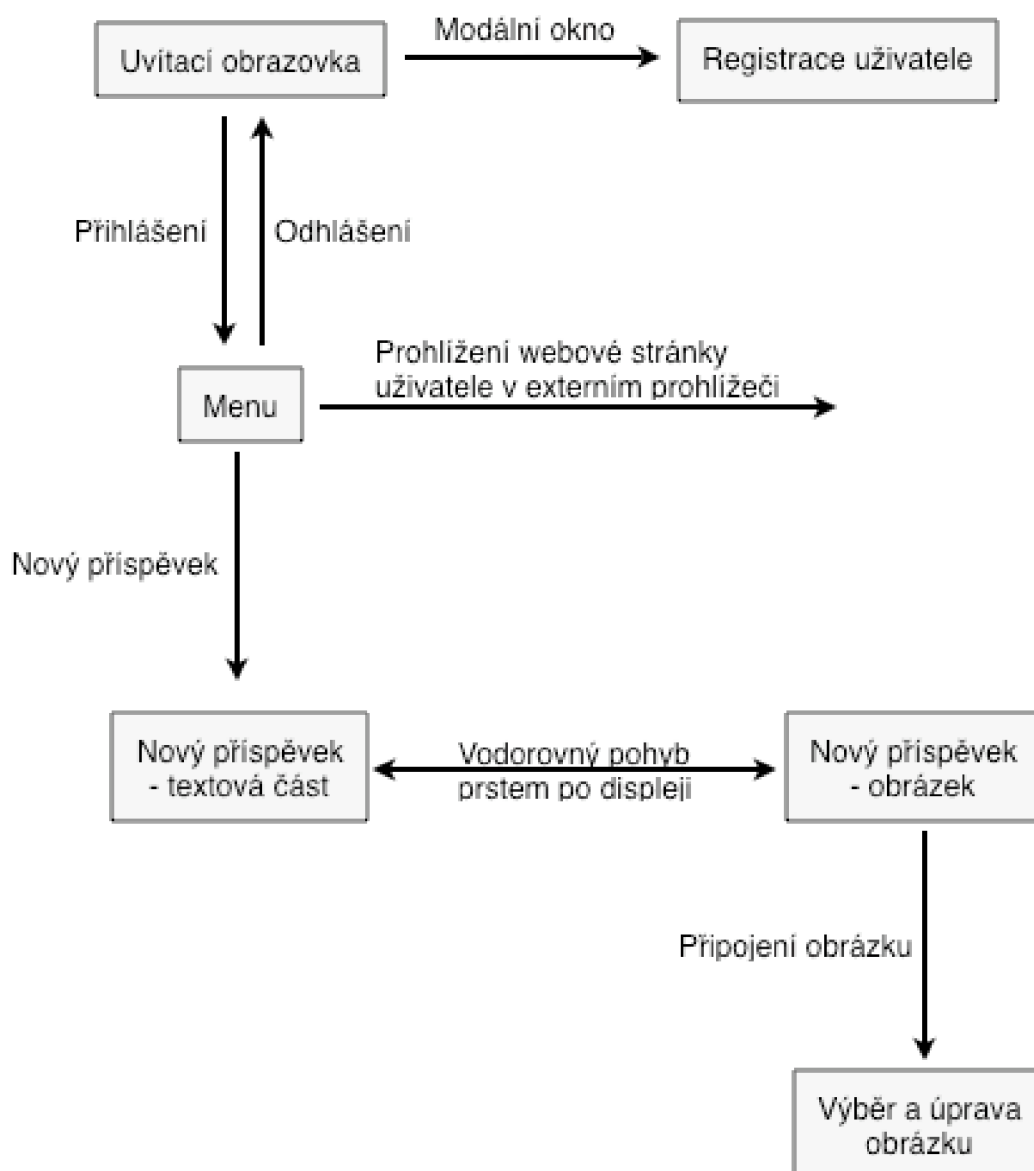
6 Implementace

Následující odstavce odpovídají na otázky týkající se implementace zadané aplikace. Budou zejména popsány jednotlivé komponenty aplikací a zprůhledněna jejich funkčnost.

6.1 Mobilní část – iPhone aplikace

Mobilní část aplikace a její uživatelské rozhraní představuje těžiště celé práce. Bylo na ní vynaloženo nejvíce úsilí, zejména pak na její teoreticko-studijní část. Důraz je kladen především na přehlednost, jednoduchost a intuitivnost obsluhy. Požadavky kladené na iPhone aplikaci byly primárně tyto:

- zachycení textových dat,
- odeslání na server,
- registrace uživatelů,
- autentizaci uživatelů,
- zachycení obrazu,
- prohlížení výsledného výstupu.



Obrázek 6.1 - Struktura iPhone aplikace

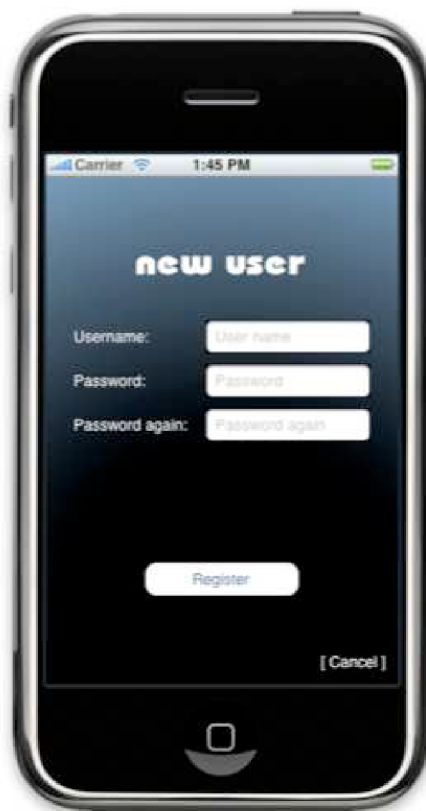
Vstupním bodem aplikace je **uvítací obrazovka**, kterou tvoří dvě textová pole a dvě tlačítka. V nejhornější části displeje je stavová lišta s údaji o síle signálu, operátorovi, čase a stavu baterie. Tato lišta je součástí každé obrazovky. Pod ní se nachází navigační lišta, která zatím na této obrazovce nemá náležité využití. Textová pole jsou označena slovy *Username* a *Password* a slouží společně s tlačítkem *Login* pro autentizaci již existujícího uživatele. Tlačítko *[Register]* je tu pak pro registraci nového uživatele.



Obrázek 6.2 - Uvítací obrazovka

Po označení textového pole, do kterého chce uživatel psát, se objevuje kurzor, klávesnice a obrazovka se posouvá směrem vzhůru, tak aby objevující se klávesnice nepřekryla žádné z textových polí. Při vkládání textu do pole *Password* zůstává pro zvýšení přesnosti zadávaného textu vždy poslední písmeno po dobu dvou vteřin neskryté. Stisknutí tlačítka *Done* na klávesnici schová klávesnici a posune obrazovku zpět do původní pozice.

Po stisknutí tlačítka [*Register*] na úvodní obrazovce vystoupí ze spodní části obrazovky modální okno pro **registraci nového uživatele**. Zde je možné korektním vyplněním všech polí a stisknutím tlačítka *Register* zaregistrovat nového uživatele, popř. stisknutím [*Cancel*] registraci zrušit.



Obrázek 6.3 - Obrazovka registrace nového uživatele

Úspěšným přihlášením se uživatel přenese do **menu**. Zde jsou v současném stavu k dispozici dvě možnosti. První se skrývá pod tlačítkem *New note*, což umožní uživateli vkládat nové příspěvky na svou osobní stránku. Druhá možnost je prohlížet tuto osobní stránku. Po stisknutí tlačítka *My page* se aplikace zavře a webová stránka s příspěvky uživatele se otevře v nativním prohlížeči přístroje iPhone Safari. Se vstupem do menu se aplikace zanořuje do hierarchie oken a na navigační liště přibývá tlačítko pro odhlášení uživatele. Na této obrazovce je také pro kontrolu zobrazeno uživatelské jméno právě přihlášeného uživatele.



Obrázek 6.4 - Obrazovka menu

Okno pro **přidávání nových příspěvků** využívá prvek uživatelského rozhraní zvaný *ScrollView*. Tento prvek umožňuje uspořádat několik stránek vedle sebe do jedné obrazovky. V tomto případě jsou použity dvě stránky, mezi kterými lze navigovat dvěma způsoby. Buď pomocí ovládacího prvku *pageControl* ve spodní části obrazovky, který zároveň indikuje aktuální stránku, anebo pomocí vodorovného pohybu prstem po displeji ve směru posunu stránky. Levá stránka obsahuje dvě textová pole. První je určené pro titulek článku a druhé větší pak pro jeho obsah. Pro větší textové pole je využita třída *UITextView*, která dědí ze třídy *UIScrollView*. Má tedy stejné vlastnosti až na výjimku, že obsah tohoto pole není zobrazován po stránkách, ale plynule navazující. V pravé stránce je umístěno tlačítko pro přiložení obrázku. Z této funkce je však implementována pouze uživatelská část. Obrázek tedy není odeslán na sever a připojen do osobní stránky uživatele. Po stisknutí tlačítka *Attach picture* se objeví dialog, ve kterém si uživatel zvolí, zda chce pořídit nový snímek pomocí integrovaného fotoaparátu, nebo připojit již pořízený snímek z galerie. Po zvolení obrázku je možné jej dále upravovat. K tomu jsou využita již dříve zmíněná tzv. *multi-touch* gesta. Obrázek je možné pohybem dvou prstů po displeji přibližovat a oddalovat a pomocí jednoho prstu také volit oblast výřezu. Po potvrzení úprav obrázku je jeho vybraný výřez umístěn pro kontrolu na pravé stránce pod tlačítkem *Attach different picture*.



Obrázek 6.5 - Zleva: Levá stránka, pravá stránka, dialog pro výběr zdroje, úprava obrázku, náhled na vybraný obrázek

Na navigační liště je stejně jako v předchozí úrovni tlačítko pro návrat o úroveň výš, tedy do menu. V případě, že uživatel vkládá text do většího textového pole pro obsah příspěvku, se na navigační liště objevuje další tlačítko pro ukončení vkládání textu a schování klávesnice. U jiných textových polí je toto tlačítko přímo na výsuvné klávesnici. U textového pole pro vkládání většího obsahu je však stejné tlačítko na klávesnici vyhrazeno pro přechod na nový řádek. Na levé stránce se nachází tlačítko *Post note*, které po stisknutí odesílá zaznamenaná data na server.

V mobilní části aplikace je ještě jedna třída, která však nemá žádný vizuální výstup. Jmenuje se *Networking* a zabezpečuje komunikaci se serverem. Třída implementuje několik metod. Každá metoda zabezpečuje jinou operaci. Všechny metody vracejí hodnotu typu *integer*, podle které lze rozhodnout o úspěšnosti provedené operace. V následujícím obrázku jsou uvedeny prototypy implementovaných metod včetně možných návratových hodnot. Aplikace pak informuje uživatele podle těchto návratových hodnot o úspěchu, neúspěchu anebo chybě prováděné operace formou *popup* upozornění.


```

/**
 * Initializes connection with server. Returns number of created BSD socket. In case of error -1 is returned.
 */
+ (int)initConnection;

/**
 * Creates new user. Returns 1 - OK, 0 - username taken or -1 error occurred.
 */
+ (int)registerNewUser:(NSString *)login:(NSString *)password;

/**
 * Checks user name and password. Returns 1 - OK, 0 - incorrect login or -1 error occurred.
 */
+ (int)checkLogin:(NSString *)login:(NSString *)password;

/**
 * Post a new note. Returns 1 - OK or -1 error occurred.
 */
+ (int)postNote:(NSString *)username:(NSString *)noteTitle:(NSString *)noteText;

```

Obrázek 6.6 - Ukázka zdrojového kódu - Metody třídy *Networking*

Metoda *initConnection* inicializuje spojení se serverem a je volána v každé z následujících metod. V případě úspěšného provedení vrátí číslo vytvořeného socketu, jinak -1.

Metoda *registerUser* posílá na server příkaz k vytvoření nového uživatele. Parametry této metody jsou uživatelské jméno a heslo registrovaného. V případě úspěchu je návratový kód 1, v případě existence uživatele se stejným jménem je kód 0 a v případě selhání spojení nebo jiné chyby je návratová hodnota -1.

Metoda *checkLogin* má za parametry stejné informace jako metoda *registerUser*. Účelem této metody je však ověřit v databázi jméno přihlašujícího se uživatele a zkontrolovat správnost zadaného hesla. Návratové hodnoty jsou rovněž obdobné jako u předcházející metody. 1 značí kladné ověření jména a hesla. 0 značí neexistující uživatelské jméno nebo chybné heslo a -1 signalizuje chybu.

Poslední ve výčtu metod je metoda s názvem *postNote*. Tato metoda informuje server o novém příspěvku daného uživatele. Parametry metody jsou tedy logicky uživatelské jméno přispívajícího, titulěk článku a text příspěvku. Metoda vrátí opět 1 (resp. 0) v případě úspěšného (resp. neúspěšného) provedení.

Jak pak vypadají jednotlivé zprávy zasílané serveru a jak se mobilní aplikace dozvídá o výsledcích provedených operací je popsáno ve třetí podkapitole této kapitoly.

6.2 Serverové pozadí

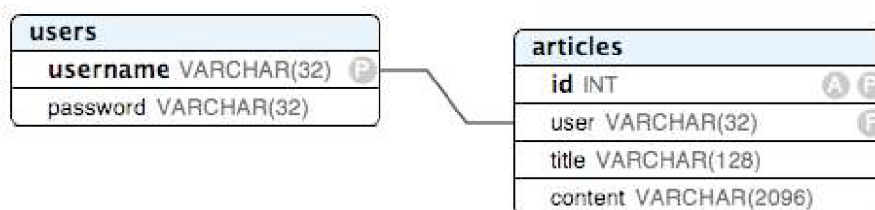
Serverová část aplikace má několik úkolů. Mezi ně patří zejména

- přijímat a obsluhovat požadavky klienta,
- informovat klienta o výsledcích operací,
- uchovávat informace o uživateli,
- ověřovat identitu uživatelů,
- archivovat příspěvky,
- generovat výslednou webovou prezentaci.

Program serveru není objektově orientovaný, mohl by tedy teoreticky splňovat normu ISO C99, využívá však některé konstrukce a knihovny jazyka C++ např. pro práci s řetězci. Komunikace s klientem je, jak již bylo řečeno, zabezpečena na té nejnižší úrovni pomocí BSD sockets. Server je konkurenční, zvládá tedy obsluhu více klientů současně. Detekci příchozích spojení zaopatrjuje rodičovský proces. Pro obsluhu každého požadavku, přijetí příkazu, vykonání jednotlivých procedur a

zaslání odpovědi je vždy vytvořen nový proces. Hlavní smyčka programu se stará o příjem požadavků klienta, které následně předává funkci *serve()* k dekodování a následné obsluze.

Zásadní otázkou bylo, jakým způsobem řešit uchovávání dat. Jasnou volbou se nakonec ukázala být databáze MySQL, a to především pro jednoduchý přístup z různých aplikací a možnosti snadného rozšíření do budoucna. Databáze pod serverovou částí aplikace obsahuje v současné době dvě tabulky. Jedna z nich uchovává informace o uživateli a druhá o článcích. Tabulka s příspěvky uživatelů má v tomto stavu aplikace čistě archivační účel. K vykonávání dotazů nad MySQL databází je využita knihovna *mysql.h*.



Obrázek 6.7 - ER diagram¹

Dalším problémem k řešení byl výstup aplikace a tvorba webové stránky. Jelikož, jak bude uvedeno v kapitole o možných rozšíření aplikace, se tato úloha serveru s největší pravděpodobností stane slepou uličkou, byla zajištěna jen základní funkcionalita této role. V prvotní úvaze o konceptu aplikace bylo počítáno s využitím DOM pro editaci výsledné HTML stránky. Bylo shledáno, že pro účely této aplikace bohatě postačí editace souboru pomocí C++ knihovny *fstream.h*.

Server v současné fázi obsluhuje několik požadavků klienta. Každý požadavek se skládá z několika podprocedur:

Funkce **vytvoření nového uživatele** dostává dva parametry, kterými jsou uživatelské jméno a heslo nového uživatele. Jako první tato funkce kontroluje databázi, zda toto jméno již neobsahuje. Jestliže ano, končí neúspěchem, tím je zabezpečena integrita databáze. Pokud databáze jméno neobsahuje, je v ní vytvořen příslušný záznam o uživateli včetně hesla. Dále je uživateli vygenerována jeho osobní webová stránka s názvem *User.html*:

```
<html>
<head>
<meta http-equiv='Content-Type' content='text/html; charset=iso-8859-2'>
<link href='style.css' rel='stylesheet' type='text/css' />
<title>User's personal log</title>
</head>
<body>
<h1>User's personal page
</h1>
</body>
</html>
```

Obrázek 6.8 - Ukázka zdrojového kódu - Webová stránka

Další nezbytnou operací je v logickém pořadí **ověření identity uživatele**. Parametry jsou u této funkce totožné jako u předchozího případu. Program se dotazuje databáze na hesla uživatele se stejným jménem, jako je předaný parametr. Výsledek dotazu pak porovnává s druhým parametrem

¹ P značí primární klíč, F je cizí klíč, A značí auto increment

funkce. V případě nenalezení záznamu o požadovaném uživateli nebo v případě negativního výsledku porovnání hesel operace končí neúspěchem.

Funkce pro **vložení nového příspěvku** má za parametry uživatelské jméno, titulek a samotný obsah článku a skládá se také z několika kroků. Nejprve je příspěvek vložen do databáze, což, jak je zmíněno výše, se prozatím děje čistě z archivačních důvodů. Na tento krok je však v plánu navázat v budoucích rozšířeních aplikace. Poté je nalezen soubor obsahující webovou stránku uživatele a za řádek obsahující značku `</h1>` jsou jednotlivé části nového příspěvku vloženy v pořadí: titulek, aktuální datum a samotný text. Tím je zabezpečeno inverzní chronologické pořadí, jak je požadováno v konceptu aplikace. U této operace není mezi parametry uživatelské heslo, neboť z klientské strany není možné vyslat požadavek na tuto operaci, aniž by se uživatel se správnými údaji do aplikace přihlásil.

```
<div class='article'>
<h2>Titulek clanku</h2>
<span class='date'>Fri May 8 22:22:27 2009
</span><p>
Zde je umisten cely dlouhy text prispevku.
</p></div>
```

Obrázek 6.9 - Ukázka zdrojového kódu – nový příspěvek

Ať už výše zmíněné funkce skončí jakýmkoliv výsledkem, je toto rozhodnutí předáno hlavní smyčce programu, která tuto zprávu odesílá zpět klientovi. Jak tyto odpovědi vypadají, je popsáno v následující podkapitole o komunikaci.



Obrázek 6.10 - Výsledná stránka zobrazená na zařízení iPhone

6.3 Síťová komunikace mezi serverem a klientem

V kapitole o implementaci bylo zmíněno již téměř vše kromě posledního článku skládanky celé aplikace, který tato podkapitola doplňuje. Jedná se o síťovou komunikaci mezi klientským a serverovým programem. Volbou úrovně komunikace se z důvodu minimálních nároků na přenosovou kapacitu a největší kontroly stala varianta nejnižší úrovně, tedy BSD sockets. Podobně tomu bylo i s komunikačním protokolem, nejjednodušší a nejpřímochařejší řešení bylo vytvoření vlastního protokolu.

Jelikož ohnisko celé aplikace nespočívá v síťové komunikaci, nýbrž v uživatelském rozhraní a vůbec v klientovi, jsou v této části aplikace implementována naprosto minimální bezpečnostní opatření. Komunikace mezi prvky aplikace je bezstavová, čili není žádným způsobem zachována informace o předchozích zprávách obdržných od klienta ani o jejich výsledcích. Toto by pro zvýšení bezpečnosti bylo potřeba zřejmě změnit. Jelikož komunikace není ani žádným způsobem šifrována, nebylo by pro potenciálního útočníka zřejmě problémem nějakou zprávu serveru podvrhnout za podmínky, že by znal komunikační protokol aplikací. Možným zlepšením zabezpečení aplikace bude věnována podkapitola v kapitole o rozšíření aplikace.

Již několikrát bylo zmíněno, že pro účely komunikace je využít původní komunikační protokol. Skutečnost použití netradičního komunikačního protokolu by se dala do jisté míry také považovat za bezpečnostní prvek. Všechny zprávy komunikačního protokolu vycházejí ze stejného vzorce, který vypadá následovně:

příkaz:parametry:oddělené:dvojtečkami[E];

Jelikož je použit nízkourovňový způsob komunikace pomocí BSD sockets a zprávy mají proměnnou délku, je za každou z nich připojena ukončovací sekvence znaků *[E];* pro snadné rozpoznání konce zprávy a tím zvýšení efektivity a spolehlivosti přenosu. Přehled zpráv zasílaných mezi klientem a serverem lze nalézt v následujících tabulkách.

Operace	Šablona	Příklad zprávy
Zprávy klienta		
Registrace uživatele	<i>register:username:password[E];</i>	register:User:mojeHeslo[E];
Ověření identity	<i>login:username:password[E];</i>	login:User:mojeHeslo[E];
Nový příspěvek	<i>newpost:username:title:content[E];</i>	newpost:Titulek clanku:Zde je umisten cely dlouhy text prispevku.[E];

Tabulka 6.2 - Komunikační protokol – klient

Odpovědi serveru pak následují tento vzorec:

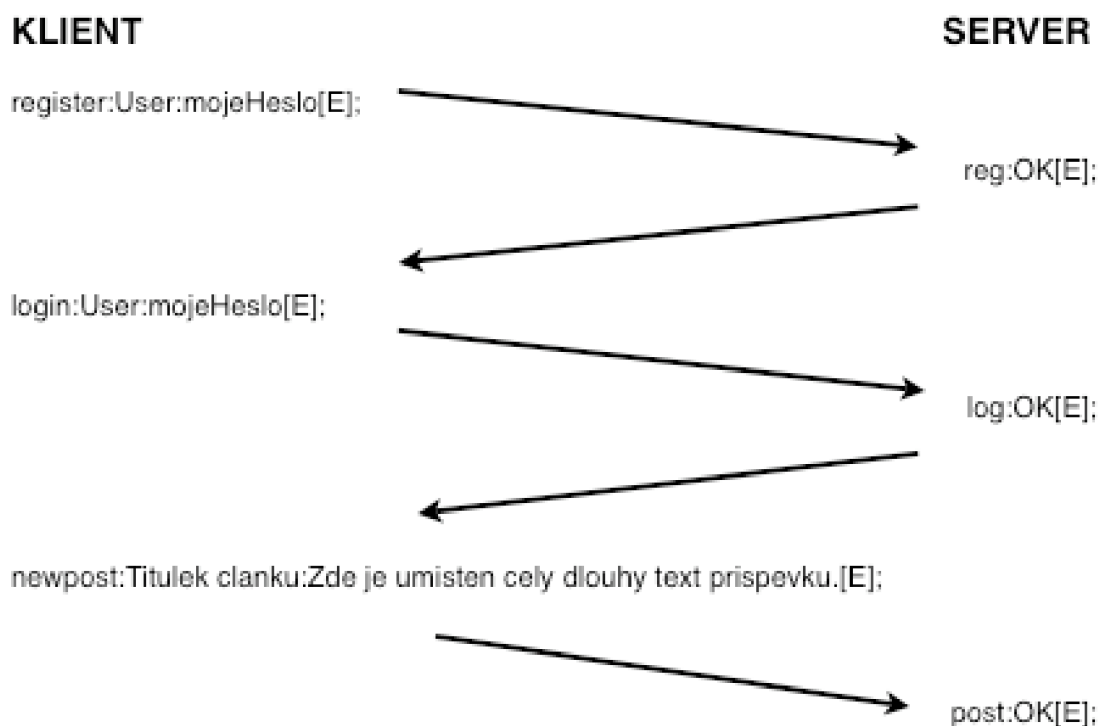
operace:výsledek[E];

Pod pojmem operace se v tomto vzorci může vyskytovat buď prováděný příkaz, nebo signalizace chyby. Výsledek pak značí úspěch nebo neúspěch prováděného příkazu, nebo specifikuje chybu.

Operace	Odpověď	Význam
Odpovědi serveru		
Registrace uživatele	reg:OK[E];	Úspěšná registrace.
	reg:NA[E];	Uživatel se stejným jménem již existuje.
	err:DB[E];	Chyba serveru, databáze.
Ověření identity	log:OK[E];	Úspěšné přihlášení.
	log:NA[E];	Chybné uživatelské jméno nebo heslo.
	err:DB[E];	Chyba serveru, databáze.
Nový příspěvek	post:OK[E];	Příspěvek úspěšně přidán.
	post:ERR[E];	Chyba serveru, zapisování do souboru.

Tabulka 6.2 - Komunikační protokol – server

Klient zprávy po přijetí dekoduje a provádí odpovídající reakce, případně ještě informuje o výsledku uživatele. Příklad komunikace by mohl vypadat následujícím způsobem.



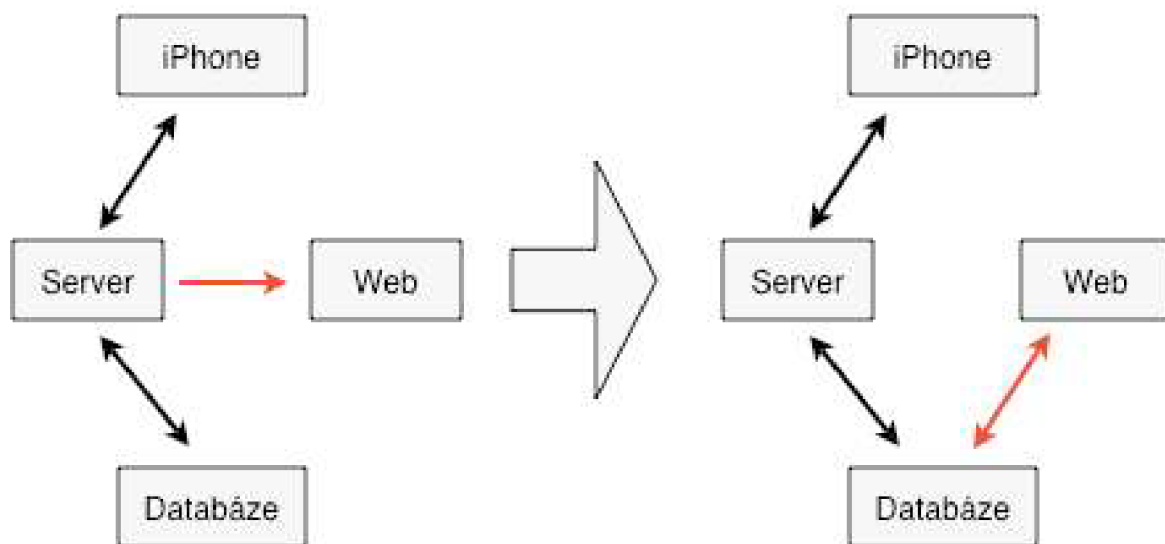
Obrázek 6.11 - Příklad komunikace klient – server

Služba v současném stavu funguje na portu 51515 s ohledem na regule organizace IANA, které dle [3] udávají tři různé rozsahy: dobře známé porty 0 až 1023, registrované 1024 až 49151 a dynamické nebo privátní 49152 až 65535. Port 51515 byl tedy náhodně vybrán ze třetí kategorie.

7 Možnosti dalšího rozšíření aplikace

Rozšíření současného stavu aplikace se nabízí hned několik druhů. Zlepšit by bylo možné bezpečnost aplikace, rozšířit její funkčnost, rozšířit přenášené spektrum dat o další typy, upravit strukturu aplikace atd.

Jako další krok při vývoji aplikace by bylo vhodné upravit strukturu aplikace následujícím způsobem a umožnit tak samostatnou správu serverové části a webového výstupu.



Obrázek 7.1 - Budoucí vývoj struktury aplikace

Této struktury by se dosáhlo zamezením generování HTML stránky ze serveru a implementováním webové prezentace prostředky PHP, Java Script popř. AJAX. To by umožnilo mnohem dynamičtější vzhled webových stránek a především také oboustrannou komunikaci s databází. Bylo by tedy možné implementovat i klientskou stranu do webových stránek a přizpůsobit tak celou aplikaci i uživatelům, kteří nejsou vybaveni mobilním zařízeními iPhone. Díky asynchronnímu dotazování AJAX by stránka mohla dosáhnout také daleko interaktivnějšího a atraktivnějšího vzhledu.

Dalším možným krokem k vyšší kvalitě by byla zřejmě zvýšená bezpečnost aplikace. Toho by šlo dosáhnout např. zavedením stavové komunikace mezi klientem a serverem. Server by si pomatoval korektního uživatele ustanovením relace a podvrhnutí zprávy útočníkem by bylo komplikovanější. Z hlediska bezpečnosti by bylo zajisté vhodné, aby aplikace šifrovala veškerou síťovou komunikaci, nebo alespoň hesla uživatelů. Ukládání hesla v databázi v otevřené podobě také zajisté není příliš dobrý aspekt bezpečné aplikace.

Množství příležitostí rozšíření aplikace nabízejí samotné schopnosti zařízení iPhone. Jak bylo zmíněno v podkapitole 2.2, iPhone disponuje fotoaparátem, mikrofonom a GPS přijímačem. Nabízí se tedy rozšíření ukládaných příspěvků o data zachycená těmito perifériemi. Současná implementace klientské části již podporuje sběr obrazové informace, stačilo by tedy rozšířit o tuto schopnost přenosový kanál a serverovou část aplikace. K přenosu souborů by bylo možné využít některou z vysokoúrovňových řešení komunikace popsanych v podkapitole 3.3. Článek by tak bylo možné obohatit např. o krátkou zvukovou stopu zaznamenanou přímo v terénu. Schopnosti přístroje sahají až do takové úrovně, že by teoreticky bylo možné přiložit k příspěvku i videosekvenci. Jelikož ale iPhone OS v současné verzi video nepodporuje, není k této možnosti pro vývoj poskytnuta příslušná dokumentace ani API. Zajímavou možností rozšíření poskytuje GPS modul. V současné verzi operačního systému je možné připojit informaci o aktuální pozici k právě pořizované fotografii.

Po přiložení informace o stávající pozici ke článku by bylo možné později např. zobrazit, kde byl který příspěvek pořízen pomocí externí aplikace Google Maps.

8 Srovnání s existujícími produkty

Různých serverů pro vytváření blogů jsou na internetu stovky. Značnou část z nich tvoří servery s univerzálnějším zařízením, servery pro publikování. Ty umožňují individualitám, společností nebo organizacím zveřejňovat informace jakéhokoliv typu, dávají tedy volný prostor pro vznik osobních a jiných blogů. Mezi nejznámější patří Blojsom, Drupal, ExpressionEngine, WordPress aj. Tyto projekty jsou často typu *open-source* a jsou tedy vyvíjeny komunitou vývojářů.

S otevřením iPhone API a vydáním vývojového prostředí začátkem roku 2008 bylo vytvořeno množství klientských aplikací, které podporují výše zmíněné publikační servery. Několik takových aplikací podporuje dokonce většinu těchto serverů zároveň. Nejpopulárnější jsou např. iBlogger nebo BlogPress.

Serverové i klientské aplikace mají za sebou již dlouhou dobu existence a vývoje. Jsou udržovány jak jednotlivci, skupinami tak i celými komunitami. Pro nové projekty je tudíž poměrně obtížné se mezi již zaběhnutými a osvědčenými aplikacemi prosadit, obzvláště mají-li vybudovanou stálou uživatelskou základnu. Pro projekt, jako je tento, který implementuje obě části nezávisle na existujících produktech, bude tedy pravděpodobně nemožné se prosadit mezi stávající výrobky na trhu. Nanejvýš by si mohl získat úzkou základnu uživatelů, pro které by implementoval specifickou funkci, kterou jiné masově rozšířené aplikace nenabízejí.

Do extrémů sahají pak internetové portály jako, jsou Facebook nebo Myspace, kde je implementován blog uživatele jen jako okrajová doplňková funkce. Tyto portály ihned po otevření iPhone API také vyvinuly pro své uživatele klientské aplikace pro většinu vyspělejších mobilních zařízení, iPhone nevyjímaje. Těmto robustním aplikacím lze jen těžko konkurovat z několika důvodů, mezi které lze zahrnout dlouhou dobu existence, vysokou funkčnost, povědomí na trhu a obrovskou uživatelskou základnu.

9 Závěr

S ohledem na vykonanou práci je možné říci, že byla splněna valná většina zadání. Podařilo se vytvořit takovou koncepci a zvolit takové prostředky, které umožňují splnit zadané požadavky. Mobilní část aplikace sbírá data a odesílá je na server. Server data přijímá, zpracovává a dále vytváří odpovídající výstup a patřičnou zpětnou vazbu. Aplikace tohoto typu nabízí však veliké množství možností na rozšíření. Všechny její části jsou implementovány s ohledem na budoucí vývoj. Mobilní, serverová část i komunikační protokol jsou koncipovány tak, aby byla zajištěna lehká rozšiřitelnost s minimálními zásahy do současného kódu a struktury aplikace. Ostatně k tomu již byly učiněny jisté kroky. Uživatelské rozhraní mobilní části poskytuje prostředky pro zachycení a upravení obrázků. Veškerá data přijatá serverem jsou ukládána do databáze a je směřováno k úpravě struktury celé aplikace, jak je naznačeno v kapitole 7.

Jeden z bodů zadání se ale podařilo naplnit jen do určité míry. Tím bylo testování implementované aplikace. Aplikace byla otestována do výše, do které jí bylo možné otestovat pomocí nástroje iPhone simulator. Tento jistě znamenitý nástroj poskytuje kvalitní prostředky pro testování iPhone aplikací co se týče uživatelského rozhraní, navigace mezi obrazovkami a do jisté míry např. i síťové komunikace. Různé vymoženosti zařízení iPhone se ale iPhone simulátoru nedostává. Není např. možné otestovat funkčnost fotoaparátu ani v případě, když počítač, na kterém je aplikace vyvíjena, fotoaparátem či kamerou disponuje. Aspekt, který by si jistě zasloužil důkladnější testování, je zmíněná síťová komunikace. iPhone simulator je sice připojený k internetu, avšak připojení pomocí mobilní sítě, které by při používání aplikace bylo využito ve větší míře než Wi-Fi, již otestovat nelze. Další prostředky, které by zřejmě byly využity během budoucího vývoje aplikace a které by nebylo možné otestovat na iPhone simulátoru, jsou např. zjištění lokace mobilního zařízení či spuštění několika instancí aplikace zároveň, což by bylo využitelné při testování konkurenčnosti serveru.

Důvody, které zabránily otestování aplikace na skutečném zařízení, jsou přinejmenším dva. Prvním jsou poměrně vysoké náklady na pořízení a druhým je politika společnosti Apple směrem k vývojářům. Stejně jako s distribucí je tomu tak, že vývojář, který má zájem o testování své aplikace na přístroji a ne v simulátoru, musí zažádat o developerský program v ceně 99 až 299 amerických dolarů. Teprve potom je mu umožněno instalovat aplikace na autorizovaný iPhone.

Tato práce zprůhledňuje tvorbu internetové iPhone aplikace se serverovým pozadím a shromažďuje informace nutné pro pochopení základních konstrukcí při tvorbě takového programu. Vývoj iPhone aplikace nabývá na obtížnosti úměrně s množstvím použitých prostředků. Bude-li chtít vývojář vytvořit pouze základní aplikaci obsahující okna a jednoduchými prvky uživatelského rozhraní a navigaci mezi nimi, bude mu pravděpodobně stačit prostudování této práce. Bude-li však chtít využít prostředky jako 3D grafiku, akcelerometr, vícedotykové události, záznam zvuku nebo videa, bude pravděpodobně muset podstoupit delší vzdělávací program v tomto odvětví. Na konkrétním případě pak práce aplikuje předem shromážděné poznatky, předkládá koncepci a implementaci iPhone aplikace.

Literatura

- [1] *iPhone Dev Center* [online]. 2009 [cit. 2009-05-01]. Dostupný z WWW: <<http://developer.apple.com/iphone/>>.
- [2] *Apple - iPhone - Gallery - Hardware* [online]. 2009 [cit. 2009-05-01]. Dostupný z WWW: <<http://www.apple.com/iphone/gallery/>>.
- [3] *Apple - iPhone - App Store* [online]. 2009 [cit. 2009-05-10]. Dostupný z WWW: <<http://www.apple.com/iphone/appstore/>>.
- [4] IANA - *PORT NUMBERS* [online]. 2009 , 2009-05-15 [cit. 2009-05-16]. Dostupný z WWW: <<http://www.iana.org/assignments/port-numbers>>.
- [5] *HTC* [online]. 2009 [cit. 2009-05-15]. Dostupný z WWW: <<http://www.htc.com/www/>>.
- [6] *SAMSUNG Česká Republika* [online]. 2009 [cit. 2009-05-15]. Dostupný z WWW: <<http://www.samsung.com/cz/>>.
- [7] *Android Developers* [online]. 2009 [cit. 2009-05-15]. Dostupný z WWW: <<http://developer.android.com/>>.
- [8] *American Dialect Society Mailing List* [online]. Sun, 20 Apr 2008 [cit. 2009-05-10]. Dostupný z WWW: <<http://listserv.linguistlist.org/cgi-bin/wa?A2=ind0804C&L=ADS-L&P=R16795&I=-3>>.
- [9] *Twitter: What are you doing?* [online]. 2009 [cit. 2009-05-10]. Dostupný z WWW: <<http://twitter.com/>>.
- [10] *Veřejné zprávy - mikroblog.cz* [online]. 2009 [cit. 2009-05-10]. Dostupný z WWW: <<http://mikroblog.cz/>>.

Seznam příloh

Příloha 1. DVD