VYSOKÉ UČENÍ TECHNICKÉ BRNO
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

VESTAVĚNÉ MODULY MODERNÍCH MOBILNÍCH ZAŘÍZENÍ

EMBEDDED MODULES OF MODERN MOBILE DEVICES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — ONDŘEJ PELIKÁN
AUTHOR

VEDOUCÍ PRÁCE — ING. KŘIVKA ZBYNĚK, PH.D.
SUPERVISOR

BRNO 2011

Abstrakt

Práce pojednává o vývoji mobilní aplikace pro platformu Windows Phone 7 zaměřenou na změnu životního stylu a řešení problémů s váhou, kondicí a fyzickým vzhledem. Aplikace využívá vestavěných modulů moderních mobilních telefonů, jako jsou GPS navigace, kamera s vysokým rozlišením a připojení k internetu. Vývoj je popsán od obecného úvodu do problému, přes popis hlavních rysů platformy Windows Phone 7, konkrétní architekturu aplikace a řešení implementačních problémů, až po popis použitých metod a nástrojů při návrhu a plánování, popis knihoven a rámců třetích stran.

Abstract

This bachelor thesis concerns mobile application development for Windows Phone 7 platform focused on lifestyle changes and dealing with problems concerning weight, physical condition and appearance. The application uses built-in modules of modern mobile phones like GPS navigation, high resolution camera and Internet connection. The application development description starts from general introduction of the issue, through description of main features of the platform Windows Phone 7, concrete architecture of the application and solution of implementation problems, to description of methods and tools used for the design and planning, library and third-party framework description.

Klíčová slova

Hubnutí, redukce váhy, zdraví, obezita, zdravý životní styl, Windows Phone 7, mobilní aplikace, klient-server aplikace, GPS, rozpoznávání čárového kódu

Keywords

Weight loss, slim, health, obesity, healthy lifestyle, Windows Phone 7, mobile application, klient-server application, GPS, barcode recognition, barcode scanning

Citace

Pelikán Ondřej: Vestavěné moduly moderních mobilních zařízení, **bakalářská práce, Brno, FIT VUT v Brně**, 2011

Vestavěné moduly moderních mobilních zařízení

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Zbyňka Křivky, Ph.D.
Další informace mi poskytli Mgr. Michal Hrabí a Lucie Opravilová, DiS z Jihomoravského Inovačního Centra.
Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Ondřej Pelikán
15. května 2011

Poděkování

Rád bych poděkovat panu Ing. Zbyňku Křivkovi, Ph.D., za cenné rady a připomínky při tvorbě práce. Dále bych chtěl poděkovat Mgr. Michalu Hrabímu a Lucii Opravilové, DiS, za všechny jejich nápady a poskytnutý mentoring.

© Ondřej Pelikán, 2011
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 3
2 Zdravý životní styl 4
2.1 Stav populace v Evropě 4
2.2 Obvyklá řešení 4
2.3 Propojení s mobilními technologiemi 4
3 Windows Phone 7 6
3.1 Základní paradigma 6
3.1.1 Hardwarová specifikace 6
3.1.2 Běhové prostředí 7
3.1.3 Možnosti WP7 pro vývojáře 8
3.2 Specifický grafický styl 9
4 Aplikace Zhubni! 11
4.1 Specifikace 11
4.1.1 Ovládání a vzhled 11
4.1.2 Funkce 11
4.1.3 Inovace 12
4.2 Návrh 13
4.3 Architektura 15
4.3.1 Klient 16
4.3.2 Komunikace 18
4.3.3 Server 19
4.4 Grafické rozhraní 20
4.4.1 Menu 21
4.4.2 Jídlo 22
4.4.3 Aktivita 22
4.4.4 Vyhodnocení 22
4.4.5 Nápověda 24
4.4.6 Zpracování chyb 24
4.5 Použité knihovny a postupy 24
4.5.1 MVVM Light Toolkit 25
4.5.2 Location Mock 25
4.5.3 John Papa 26
4.5.4 Windows Phone 7 Design Templates 26
4.5.5 Coding4Fun Windows Phone Toolkit 27
4.5.6 ZXing Barcode Scanning Library 27
4.6 Implementace 27
4.6.1 Model 28
4.6.2 View – ViewModel 29
4.7 Uživatelské testy 30

4.8 Použité nástroje 31
4.8.1 Visual Studio 2010, ReSharper, Emulator 31
4.8.2 Expression Blend 32
4.8.3 Ostatní nástroje 33
4.9 Vize do budoucna 35
5 Závěr 36
Bibliografie 37
Seznam příloh 39

1 Úvod

Žijeme ve světě, kde si nebezpečí z nedostatku potravy umí málokdo představit. Všude kolem nás jsou plné hypermarkety levných, dostupných, průmyslových potravin, které mají lákavé obaly i ceny. Bohužel velká většina z nich není vhodná pro zdravý životní styl, a některé potraviny dokonce pro konzumaci vůbec. Lidé se v této džungli přestali orientovat. Nechali se strhnout cenami, vzhledem a snadnou dostupností místo toho, aby se řídili zdravým rozumem. Díky tomu lidé stále více trpí obezitou a nadváhou nejenom v USA, ale i v ČR nebo dokonce v asijských státech, kde se tyto problémy ještě před pár lety vůbec nevyskytovaly.

Téma bakalářské práce mi nezadávalo přímo konkrétní problém k řešení, ale pouze nástroje, které mám použít a demonstrovat jejich funkčnost. Funkce modulů moderních mobilních zařízení jsem se rozhodl demonstrovat právě na řešení problému, který trápí obrovskou část populace ČR i celého světa – nezdravý styl života.

Nejprve rozeberu problém se zdravým životním stylem obecně. Poté se podíváme, jak by se řešení problému dalo usnadnit pomocí moderních mobilních zařízení. Dále nastíním základní vlastnosti platformy Windows Phone 7 a základy vývoje pro tento operační sytém. V části o návrhu se podíváme na to, jak jsem postupoval a jaké metody jsem použil při návrhu aplikace a její implementace. Poté projdeme architekturu klienta, způsob komunikace se serverem, grafické uživatelské rozhranní, použité knihovny a rámce, vlastní implementaci a nakonec použité nástroje při vývoji. Program prošel i uživatelskými testy, které jsem vyhodnotil, a na základě toho jsem určil další vývoj aplikace. V závěru je celá práce shrnuta.

2 Zdravý životní styl

V následujících kapitolách se budu snažit nastínit, jaký je stav problémů s nadváhou v Evropě a České republice. Následně se podívám na možná řešení a to, jak by se daly uplatnit mobilní zařízení.

2.1 Stav populace v Evropě

O obezitě se někdy mluví jako o *epidemii třetího tisíciletí*. Je velké množství studií dokazující vhodnost tohoto označení.

World Health Organization (WHO) v dokumentu o stavu obezity a nadváhy v Evropě [1] uvádí, že obezita se týká 30 – 80% dospělých v jednotlivých zemích a kolem 20% dětí. Tento už tak vážný stav se ještě každým rokem prudce zhoršuje.

Stav v České republice je ze všech evropských států jeden z nejhorších. Ústav zdravotnických informací a statistiky České republiky vydal v roce 2008 Aktuální informace o zdravotním stavu v ČR [2], kde uvádí, že v roce 2008 mělo problémy s nadváhou 63% mužů a 46% žen. Dále web obezita.cz [3] nebo novinky.cz [4] uvádí, že když sečteme všechny lidi trpící obezitou nebo nadváhou, dostaneme obrovská čísla 72% u mužů a 68% u žen.

2.2 Obvyklá řešení

Při pohledu na tato enormně vysoká čísla je patrné, že se jedná opravdu o velký problém. Jeho řešení není nijak snadné, i když doporučených postupů je nepřeberné množství. Známe desítky a možná stovky druhů diet. Některé jsou čistě odborné a spoléhají na přesné měření gramáže a dodržování základních složek potravy. Jiné diety připomínají spíše šamanské rituály. U velkého počtu diet už se také dokázal negativní vliv na lidské zdraví, ale přesto si je mnoho lidí vybírá třeba jen kvůli tomu, že se pod ně podepsala nějaká slavná herečka (i když životní styl mnoha celebrit bývá velmi pochybný a rozhodně se nedoporučuje jej následovat).

Snad každý, kdo se dostal do problémů s nadváhou a obezitou zkoušel tento problém řešit, ale mnozí velmi záhy ztroskotali. Většina lidí končí proto, že k dodržování všech podmínek a požadavků zdravého životního stylu je potřeba mít poměrně velké znalosti, hodně času, musí se provádět nezábavné a časově náročné úkoly jako např. psaní deníků a další. Tyto negativní vlivy ubírají velkými kusy z počáteční motivace, protože žít nezdravě je dnes přeci o tolik snazší.

2.3 Propojení s mobilními technologiemi

Když přijdeme do obchodu a máme v úmyslu nakoupit jen samé zdravé potraviny, často už si po pár minutách vůbec nevíme rady. Jen podle nutričních hodnot, bez dalších informací, málo lidí opravdu pozná, zda je potravina zdravá a vhodná. Neusnadňují nám to ani matoucí nápisy jako *light* apod., které většinou vůbec nesouvisí s tím, jestli je potravina zdravá nebo ne. Viz [5].

Většinou máme u sebe ale mobilní telefon, který nám může celou situaci velice usnadnit. Přes mobil se můžeme vzdáleně připojovat k databázi potravin a získávat tak podrobné informace bez toho, abychom je museli dlouze studovat. Prohlížení databáze může být ještě usnadněno, když máme

možnost oskenovat čárový kód na obalu a během vteřiny se dozvědět, zda si potravinu odnést k pokladně, nebo ne.

Rozhodování přímo v obchodě je velice důležité, protože když si budeme nosit domů jen zdravé potraviny, budeme se nutně stravovat zdravě. Často se lidé dostávají do situace, že si špatnou volbu potravin uvědomí až po nákupu, ale to už je pozdě. Není moc lidí, kteří by polovinu nákupu vyhodili – většinou nevhodné potraviny snědí a vždy si říkají, že do příště se polepší.

Kdo to s hubnutím myslí vážně, většinou se dostane do situace, kdy by si měl vést deník jídla a aktivit, počítat energii a jiné údaje, jak už jsem naznačil v kapitole 2.2. V této oblasti může mobilní telefon také velmi dobře posloužit. Nejen že se dá rychle použít k zapsání jakékoliv činnosti, ale může všechny výpočty provést okamžitě a automaticky.

S takovým pomocníkem se motivace nebude ztrácet tak rychle a celý proces se bude zdát pohodlnější a možná i zábavnější. Rozhodl jsem se právě takovou aplikaci vytvořit a pomoci tak lidem k zdravějšímu a snad i šťastnějšímu životu.

3 Windows Phone 7

Pro mobilní aplikaci jsem si vybral platformu Windows Phone 7 (dále jen WP7) od firmy Microsoft. Tato platforma je úplně nová a v počátcích bakalářské práce nebylo na trhu v České republice ani jedno zařízení. V době odevzdání bakalářské práce přístroje WP7 poskytují již výrobci LG, HTC, Samsung a dokonce Nokia uzavřela s firmou Microsoft smlouvu a v blízké budoucnosti bude do svých telefonů používat hlavně WP7.

V době psaní této bakalářské práce nebyl v ČR otevřen Marketplace[1] pro WP7. Tento fakt může být brán i jako výhoda – zatím není žádná konkurence, tato aplikace bude jako jedna z prvních a otevření marketplacu bude jistě provázet silná mediální kampaň vedená firmou Microsoft, jejíž součástí by se tato aplikace mohla stát.

WP7 platformu jsem si vybral kvůli dobrým zkušenostem s vývojem v technologiích firmy Microsoft a také kvůli spolupráci s Microsoft inovačním centrem a Jihomoravským inovačním centrem, které mi slíbilo zapůjčit zařízení s WP7 systémem odblokovaným pro vývoj aplikací, což může být v současné době v ČR problém. Zároveň mě lákalo tuto novou platformu prozkoumat a být jeden z prvních, kdo se programováním pro tento mobilní operační systém bude zabývat.

3.1 Základní paradigma

Každá platforma má nějaká svoje specifika, se kterými je potřeba při vývoji počítat.

3.1.1 Hardwarová specifikace

Všechny telefony běžící na platformě WP7 musí v současné době splňovat hardwarové podmínky:

- dotyková obrazovka o přesně daném rozlišení – 480 × 800 nebo 320 × 480;
- tlačítka *Back*, *Start* a *Search* pod dotykovou obrazovkou;
- připojení k internetu přes Wi-Fi;
- nejméně 5 MPx kameru;
- akcelerometr;
- GPS modul;
- programově dostupné vibrace;
- FM rádio přijímač;
- *Push Notifications* – systém pro oznamování událostí z internetu (viz [6]).

Tyto hardwarové specifikace jsou pro programátory velmi výhodné, protože umožní spolehnout se vždy na stejnou minimální konfiguraci. To je velký problém třeba u telefonů s operačním systémem Android, kde každé zařízení může být dost odlišné a programátor pak musí zjišťovat, na jak vybaveném telefonu je jeho program spouštěn. Na druhou stranu oproti iPhone je výhoda ta, že

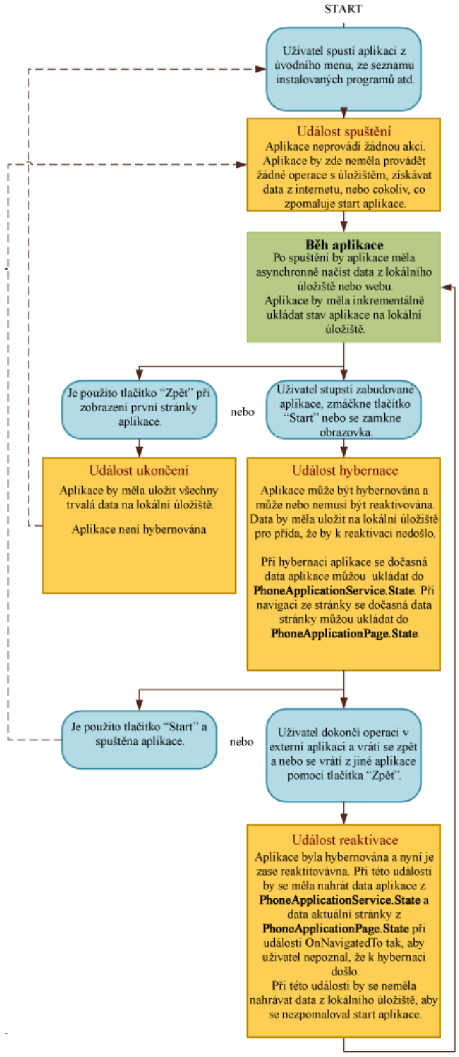
[1] Marketplace je oficiální webová služba společnosti Microsoft a jediné místo kde se dají běžně získávat aplikace pro WP7 a to i ty zdarma. Jediný další způsob je, mít odemčený telefon pro vývoj a nahrávat aplikace pomocí vývojářských nástrojů. Ostatní způsoby jsou neoficiální a v rozporu s licenčním ujednáním.

mobilní telefony jsou vyráběny různými výrobci, kteří se snaží konkurovat – vybavením, designem apod., takže o rozmanitost přístrojů není nouze.

3.1.2 Běhové prostředí

Pro WP7 je programovací prostředí vycházející ze Silverlight verze 3/4[2]. Mezi klasickou a mobilní verzí Silverlight jsou v určitých vlastnostech znatelné rozdíly, protože je samozřejmě rozdíl v běhu webové a mobilní aplikace.

Nejvýraznější změnou je životní cyklus aplikace. Na WP7 není programátorům umožněno využití *multitaskingu*, takže se musel zavést speciální model vykonávání programu. Zavádí se pojem *hybernace (tombstoning[3])* a programátor se musí postarat o správné ukončení a následné spuštění aplikace, aby uživatel nepoznal, že k této hybernaci aplikací dochází. V jádře WP7 jsou k tomu připravené nástroje, takže to není nijak složité. Více viz `TransientState` v [7].



Obr 3.1 Životní cyklus aplikace převzato z [7]

[2] WP7 vychází hlavně z verze 3, ale obsahuje i některé funkce verze 4.

[3] Pojem Tombstoning označuje akce, které vedou k ukončení aplikace z důvodu přerušení (telefonní hovor, spuštění jiné aplikace apod.). Do aplikace je možnost se vrátit a správné načtení posledního stavu se musí programově ošetřovat. Více viz [7].

Více o změnách mezi verzemi Silverlight pro webovou a mobilní platformu, se dá dohledat na MSDN [8].

Výrazným rysem programování pro WP7 je běh ve vyhrazeném prostoru (*sandbox*). Tato izolace byla z důvodu bezpečnosti převzata z klasické verze Silverlight a znamená to, že program se nemůže dostat k fyzické struktuře souborového systému, ale je mu přiřazen izolovaný a omezený prostor, do kterého si může ukládat data jen sama aplikace a nikdo jiný k nim nemá přístup. Je to poměrně značné omezení, které se řeší v těch jednodušších případech pomocí vestavěných aplikací (viz kapitola 3.1.3) nebo v těch složitějších pomocí webových služeb.

Navíc díky tomu, že aplikace běží v izolovaném prostoru, neexistuje možnost dostat se na funkce operačního systému, které vývojáři přímo nezpřístupnili. Proto je prozatím jen velmi omezený výběr funkcí, které můžeme programově ovládat. Více viz kapitola 3.1.3.

Znamená to také, že nelze spouštět nativní kód. Není možné používat již napsané knihovny v C/C++ ani v Jazyce symbolických instrukcí. Proto se veškeré funkce a programy musí pro WP7 převádět/přepisovat. V době psaní bakalářské práce byla například pro platformu WP7 přepsaná jen jedna knihovna pro rozpoznávání čárového kódu, i když pro jiné platformy jich bylo naprogramováno velké množství a mnohé dokonce jako *open source*.

Dalším specifikem je jazyk XAML pro popis grafického uživatelského rozhranní (GUI). Tento jazyk vychází z jazyka XML a vytvářejí se pomocí něj hlavně prvky uživatelského rozhranní (ale i jakékoliv jiné třídy s bezparametrickým konstruktorem). Na uživatelské rozhraní popsané jazykem XAML se pak napojuje tzv. *code behind* neboli kód v pozadí, což je klasický objektový kód psaný v jazyce Visual Basic nebo C#. Výhodou toho přístupu je větší přehlednost a pochopitelnost. Na deklarativní popis toho, co má jak vypadat a z čeho se to má skládat, je značkovací jazyk velice vhodný. Chování se pak popíše imperativním/objektovým stylem. Podobný styl programování použila firma Adobe pro platformu JavaFX, což měla být konkurence platformy Silverlight, ale bohužel se moc neuchytila.

O WP7 jsou napsány stovky článků a rozborů, proto zde nebudu tuto problematiku více rozebírat.

3.1.3 Možnosti WP7 pro vývojáře

Jediný hardware na WP7 přístupný programátorům, je GPS modul a akcelerometr. Pro tyto dva moduly je poskytnuto veřejné rozhranní (API[4]). Další užitečná API se teprve chystají ke zveřejnění (jako například přímý přístup ke kameře) a u některých není zveřejnění jisté vůbec (GSM modul). Díky takovýmto omezením, není například možné naprogramovat nahrávání hovorů a prozatím, ani funkce rozšířené reality (to se změní na podzim 2011 s příchodem aktualizace Mango).

WP7 však poskytuje několik předinstalovaných aplikací pro využívání zakázaných funkcí operačního systému a sdílení dat napříč aplikacemi. Tyto externí aplikace mohou být volány z kódu buď jen za účelem vykonání nějaké akce, nebo vykonání nějaké akce a vrácení výsledku (*Launchers and Choosers).*

[4] Jde o sbírku procedur, funkcí či tříd nějaké knihovny (ale třeba i jiného programu nebo jádra operačního systému), které může programátor využívat, viz http://cs.wikipedia.org/wiki/API.

Hlavní předinstalované aplikace jsou:

- `EmailComposeTask` spouští aplikaci pro psaní emailů.
- `MediaPlayerLauncher` spouští aplikaci *Media Player* a přehrává vybrané video.
- `PhoneCallTask` spouští aplikaci pro telefonní hovory.
- `SmsComposeTask` spouští aplikaci umožňující napsat SMS.
- `WebBrowserTask` spouští webový prohlížeč na zadané adrese.
- `CameraCaptureTask` spouští aplikaci pro focení a vrací výsledek.
- `PhotoChooserTask` spouští aplikaci pro výběr pořízených fotek.
- Další viz [9].

Zajímavé možnosti poskytují i ovládací prvky GUI přímo zabudované do WP7. Poskytují kompletní funkčnost pro prohlížení webu, přehrávání videí v HD rozlišení nebo zobrazování mapy a snadné zakreslování do map. Bohužel v současné době je možné zobrazovat mapy jen v online režimu a pouze *Bing Maps* od společnosti Microsoft. Mapy ostatních poskytovatelů jsou v rozporu s licenčními podmínkami ovládacího prvku, nicméně už se pracuje na dalších ovládacích prvcích, které by používaly mapové podklady třetích stran, a očekává se, že budou fungovat offline.

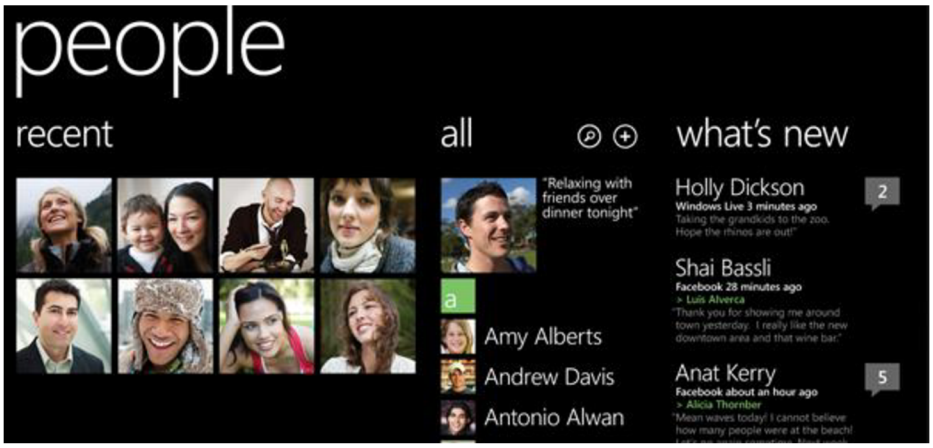
3.2 Specifický grafický styl

Výrazné odlišení oproti ostatním platformám je vidět na první pohled v grafickém návrhu aplikací. Microsoft se rozhodl vytvořit netradiční, velmi jednoduchý a přehledný styl grafického návrhu nazvaný *Metro*, inspirovaný především stylem informačních tabulí, návěští a ukazatelů v metru a na letištích.

Hlavní filosofie grafického stylu *Metro*:

- čisté, lehké, otevřené a rychlé;
- zaměření na informaci ne dekoraci;
- integrace hardware a software. Konzistentní vnímání GUI mezi aplikacemi díky hardwarovým tlačítkům (viz kapitola 3.1.1);
- stejná gesta a pohyby na dotykovém displeji, jako na klasických Windows. Stejná rychlost i odezva díky hardwarové akceleraci animací, transformací a přechodů.

Obr 3.2 Styl metra a letišť, který se stal inspirací.



Obr 3.3 Příklad jednoho z nových UI schémat zvané Panorama

Pro další seznámení se systémem WP7 a jeho grafickým vzhledem se můžete více dočíst v [10].

4 Aplikace Zhubni!

Po prozkoumání problému moderní civilizace (obezita – viz kapitola 2.1), jsem se rozhodl vytvořit aplikaci, která pomáhá s jeho řešením nebo aspoň jeho řešení usnadňuje. Je zaměřena hlavně na jednoduchost a motivaci, aby i lidé, kteří při běžném postupu (viz kapitola 2.2) nevydrží a nedostanou se daleko, mohli aplikaci používat dlouhodobě. Nemá nahrazovat nutriční poradce a odborníky na výživu, spíše by měla být další možností pro lidi, kteří nemají na tyto odborníky peníze nebo se třeba jen stydí někam docházet osobně.

4.1 Specifikace

Nejdůležitější specifikací je zadání bakalářské práce samotné a to je využití tří modulů moderního mobilního telefonu. Jak jsem uvedl v kapitole 3.1.3, nelze použít všechny funkce telefonu a tak jsem si po konzultaci s vedoucím práce vybral GPS modul, kameru a připojení k internetu.

4.1.1 Ovládání a vzhled

Specifikace v bodech:

- Hlavní důraz je kladen na jednoduchost a přehlednost.
- Dbá se na zachování a integraci stylu *Metro*.
- Nenutit uživatele ke zbytečně složitým úkonům.
- Při vyhledávání je nutno poskytovat funkci našeptávání.
- Je nežádoucí zatěžovat uživatele nadbytečnými číselnými údaji o jídle. Je potřeba ho co nejvíce odstínit od nutričních hodnot a jakéhokoliv počítání.
- Důraz je kladen i na příjemné a jednoduché zobrazení vyhodnocení stavu.
- Každý úkon by měl jít provést maximálně s přechodem do tří obrazovek.
- Je nutné minimalizovat psaní na klávesnici.
- Uživateli je poskytnuta nápověda, jak aplikaci ovládat.

4.1.2 Funkce

Specifikace v bodech:

- Aplikace dokáže vyfotit a rozpoznat čárový kód s určitou pravděpodobností (záleží na použité knihovně pro rozpoznávání).
- Aplikace umí pomocí GPS měřit vzdálenost a rychlost u sportů, které využití GPS umožňují. To napomáhá k lepšímu výpočtu indexu vhodnosti a k motivaci uživatele, aby posouval hranice svých aktivit.
- Aplikace se dokáže připojit ke vzdálené databázi a vykonávat tam následující akce:
    - Vyhledat jídla obsahující daný podřetězec.
    - Vyhledat jídla podle čárového kódu.
    - Vyhledat dodatečné informace o jídle (doporučení, vlastnosti, alternativy).
    - Vyhledat aktivity obsahující daný podřetězec.
    - Nechat si ohodnotit jídlo nebo aktivitu indexem vhodnosti.

- Vložit zkonzumované jídlo do uživatelské historie.
- Vložit vykonanou aktivitu do uživatelské historie.
- Získat vyhodnocení uživatelské historie k určitému datu.

Výpočty prováděné na serveru nebyly součástí zadání a je možné, že v době odevzdání bakalářské práce nebudou finální.

4.1.3 Inovace

Odborníky doporučované přístupy k hubnutí (sledování poměru příjmu a výdeje energie, dodržování vhodné skladby jídelníčku) jsou pro člověka složité v tom, že si musí pamatovat pravidla a vzorce, jak hodnotit potraviny, jak si sestavit správný jídelníček apod. V této oblasti jsem se snažil uživatelům práci ulehčit a zavedl jsem proto index zátěže a zdravoměr.

Index zátěže

Jako zásadní inovace je zaveden index zátěže. Je to jedna číselná hodnota vyjadřující zátěž na organismus. U jídla – čím je hodnota vyšší, tím je situace horší a u sportu je to naopak – vyšší je lepší[5].

Tato hodnota se počítá dynamicky a pro každého uživatele bude ve výsledku jiná, i když se bude jednat o stejné jídlo ve stejně velké porci.

Každý člověk má totiž jinou historii stravování a jiné hodnoty každé složky v jídelníčku. Pro většinu lidí bude například tučnější jídlo mít vysoký index zátěže, ale tomu, kdo jí jen samou zeleninu a ovoce, můžou v potravě tuky scházet. Tuk je potřebný pro správné trávení a naše tělo určitou míru tuků pro správné fungování potřebuje. Takový člověk, ale na druhou stranu, bude mít vyšší index u ovoce, protože ho jí hodně. Ovoce obsahuje vysoké množství cukrů a ani s ním by se to nemělo přehánět. Naopak člověk stravující se výhradně na mase (tuky a bílkoviny), bude potřebovat doplnit zdravý cukr a vitamíny, proto u ovoce bude mít velmi nízký index (pro něj bude ovoce velmi vhodné) [11].

Postup výpočtu indexu u jídla:

- Základem jsou energetické a nutriční hodnoty.
- K tomu se dají přidat různé nálepky (*tagy*), které mohou index výrazně změnit. Příklad takových nálepek je: smažení, vaření, dušení, grilování, velmi kořeněné, velmi slané atd.
- Potom se projde uživatelská historie a zjistí se přebytky a nedostatky jednotlivých složek. To může posunout index vhodnosti i dost výrazně.

Index vhodnosti se nepočítá jen pro jídlo ale i pro sport. Hodnoty indexu pro jídlo a sport jsou na stejné škále, takže se dají vzájemně porovnávat. Díky tomu se dá vyjádřit, že např. 300g porce smaženého řízku s bramborovým salátem se bude rovnat dvěma hodinám běhání. To by mělo uživatele motivovat k tomu, aby si vybral spíše nějaké lehčí jídlo, které se dá snadněji spálit. Navíc se tak snadno počítá uživatelské vyhodnocení, protože stačí jen sečíst indexy jídla a odečíst indexy sportu. Je to dost podobné jako čisté porovnávání příjmu a výdeje energie, ale index zátěže zahrnuje víc důležitých informací, než jen energii.

[5] Index zátěže se v budoucnosti změní na index vhodnosti se stupnicí od -10 do 10. Tato varianta je na výpočet složitější a proto není zatím implementována.

Zdravoměr

Další větší inovace je funkce zdravoměru. Jedná se o uživatelsky příjemné a motivující zhodnocení aktuálního stavu. Hlavní myšlenka je v tom, aby uživatel jedním pohledem věděl, jak na tom zrovna teď je, a dostal i doporučení, co by měl/neměl dělat. Takové zhodnocení je vždy aktuální a je zaměřeno na každodenní motivaci. Nutí uživatele, aby zdravoměr ukazoval vždy pozitivní hodnocení. Pro uživatele by nemělo být příjemné končit den s tím, že jim mobilní telefon konkrétně sděluje, že dnes svému cíli nejdou vstříc a naopak se třeba od cíle i vzdalují.

Takový druh motivace je zaveden z toho důvodu, že lidé dost často při dlouhodobých činnostech zapomínají na výsledný cíl a ztrácejí motivaci. Takto mají vždy malý cíl na každý den. Na takové malé krůčky se dá lépe soustředit.

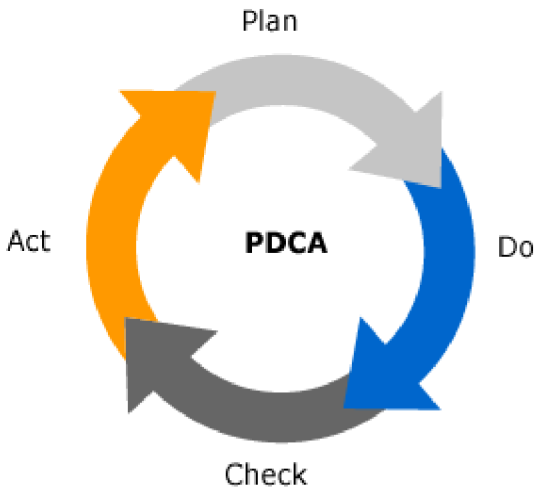
4.2 Návrh

Návrh celé aplikace jsem se snažil pojmout spíše v duchu agilních metodik. Snažil jsem se vždy o zprovoznění menších celků a aplikaci navrhoval modulárně. Takový přístup obecně dobře řeší metodika PDCA (*Plan, Do, Check, Act*). Pro řízení vývoje projektu jsem používal podmnožinu metodiky SCRUM a implementaci jsem navrhoval pomocí testy řízeného programování (TDD – *Test Driven Design*).

PDCA

Metodika PDCA se používá často v procesu zlepšování.

- *Plan* – shromažďují se data o hlavních problémech a jejich příčinách. Podle toho se pak naplánuje nejvhodnější řešení.
- *Do* – realizuje se vybrané řešení.
- *Check* – testuje se a vyhodnocuje, zda bylo dosaženo zamýšlených výsledků.
- *Act* - hodnotí se celý proces, aplikují se další vylepšení a vyvozují se závěry jako poučení pro příště. Přechází se opět na první fázi *Plan*.



Obr 4.1 PDCA cyklus, převzato z [12]

Tohoto cyklu jsem využíval při vymýšlení dalších postupů a různých vylepšení. Když jsem ukončil vývoj jedné části a rozmýšlel se jak pokračovat. Například jsem si naplánoval vytvoření našeptávače, který se bude připojovat k serveru. Sehnal jsem všechny potřebné informace. Metodou

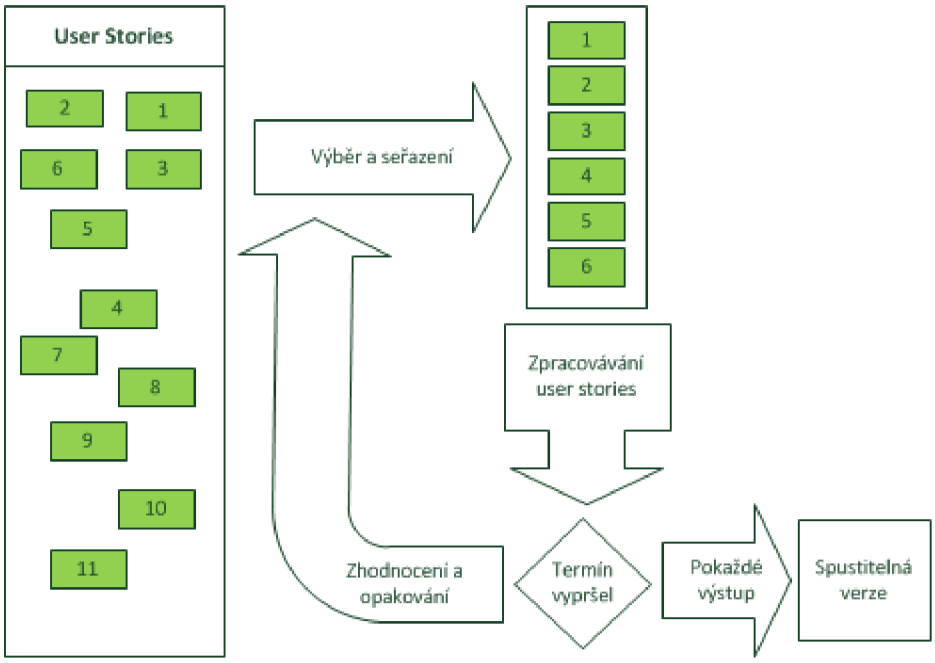
SCRUM jsem naplánoval dílčí kroky v čase a konkrétní implementaci jsem navrhl pomocí TDD. Testování se zčásti pokryje metodou TDD, ale je nutné provést i ruční testování při reálném nasazení. Na závěr jsem zhodnotil celý postup a uvědomil si, kde jsem udělal chyby a jak se jich pro příště vyvarovat.

SCRUM

Metodika SCRUM je hodně používaná v moderních softwarových firmách a stále více firem na ni přechází ze zastaralých vodopádových modelů.

Velká část této metodiky se týká spolupráce v týmu a hodnocení, ale protože jsem pracoval sám, nebudu tyto části rozebírat. Je to jedna z nejpoužívanějších agilních technik vývoje, takže materiálů je hodně i v češtině – doporučuji hlavně [13]. Zde zdůrazním jen ty části, které jsem opravdu využil.

Nejprve všechno začíná rozdělením problému na *user stories*. To jsou malé úkoly, které zákazník potřebuje implementovat (ohodnocení jídla indexem, zobrazení indexu, menu s tlačítky apod.) Zákazník tyto *user stories* ohodnotí prioritou. Priorita musí být celé číslo a nesmí se opakovat. Programátor si pak zvolí časový úsek (v mém případě 14 dní) a vybere si jen takový počet *user stories*, které může v daném období splnit. Podle priorit pak bere jednu *user story* po druhé a provádí implementaci. Po uplynutí termínu se vydá spustitelná verze projektu a práce se zrekapituluje. Vyhodnotí se, jak dobře byly provedeny časové odhady a jaké problémy při implementaci nastaly. Tato část je vždy nutná pro lepší budoucí plánování a předcházení příštím problémům.



Obr 4.2 Scrum diagram

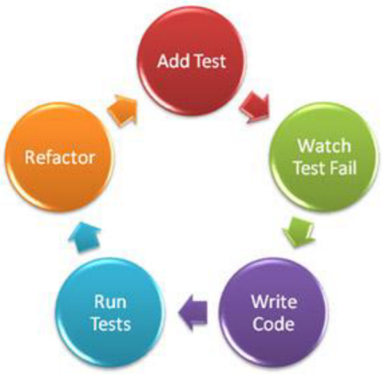
TDD

Na první pohled možná není jasné, proč metoda *Testy řízené programování* [14] patří do metodik návrhu. V tomto případě se jedná o návrh zespodu. Používá se při implementaci jednotlivých funkcí a tříd. Probíhá to tak, že si nejprve napíšeme testy pro budoucí třídu nebo jakoukoliv funkčnost. Ná-

sledně podnikneme co nejmenší kroky k tomu, aby se nám tesy spustily. Dále podnikneme co nejmenší kroky k tomu, aby testy úspěšně prošly a po tom co testy projdou, se provádí *refaktoring*[6]. To znamená odstranit kopírování kódu, rozdělit metody apod. Výborná kniha o změnách v kódu a správných zásadách funkčního a čitelného kódu je [15].

Metoda TDD při návrhu pomáhá opravdu znatelně. Pokud si totiž nejdřív ujasníme cíle, srovnáme si v hlavě, jak všechno má do posledního detailu fungovat, a nemusíme to mít v hlavě, ale všechno je zaznamenáno v testech, je snazší soustředit se jen na řešení konkrétních funkcí. Tím odpadá uvažování o programu v celé jeho komplexnosti. Jednotlivé malé kroky a následný *refaktoring* vedou k minimalistickému řešení – vždy se implementuje pouze daný problém a neřeší se zbytečnosti.

I přes to, že velká většina kódu je otestována, musí se provádět různé další testy v reálných podmínkách. Výhoda je ale hlavně v tom, že si musíme dopředu rozmyslet jaký je výsledek každé, i té nejmenší funkce.

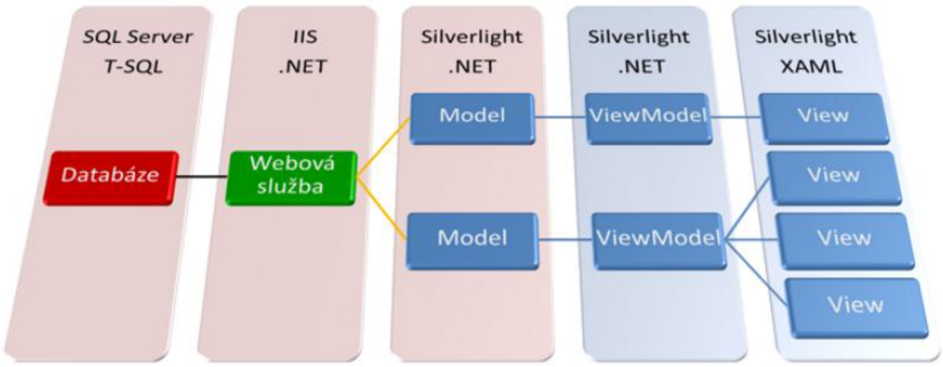


Obr 4.3 Cyklus metody TDD, převzato z [16]

4.3 Architektura

Klasická architektura aplikace psané v Silverlightu se skládá většinou z databáze, webové služby poskytující veřejné rozhranní (SOAP nebo REST) a klienta. Klient může Silverlight ve webové nebo mobilní verzi. Základní struktura klientských aplikací na platformě Silverlight se používá buď tak jak je navržená od tvůrců – každá stránka se skládá z XAML popisu a kódu v pozadí v C#, nebo ve složitějších aplikacích se používá návrh pomocí vzoru *Model-View-ViewModel* (MVVM), což je obdoba klasického *Model-View-Controller* (MVC) přizpůsobená pro Silverlight.

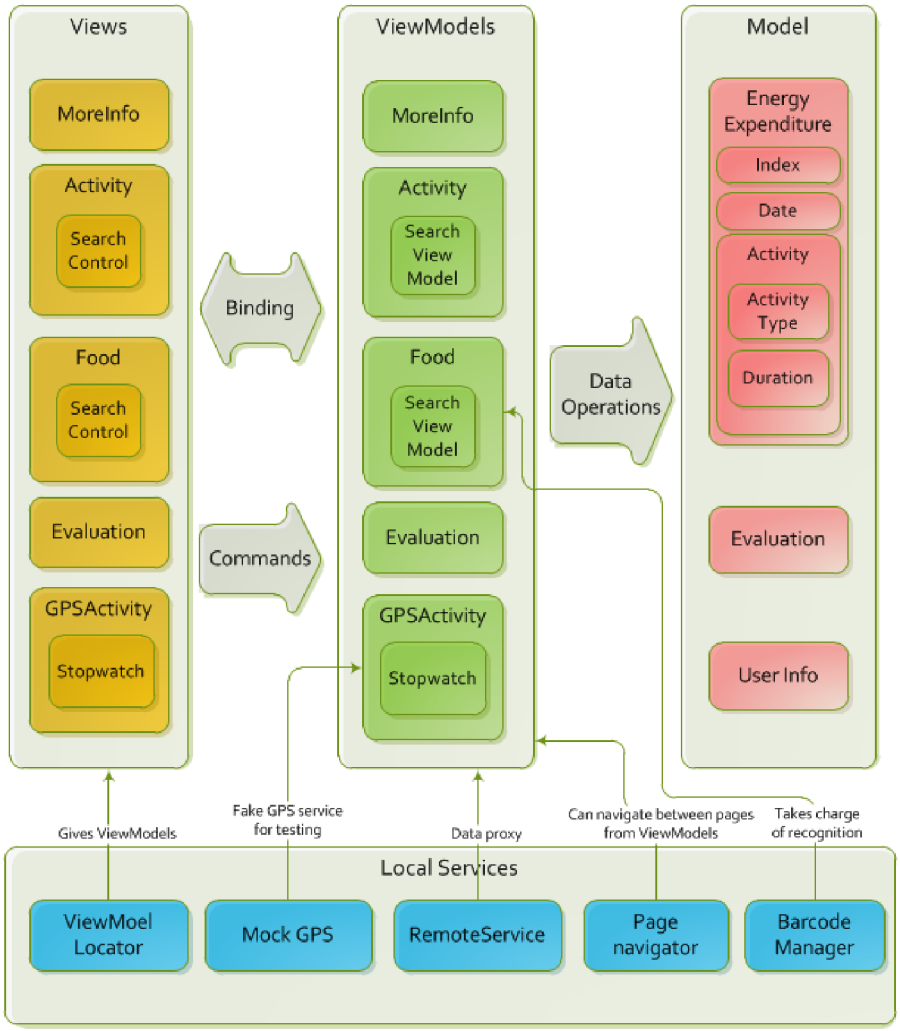
[6] Refaktoring je zlepšení kódu, které nemá vliv na jeho chování.



Obr 4.4 Klasická architektura Silverlight aplikace, převzato z [17]

4.3.1 Klient

Architekturu klienta jsem navrhl podle vzoru MVVM (více viz další podkapitola). Abych co nejvíc oddělil zodpovědnosti jednotlivých tříd, navrhl jsem si jednoduché *lokální služby*, které poskytují svému okolí funkce s určitým zaměřením. Nejsou to služby, které by běžely někde mimo program, ale jen zapouzdřené skupinky tříd nesoucí zodpovědnost za určitou funkčnost aplikace. Některé služby jsou převzaty z knihoven a rámců třetích stran (viz kapitola 4.4.5). Třídy reprezentující *Model* slouží k reprezentaci dat a jejich vazeb.



4.5 Ukázka architektury klienta. Na obrázku nejsou vyobrazeny všechny třídy. Slouží pro ilustraci.

Model, View, ViewModel (MVVM)

V klasickém přístupu programování pro platformu Silverlight by se všechna funkčnost měla psát do kódu v pozadí (*code behind*) přímo na stránce, která je zobrazena a kde se kód vykonává. Pro jednoduché aplikace je to, podle mého názoru, nejlepší postup. Složitost třídy ale s její chytrostí/zodpovědností roste exponenciálně. Proto základní pravidlo čistého kódu je, aby třída měla vždy jen jednu zodpovědnost a byla rozdělena do několika méně mocných tříd.

Jedním z řešení rozdělení tříd a delegování zodpovědnosti pro platformu Silverlight je návrhový vzor MVVM. Je podobný klasickému návrhovému vzoru MVC, ale využívá výhod a schopností této platformy jako jsou vazby (*binding)* a příkazy (*commands*).

Popis hlavních částí MVVM [17]:

- *Model* obsahuje referenci na zdroj dat (v tomto případě na webovou službu). Obsahuje třídy pro reprezentaci dat ze služby. Implementuje se zde i *datová proxy třída*[7]. Částí představující *Model* můžeme mít i víc – např. více modelů používající jednu službu a reprezentující data v různých kontextech, nebo každý model může mít různou službu a skládat data do jedné vrstvy.
- *View* je grafické zobrazení pohledu psaném v jazyce XAML. Část kódu v pozadí by se měla co nejvíce minimalizovat. Data pro formulářové prvky se čerpají z části *ViewModel* skrze vazby a události těchto prvků se převádí na příkazy, které jsou také automaticky spouštěny ve třídách *ViewModel*. O veškeré chování a transformaci dat se pak už stará *ViewModel*. *View* může být složeno z jiných *View* jako strom. Dají se pak vytvářet složité kompozitní struktury znovupoužitelných prvků.
- *ViewModel* je prostředník mezi částí *Model* a *View*. Všechna data transformuje téměř vždy tak, aby mohla být zobrazena přes *View* automaticky jen pomocí vazby bez dalších transformací. Části *ViewModel* jsou velmi výhodné z důvodu testování. Protože veškeré chování je prováděno v těchto třídách (tyto třídy jsou normální třídy v objektovém modelu), mohou být snadno (na rozdíl od *View*) spravovány automatickými testy. Z důvodu testování se do části *ViewModel* pak *Model* předává v konstruktoru (tomu se říká *dependecy injection*) a to umožní v testech podsunout falešnou třídu *Model*, která se pak snadněji testuje – například místo webové služby bude čerpat data z lokálně uložených XML souborů.

Architektura MVVM se dá ještě různými způsoby vylepšovat. Původně byla vymyšlena pro WPF[8]. WPF je daleko bohatší než Silverlight a tak se začaly objevovat rámce, které se snaží různými způsoby zúžit tuto mezeru. O tom, které rámce a knihovny třetích stran jsem použil, pojednává kapitola 4.4.5.

---

[7] Datová proxy třída je třída, která zastupuje vzdálenou službu a pro zbytek programu vystupuje jako vzdálená služba sama. Umožňuje to oddělit logiku programu od logiky komunikace.

[8] WPF – Windows Presentation Foundation je moderní nástupce WinForms – zastaralá platforma pro vývoj desktopových aplikací od společnosti Microsoft.

Lokální služby

- `PageNavigator` – umožňuje řídit navigaci z částí *ViewModel*. Dokáže předávat při navigaci i objekty a ukládat stavy při hibernaci. Tato služba je pro návrhový vzor MVVM velmi vhodná, protože obvykle je navigace dostupná jen z tříd *View*, které by neměly definovat žádné složitější chování nesouvisející se vzhledem – což navigace a předávání stavů mezi stránkami není.
- `RemoteService` – zapouzdřuje vzdálenou komunikaci. Slouží jako výše zmiňovaná *datová proxy třída*. Provádí serializaci, deserializaci, zpracování chyb a ošetřuje různé nástrahy vzdáleného přenosu (vypršení časových limitů, nesprávné pořadí příchozí komunikace atd.)
- `ViewModelLocator` – poskytuje centrální místo pro získávání tříd *ViewModel*. Je navržen tak, aby View mohl svůj *ViewModel* získat automaticky přes vazbu. Ve výjimečných případech mohou zde i ostatní třídy získávat reference na jakýkoliv *ViewModel*, ale zbytečně to narušuje čistý návrh architektury.
- `Messenger` – slouží ke komunikaci tříd mezi sebou (nejčastěji si mezi sebou posílají zprávy třídy typu *ViewModel*. Vždy je jedna strana, která se registruje k přijímání určitého druhu zpráv, a druhá strana, která zprávy odesílá. Zpráv je několik předdefinovaných druhů nebo se vždy dají snadno vytvořit vlastní druhy. Jsou i různé způsoby, jak zasílání zpráv lépe cílit a koordinovat.
- `Barcode manager` – je třída, které se předá pouze nějaký obrázek, a ona se postará o rozpoznání čárového kódu. Vrací výsledek ve formě řetězce nebo vyvolá výjimku.
- `MockGPS` – je služba, které se dají nastavit GPS souřadnice cesty. Různé časování apod. a ona se chová jako reálný GPS modul. Je to výhodné při testování, kdy nám reálná data nemusí úplně vyhovovat nebo při běhu v emulátoru, kde GPS nefunguje vůbec.

Využití vestavěných aplikací (*Launchers and Choosers*)

Jedinou vestavěnou aplikaci, kterou jsem v programu použil, byla aplikace zastoupená třídou `CameraCaptureTask`, která zajišťuje focení při rozpoznávání čárového kódu. `CameraCaptureTask` je třída, která spustí aplikaci pro focení a po dokončení focení vyvolá událost, ve které předá výslednou fotku. Tento systém je ale poměrně nešikovný, protože celá akce trvá i několik sekund a i potom není jisté, zda se čárový kód dobře rozpozná. Pokud rozeznávání selže, celá akce se musí opakovat. V takovém případě, je mnohem rychlejší (ale datově náročnější) si potravinu vyhledat s pomocí našeptávače. Tento problém se vyřeší s příchodem aktualizace Mango, který zpřístupní přímo API kamery a bude tak možné provádět rozpoznávání několikrát za vteřinu bez spouštění pomocné aplikace. Tím se rychlost i spolehlivost mnohonásobně zvýší.

4.3.2 Komunikace

Protože se jedná o klientskou aplikaci, je třeba řešit komunikaci se serverem. Cena mobilního připojení se odvíjí od množství přenesených paketů, anebo je často množství paketů omezeno a proto hlavním cílem při návrhu komunikace bylo minimalizovat datovou náročnost aplikace.

Při plánování se nabízelo několik možností, jak vyřešit komunikaci:

- Navrhnout si vlastní datový protokol. Tato varianta by se dala udělat datově nejméně náročná. Mohla by se zahrnout i komprimace přenášených dat apod. Ovšem implementace by byla náročnější než další alternativy.
- Využití protokolu SOAP s pomocí Windows Communication Foundation. Tento protokol je zabudován do platformy Silverlight a dokonce je možné na klientovy automaticky vygenerovat *datovou proxy třídu*, která řeší všechny záležitosti komunikace. Tento protokol je však poměrně datově náročný a navíc je nativně přenášen pomocí objemného XML protokolu.
- Kompromis mezi datovou náročností a složitostí implementace, tvoří datová služba REST. Používá se hlavně pro vzdálené CRUD[9] operace. Tento protokol komunikuje skrze HTTP dotazy. Na rozdíl od řešení protokolem SOAP, musíme v tomto případě řešit skládání webového dotazu, zpracování odpovědi, případně zpracování chyb. Pokud REST komunikaci zakódujeme do formátu JSON (místo implicitního XML), přenáší se téměř už jen samotná data. WP7 podporuje formát JSON, ale funkce pro serializaci a deserializaci musíme volat ručně.

Srovnání datové náročnosti jednotlivých protokolů s konkrétní statistikou [18].

Cachování dat

Protokol REST se výborně hodí pro cachování, protože každý dotaz a odpověď jsou deterministicky určeny pouze konkrétní URL adresou. Taková data se dají ukládat do hashovací tabulky a pak podle klíče (URL adrasa) v nich rychle vyhledávat. Cachování se dá ještě optimalizovat podle toho, jaká data se nejčastětěji opakují a pak ponechávat v paměti jen ty nejoblíbenější dotazy.

V současné verzi není implementováno cachování dat. Tato funkce bude přidána hned do další verze.

Offline databáze

WP7 operační systém prozatím neposkytuje žádnou formu relační databáze. Perzistence dat se zajišťuje pouze pomocí XML souborů. Jejich načítání by však při spuštění nebo i za běhu aplikaci zpomalovalo, proto funkce offline databáze zatím není implementována. Funkce bude implementována, až po analýze nejčastějších uživatelských dotazů při betatestování a ostrém nasazení aplikace.

Po příchodu již několikrát zmiňované aktualizace Mango, by měla být SQL databáze přístupná a správa offline dat, tak bude daleko pohodlnější a rychlejší.

4.3.3 Server

V době psaní bakalářské práce, byl založen testovací server na webu www.aspone.cz běžící na doméně http://zhubni.aspone.cz. ASPone je poskytovatel hostingových služeb platformy ASP.NET. Pro testování jsem využil verzi hostingu zdarma. V této konfiguraci poskytuje server podporu 40MB prostoru, ASP.NET 4.0, MS SQL 2008 a webový server IIS 7.

[9] CRUD označuje 4 databázové operace – Create, Read, Update, Delete

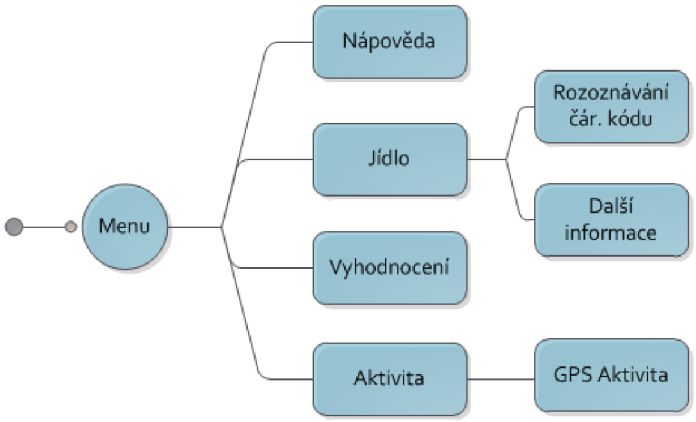
Na serveru se nachází databáze jídel, nutričních hodnot a dalších informací, které jsou přístupné přes veřejnou službu REST. Služba prozatím nevyžaduje autentizaci, která je ale naplánovaná hned v další verzi. Všechny funkce ve veřejném rozhraní serveru, které potřebují znát konkrétního uživatele, vyžadují v URL adrese dotazu uvést uživatelské jméno (zatím bez hesla).
Veřejné rozhraní serveru poskytuje následující funkce:

- Vyhledávání jídel obsahující předaný podřetězec.
- Dodatečné informace o jídle určeném přesným názvem jídla.
- Vyhledávání aktivit obsahující předaný podřetězec.
- Ohodnocení porce jídla indexem. Porce je dána názvem jídla, váhou, časem a uživatelem.
- Ohodnocení aktivity indexem. Aktivita je určená názvem, dobou trvání, intenzitou, pokročilostí uživatele a samotným uživatelem.
- Denní vyhodnocení uživatele. Dotaz je určen uživatelem a datem, ke kterému chceme získat vyhodnocení.

4.4 Grafické rozhraní

Grafické rozhraní bylo navrhováno hlavně z pohledu WP7 a jeho grafického stylu *metro*. Cílem bylo zachovat konzistentní vnímání uživatelského rozhraní jako v jiných aplikacích. Snažil jsem se využít co nejvíce předdefinovaných prvků a standardizované ikony pro WP7.

Struktura GUI je prozatím velmi základní. Byla navržena pro minimální funkčnost, aby se aplikace dala používat a testovat. Do ostrého nasazení ještě zbývá několik dalších cílů a verzí. Momentálně může uživatel zaznamenat jídla ručně, pomocí čárového kódu a může si o jídle zjistit ještě dodatečné informace. Sport lze zaznamenat ručně nebo pomocí GPS. Po zaznamenání všech činností přes den lze shlédnout výsledné hodnocení. Pokud by aplikace nebyla pochopena intuitivně, lze si přečíst nápovědu a získat potřebné znalosti k pohodlnému ovládání.



4.6 Struktura GUI

4.4.1 Menu

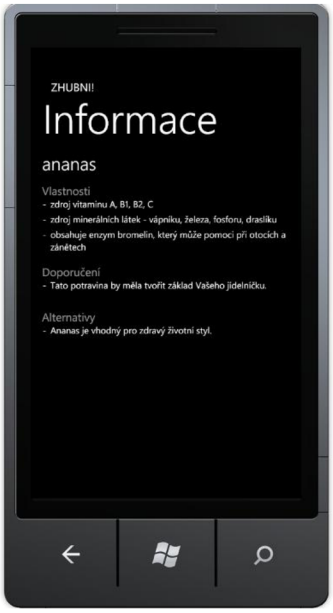
Menu by mělo být koncipováno jednoduše a přehledně. Snaha o zachování user experience se projevuje v tom, že tlačítka menu jsou navržena podle základního menu telefonu do tzv. dlaždic (angl. tiles).

Pozadí jsem se snažil vybrat takové, aby se rychle neokoukalo a zároveň, aby nepůsobilo rušivě.

4.7 Menu aplikace

4.4.2 Jídlo

Obr 4.9 Ruční výběr jídla



Obr 4.8 Více informací

Při vybírání jídla uživatel buď musí vyhledat název v databázi, nebo vyfotit čárový kód.

Dále se musí vyplnit ostatní údaje jako je váha porce (lze vybrat z předdefinovaných vah), datum a čas. Poté se může nechat aktualizovat index a jídlo zaznamenat do deníku.

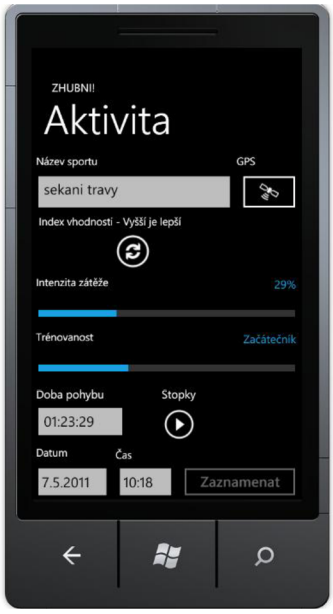
U většiny jídel je možnost se podívat na dodatečné informace – doporučení, alternativy, vlastnosti.

4.4.3 Aktivita

Aktivita se vybírá ze seznamu aktivit podle názvu. Pokud je aktivitu možné sledovat pomocí GPS, tlačítko pro tuto funkci se zpřístupní a po jeho použití se zobrazí obrazovka pro automatický záznam GPS aktivit. (Obr 4.10)

Při záznamu GPS by měla být přístupná mapa, ale bohužel nelze použít offline datové podklady. Z důvodu datové úspornosti proto zatím nebyla implementována.

Obr 4.10 Sledování aktivity přes GPS. Nejsou vyplněna reálná data.



Obr 4.11 Ruční zadávání aktivity

4.4.4 Vyhodnocení

Vyhodnocení je zaměřeno na maximální jednoduchost. Výpočty prováděné na serveru při vyhodnocování dat jsou poměrně složité, ale záměr je, neobtěžovat uživatele zbytečnými údaji. Pokud by se však při testování projevila nespokojenost s přílišnou jednoduchostí, není problém postup výpočtu

nějakým způsobem vizualizovat a více tak uživatele informovat. Fungovalo by to stejně jako u jídla, kde je tlačítko pro získání dalších informací.

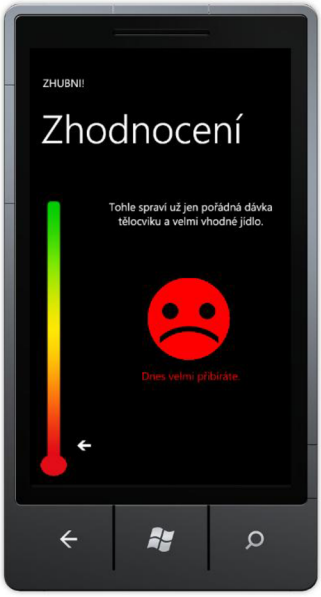
Vyhodnocení se provádí pomocí *zdravoměru*, což je jakýsi teploměr, vyjadřující zdravý/nezdravý styl života, kde uprostřed je neutrální stav. Pro příjemnější a přehlednější vyjádření stavu ještě zdravoměr doplňuje smajlík, který se snaží v uživateli vyvolat nebo umocnit odpovídající náladu. Když někdo zhřeší v jídle, většinou o tom ví. Cítí se hůř jak po psychické tak fyzické stránce. Často o tom ale nikomu neřekne a sám se s tím nějak srovná. Pokud ale uvidí mračící se telefon, možná to pomůže k tomu, aby se ho snažil rozveselit a tím zlepšit náladu i sám sobě. Do jaké míry tento přístup funguje, je otázka psychologie.

Smajlík a *zdravoměr* ještě doplňují komentáře textové. Je to zhodnocení dosahovaného pokroku a rady k dalšímu postupu.

Prozatím je 5 stavů pro smajlík i textové části, ale do budoucna je v plánu měnit komentáře dynamicky, přesně podle toho, co uživatel potřebuje a smajlík plynule animovat. Následující obrázky ukazují 3 z 5 dostupných stavů.

Obr. 4.14 Ukázka pozitivního hodnocení

Obr. 4.13 Ukázka neutrálního hodnocení



Obr. 4.12 Ukázka negativního hodnocení

4.4.5 Nápověda

Obr 4.15 Ukázka nápovědy

Nápověda obsahuje základní instrukce pro práci s programem. Není nijak rozsáhlá, protože funkce programu je koncipována pro co nejjednodušší ovládání.

Obsah nápovědy bude upraven po beta testování nebo v průběhu ostrého nasazení aplikace podle toho, s čím budou mít uživatelé největší problémy.

4.4.6 Zpracování chyb

Zpracování chyb je řešeno standardním WP7 chybovým dialogem. Takto se zpracovávají chyby téměř ve všech aplikacích pro WP7 (hlavně v těch nativních), takže uživatel nebude překvapen při zobrazení chyby či nějakého upozornění.

Obr 4.16 Oznamování chyb, upozornění apod.

4.5 Použité knihovny a postupy

Aplikace byla naprogramována s použitím několika knihoven třetích stran. Knihoven, které by byly pro projekt vhodné, je celá řada, ale někdy se jedná o tak komplexní řešení, že jen naučit se je zabere více času, než kolik času ušetří jejich použití. Z tohoto důvodu jsem použil zatím jen jednodušší knihovny funkcí a komponent, které nevyžadují téměř žádné studium. Do budoucna ale chystám apli-

kovat složitější knihovny, které automaticky řeší například cachování komunikace nebo rámec, který převádí asynchronní operace na zdánlivě synchronní (*Reactive Extensions*).

4.5.1 MVVM Light Toolkit

Autorem MVVM Light Toolkit [19] je Laurent Bugnion [20]. Tento framework se stal velmi oblíbený jak pro WP7, tak i pro Silverlight a WPF. Byl dokonce vybrán jako jedno z témat pro MIX 2011[10], což dokazuje jeho oblíbenost i rozšířenost.

Tento rámec je zaměřen na jednoduchost a univerzálnost. Nevyžaduje jakýkoliv nestandardní přístup nebo použití. Plně vyhovuje vzoru MVVM a usnadňuje jeho implementaci. Díky tomu, že je navržen tak, aby uvnitř samotného rámce bylo co nejméně vazeb, je možné použít jakoukoliv část samostatně, použít současně i jiný rámec nebo jej rozšířit ručně.

Z MVVM Light Toolkit je použito:

- `ViewModelLocator` – centrální místo pro získávání instancí tříd typu *ViewModel*. Hlavně se využívá jako zdroj pro *View*.
- `ViewModelBase` – abstraktní třída ze které dědí každý *ViewModel*. Implementuje `INotifyPropertyChanged` rozhraní, které umožňuje vytvořit vlastnost (*property*), která může sloužit jako zdroj pro vazbu (*binding*).
- `Messenger` – viz kapitola 4.3.1
- `RelayCommand` – Implementace standardního rozhraní `ICommand`. Používá se na spouštění a řízení činnosti tříd typu *ViewModel*. Bez této třídy se obvykle musí každý příkaz (*command*) třídy *ViewModel* založit jako samostatná třída implementující `ICommand`.
- `EventToCommandBehavior` – umožňuje převod jakékoliv události uživatelského rozhraní na příkaz ve třídách *ViewModel* přímo z XAML kódu. Velmi to ušetří psaní kódu v pozadí.
- Vzorové šablony pro projekty, části projektů a předdefinované, často používané části kódu (*code snippets*) usnadňují psaní stále stejných částí kódu. Šablona projektu například obsahuje `ViewModelLocator` a vzorovou stránku aplikace. Často používané části kódu jsou použitelné například při vytváření běžných vlastností (*properties*) tříd, které slouží pro vazby z *View*[11].

4.5.2 Location Mock

Při vývoji pro WP7 a používání GPS nastává problém v tom, že emulátor nepodporuje GPS modul, respektive nedá se jeho práce nijak simulovat. Vždy se zastaví ve stavu *čekám data*.

Řešení [21] nabídl Peter Torr na konferenci MIX10 a poskytl volně ke stažení zdrojové kódy. Jedná se o simulátor lokačních funkcí na emulátoru i reálném zařízení. Je to užitečné pro testování,

---

[10] MIX je prestižní shromáždění vývojářů, designérů, obchodních profesionálů atd., kteří vytvářejí nejinovativnější a nejlepší projekty publikované volně na internetu spjaté s technologiemi společnosti Microsoft [27].

[11] Vlastostni (property) tříd *ViewModel* k informování o své změně vyvolávají událost `PropertyChanged` definovanou v `INotifyPropertyChanged` rozhranní a deklarovanou ve třídě `ViewModelBase`.

protože lze nastavit vlastní trasu a časování. Služba také simuluje přechody do všech stavů jako na reálném zařízení apod.

Tato služba je navržena ale tak, že všechny třídy jsou pojmenovány stejně jako originální, jen jsou v jiném jmenném prostoru. Způsob použití se původně zamýšlel tak, že při vývoji by bylo definováno použití jmenného prostoru od Petera Torra a před finálním releasem, by se jmenný prostor změnil na originální. Takové řešení mi ale přijde poměrně krkolomné a těžkopádně, protože se musí upravovat již napsaný a odladěný kód před finálním vypuštěním, takže jsem si napsal převodní třídu `ToMixGeoLocationAdapter` podle návrhového vzoru adaptér[12] ve jmenném prostoru aplikace a při každém spuštění aplikace, tovární metoda rozhodne, zda se použije originální třída `GeoCoordinateWatcher` nebo `ToMixGeoLocationAdapter` podle toho, jestli kód běží na emulátoru nebo reálném zařízení. Obě třídy dědí rozhranní `IGeoPositionWatcher`, takže ve zbytku aplikace se pracuje právě s tímto rozhraním.

4.5.3 John Papa

John Papa je autor Silverlight TV a dosti výrazná osobnost publikující na téma Silverlight, WP7 a WPF. Při návrhu aplikace jsem se inspiroval jeho aplikací Contoso Book Shelf, kterou předvedl a vysvětlil na konferenci PDC 2010 [22]. Tato aplikace by měla sloužit jako základní učební kámen pro každého programátora pro platformu Silverlight. Aplikace je uzpůsobena k testování, poskytuje velkou *blendabilitu*[13] a data v grafickém režimu (*design time data*) a ukazuje princip *Separation of Concerns*[14].

Z této aplikace jsem převzal hlavně třídu `Navigator`. V jeho aplikaci se jmenuje `PageConductor` a stará se o navigaci a předávání objektů mezi stránkami a ukládání stavů.

Celkově jsem se však snažil použít i jeho programátorský styl, což je vzorová ukázka čistého, čtivého a samodokumentujícího zdrojového kódu.

4.5.4 Windows Phone 7 Design Templates

Jedná se o soubor ovládacích prvků a ikon vytvořených podle pravidel WP7 metro designu a zachovávající na nejvyšší míru jednotný grafický styl WP7 [23].

Tento balík obsahuje Photoshop soubory skládající se z vrstev, jež jsou připravené ke snadnému kopírování, takže vytvořit návrh WP7 aplikace ve Photoshopu je jen záležitostí pár copy & paste operací.

Zároveň obsahuje XAML soubory, které jsou složeny z ovládacích prvků, které mají nadefinované přesné rozměry a rozložení (mezery, popisky, barva, …) tak, aby vyhovovaly WP7 grafickému stylu. Ovládací prvky jsou vloženy do kontejnerů (např. *stack paneů* nebo *grid*, což umožňuje vzít celý kontejner a jednoduše ho vykopírovat i s obsaženým ovládacím prvkem a vložit ho do svého projektu. Takový postup zachová veškeré formátování a rozložení. Pokud se pracuje v programu Ex-

---

[12] Adaptér umožní spolupráci třídám, které by vzhledem k rozlišným rozhraním jinak spolupracovat nemohly (definice převzata z [25]).

[13] Blendabilita je pojem, který vyjadřuje míru přizpůsobení nástroji blend (viz kapitola 4.8).

[14] Separation of Concerns (SoC) je proces rozdělení programu do co nejmenších nezávislých jednotek, mající zodpovědnost jen za jednu funkcionalitu. Toho se dosahuje zejména skrze modularitu a zapouzdření.

pression Blend, tak tento postup je velmi podobný postupu s kopírováním vrstev v programu Photoshop.

Balík obsahuje i ikony, které jsou v různých variantách rozděleny podle toho, na jakých místech se mohou použít, nebo podle toho, jaké barevné schéma je nastaveno na telefonu.
Varianty ikon:

- S kroužkem nebo bez něj.
- V provedení pro téma s černým nebo bílým pozadím.

4.5.5 Coding4Fun Windows Phone Toolkit

Tento balík obsahuje velmi užitečné komponenty a funkce pro každou WP7 aplikaci. Obsahuje hlavně dosti zajímavé ovládací prvky, které by, podle mého názoru, měly být nativně součástí WP7 SDK nebo alespoň Silverlight Toolkit for Windows Phone 7[15].

V této práci jsem prozatím využil jen `TimeSpanPicker` pro vybírání časového období. Na rozdíl od oficiální implementace, tato verze umožňuje vybrat i vteřiny. Použita je pro výběr doby trvání aktivity.

Plánuje se využít více prvků z tohoto balíku. V další verzi jsou už zapracovány například `MessagePrompt` komponenty, které nahrazují současnou implementaci ošetření chyb a upozornění (viz kapitola 4.4.6). `MessagePromopt` z tohoto balíku se výrazně lépe konfiguruje, dají se nastavovat různá tlačítka apod. V implicitním `MessageBoxu` lze použít jen tlačítka *Ok* a *Cancel*.

4.5.6 ZXing Barcode Scanning Library

První knihovna pro rozpoznávání čárového kódu, která byla vydána pro WP7 je Windows Phone 7 Silverlight ZXing Barcode Scanning Library [24]. Jedná se o port knihovny zxing od Googlu, která byla původně napsána v Javě.
Knihovna umí rozpoznávat následující formáty:

- UPC-A and UPC-E,
- EAN-8 and EAN-13,
- Code 39,
- Code 128,
- QR Code,
- ITF.

Autor této knihovny ponechal originální jádro jako samostatnou knihovnu, navíc pak přidal ovládací třídy pro větší jednoduchost. Kvůli dosažení větší efektivnosti jsem ale těchto rozšíření nevyužil.

4.6 Implementace

Díky metodikám použitých při návrhu a plánovaní popsaných v kapitole Návrh 4.2, jsem se při implementaci nedostal do žádných výrazných problémů nebo slepých uliček.

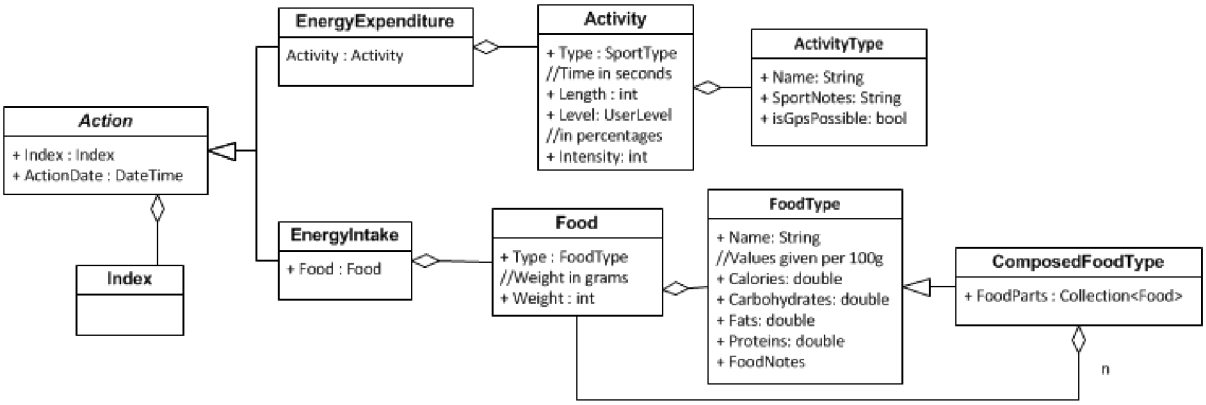
[15] Silverlight Toolkit for Windows Phone 7 je oficiální rozšíření standardních ovládacích prvků pro WP7.

Při psaní programového kódu jsem se snažil držet osvědčených postupů, jako jsou návrhové vzory (*design patterns* viz [25]) a zásady čistého kódu (*clean code* viz [15]).

V následujících kapitolách, rozebírajících implementaci, se neobjevuje popis všech tříd. Z důvodu přehlednosti jsou některé jednoduší a nedůležité implementační záležitosti vynechány. Stejně tak je vynechán popis lokálních pomocných tříd, které jsou již popsány v kapitolách 4.3.1 a 4.5.

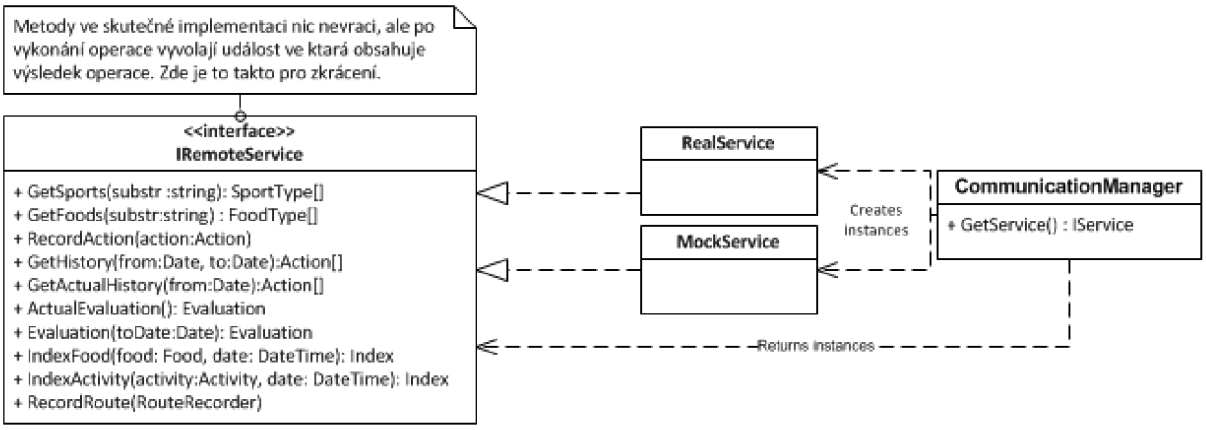
4.6.1 Model

V části *Model* bylo nejdůležitější dobře rozvrhnout reprezentaci jídla a aktivity. To se mi, myslím povedlo, protože během celého návrhu jsem strukturu skládání a dědění nemusel měnit.



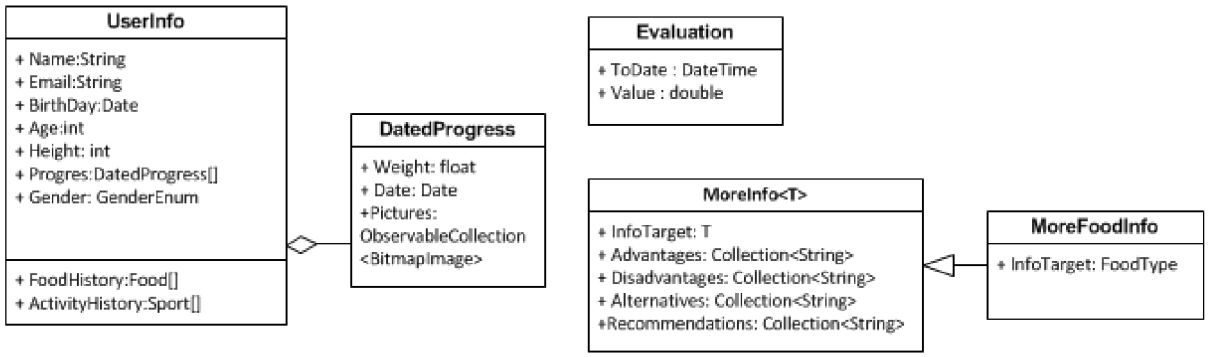
Obr 4.17 Diagram tříd pro implementaci aktivity a jídla.

Další zajímavou částí struktury je rozhraní služby a její implementace. Třída *CommunicationManager* spravuje instance reálných implementací rozhraní (vzor továrna).



Obr 4.18 Aplikace vzoru továrna pro rozhraní služby.

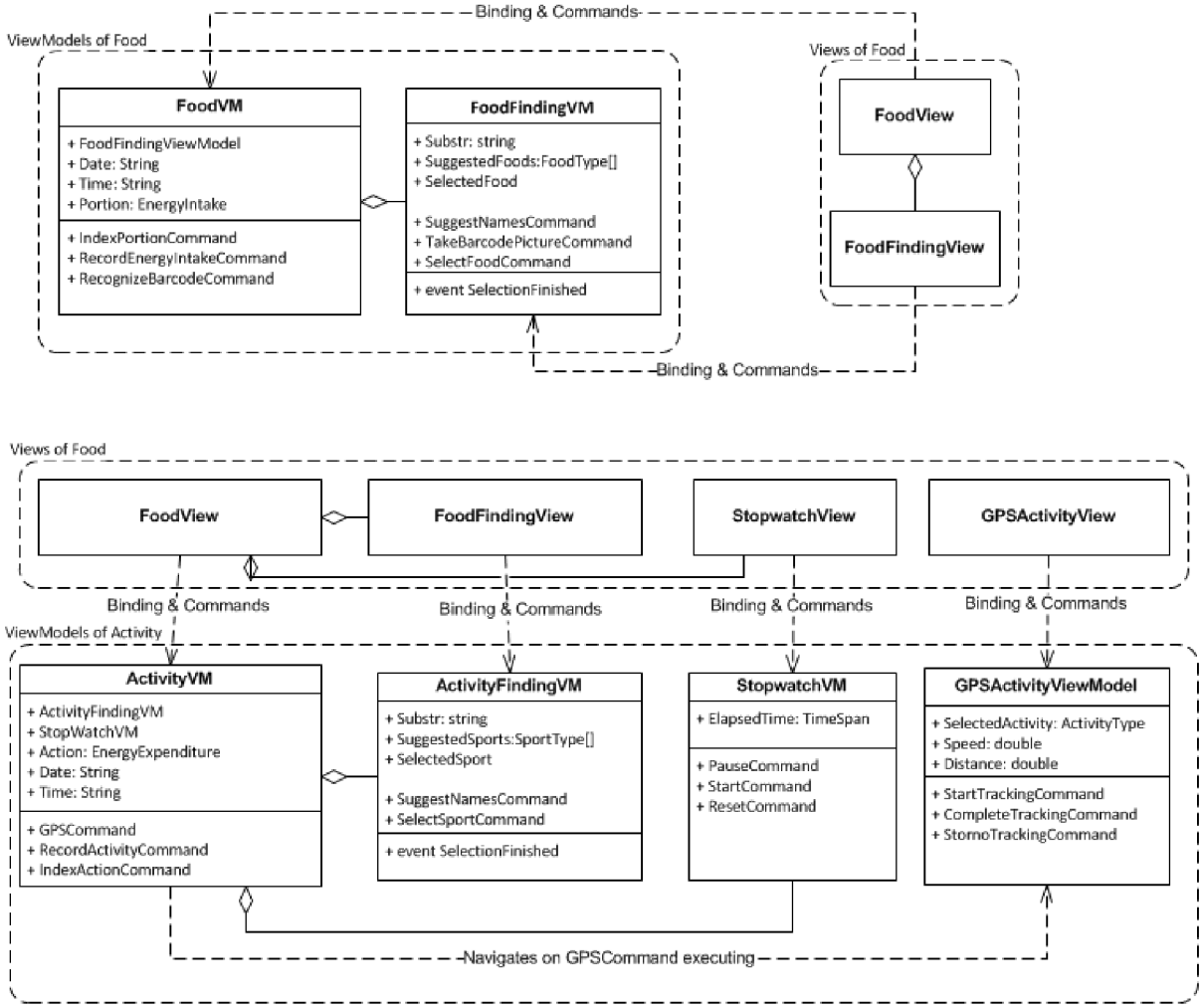
Další položky modelu jsou již jen přepravky pro data.



Obr 4.19 Ostatní třídy reprezentující data programu.

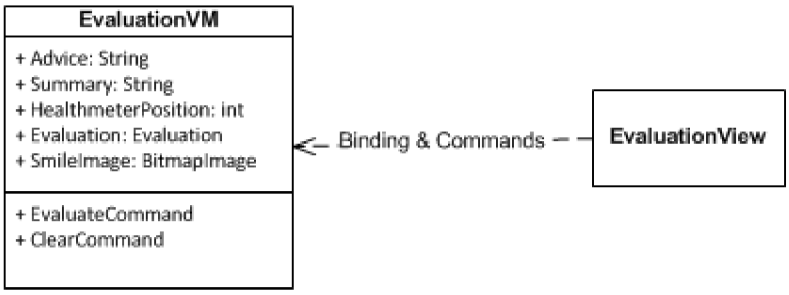
4.6.2 View – ViewModel

Nejsložitější vazby se objevují pro *ViewModel* jídla a aktivity. Je použito skládání jak pro *View* tak pro *ViewModel.*



Obr 4.20 Skládání View - ViewModel u jídla a aktivity.

Poslední třída *ViewModel* zbývá `EvaluationVM` a k ní `EvaluationView`. Tam se žádné skládání ani složitější vazby nevyskytují. U této třídy byl problém jen s univerzálním vyjádřením polohy zdravoměru, což jsem vyřešil přes vyjádření v procentech – tedy neutrální stav je 50% na stupnici, nejhorší je 0% a nejlepší je 100%. *View* si pak hodnotu zobrazí přesně tak, jak potřebuje.



Obr 4.21 ViewModel a View pro vyhodnocení.

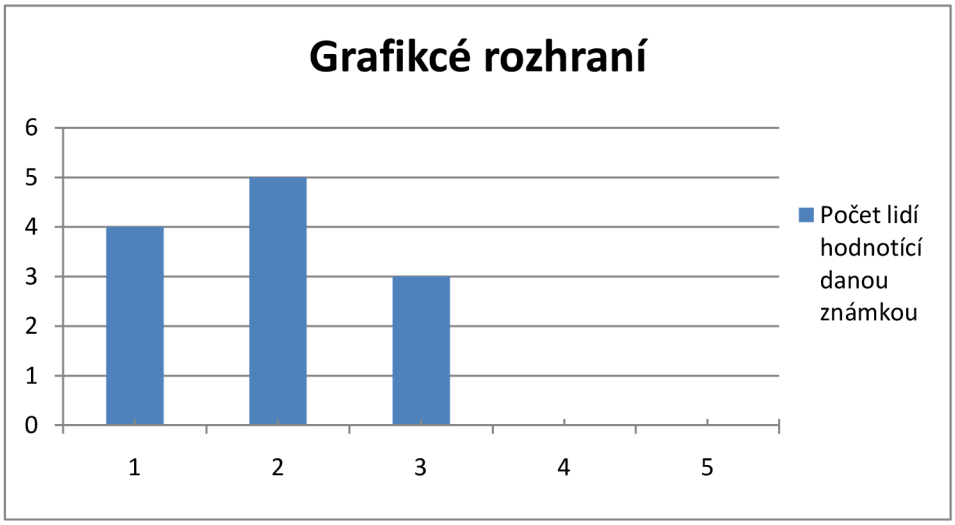
4.7 Uživatelské testy

Aplikaci jsem poskytl k testování dvanácti respondentům. Jednalo se o potencionální uživatele aplikace. Věk zúčastněných se pohyboval od 16 do 50 let. Při testování se hodnotilo grafické rozhraní, pohodlnost ovládání a míra užitečnosti.

Hodnocené skupiny se známkují jako ve škole.

Grafické rozhraní

Zpracování grafiky a přehlednost uživatelského rozhraní se většině líbila – v aplikaci nejsou potřeba výraznější změny. Jediné co chybělo, bylo znázornění zaneprázdnění aplikace při časově náročnějších operacích (např. při vzdálených požadavcích do databáze). To už ale v plánu je do další verze.

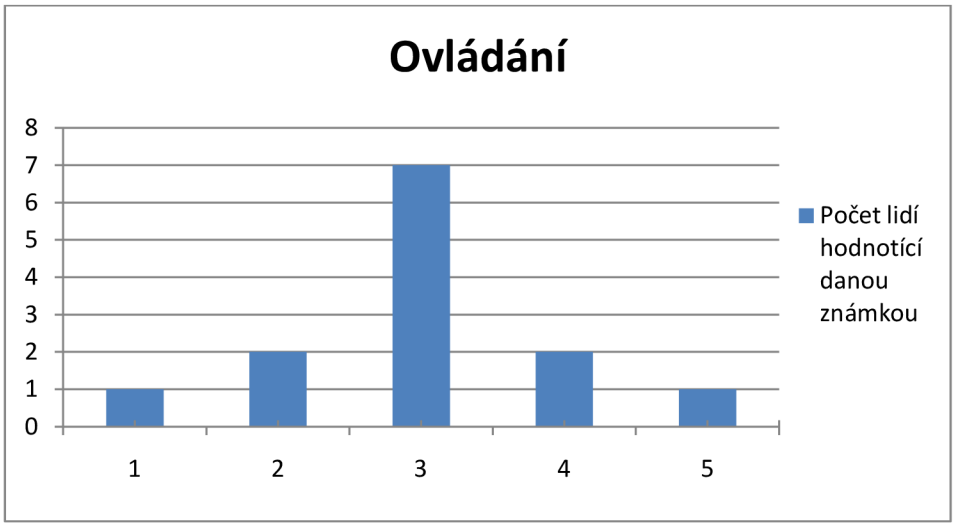


Obr 4.22 Uživatelské hodnocení grafikcého rozhraní. Známkování jako ve škole.

Pohodlnost ovládání

S ovládáním měli největší problém starší lidé, kteří neměli žádné zkušenosti s dotykovými telefony, a trochu i lidé pro který byl WP7 nový. To se (předpokládám) vyřeší v době, kdy si tuto aplikaci budou kupovat zákazníci, kteří daný přístroj znají, a ovládání jim pak bude připadat známé.

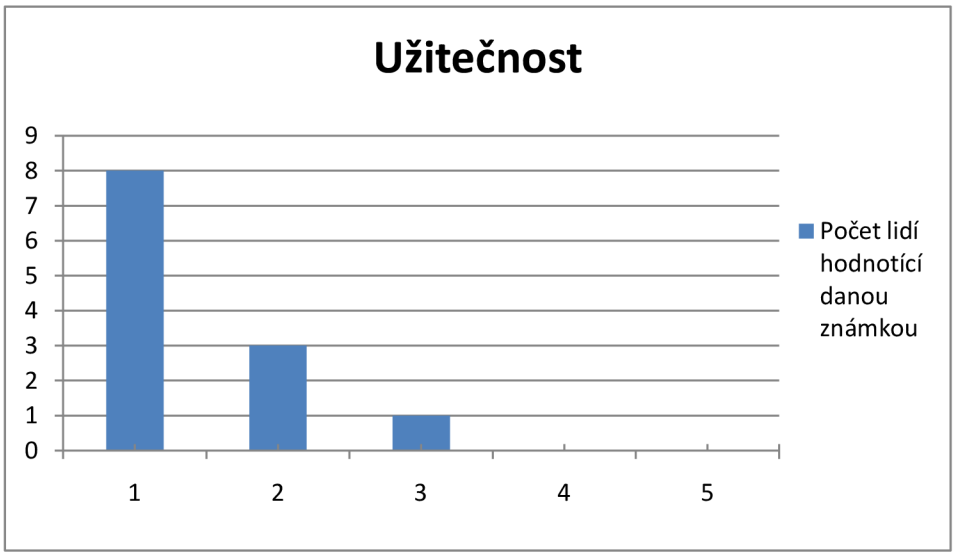
Obecně ale každý měl problém se záznamem jídla a aktivity. Velmi neintuitivní se stalo obnovování hodnocení indexu, což se neděje automaticky, ale musí se vždy obnovit ručně. Je to z důvodů datových úspor komunikace, ale tento úkon se bude muset přepracovat.



Obr 4.23 Uživatelské hodnocení ovládání. Známkování jako ve škole

Užitečnost

V této oblasti se všichni shodli na tom, že aplikace by mohla být reálně nasazená a největší přínos by měla jako rádce při nákupech. Navíc 50% (6 uživatelů) by se svěřilo aplikaci v rámci hubnutí.



4.8 Použité nástroje

Při práci na projektu jsem používal nástroje pro vývoj, návrh a plánování. Pro vývoj to byla integrovaná vývojová prostředí Microsoft Visual Studio 2010 nebo Expression Blend 4. Pro plánování a návrh FreeMind, MiniSCRUM a pro zálohování TortoiseSVN.

4.8.1 Visual Studio 2010, ReSharper, Emulator

Většinu času jsem při vývoji strávil v prostředí Microsoft Visual Studio 2010 (dále jen VS) a projekt jsem testoval na emulátoru WP7 operačního systému. VS v základní verzi nemá mnoho funkcí pro pohodlné psaní zdrojového kódu, ale existuje obrovské množství různých pluginů, které se dají doinstalovat.

Osobně mohu VS srovnávat pouze s NetBeans, které má více funkcí v základní verzi, možná i lepší doplňování zdrojového kódu a některé další věci, které mi na VS chybí, ale spotřebuje více systémových prostředků a je pomalejší než VS. Navíc v NetBeans pravděpodobně nebude podpora pro vývoj Silverlight aplikací a automatické nasazování aplikace do WP7 emulátoru nebo přímo do přístroje přes Zune software. VS se stává naprosto bezkonkurenčním prostředím po nainstalování rozšíření ReSharper [26].

ReSharper

Tento plugin pomáhá dosahovat maximální produktivity a efektivity. Je plně integrován do VS a mnohonásobně rozšiřuje jeho základní schopnosti.
Největší přínosy pluginu ReSharper:

- Zvýrazňuje chyby, upozornění a doporučení v kódu za běhu a téměř vždy nabídne automatické řešení daného problému. Řešení je možné vyvolat pomocí jediné klávesové zkratky, takže úprava nevhodně napsaného kódu je otázka vteřiny.
- Nabízí přes 30 pokročilých způsobů refaktorování kódu. Dokáže najít a změnit všechny závislosti ve všech souborech. Dokonce i v textových řetězcích nebo podobných proměnných, jejichž názvy jsou závislé.
- Poskytuje velmi rychlé navigační a vyhledávací funkce. Přes pár klávesových zkratek se dá velmi rychle pohybovat napříč celým projektem.
- Automaticky formátuje kód při odřádkování nebo dokončení bloku kódu.
- Automaticky generuje kód podle předem daných šablon.
- Mnoho dalších funkcí (viz. [26]).

Na druhou stranu, takto komplexní vylepšení celé VS mírně zpomalí, což je znatelné nejvíce při otevírání projektu, kdy ReSharper analyzuje kód a vytváří vazby. IDE se určitě nezpomalí na úroveň NetBeans ale produktivita výrazně stoupne.

Emulátor

Velmi dobrý nástroj, který poskytl Microsoft pro vývoj WP7 aplikací. Odpovídá přesně běhovému prostředí na reálném zařízení, pouze některé funkce jsou omezené. Jak už jsem zmínil v kapitole 4.3.1, nefunguje kamera a GPS.

Při spuštění aplikace na emulátoru a vyvolání funkce pro rozpoznávání čárového kódu, se vyvolá obrazovka se seznamem předdefinovaných obrázků, ze kterého se může vybrat a tím se nahradí reálné focení čárového kódu.

Funkce GPS je naopak simulována pomocí `LocationMock` (viz kapitola 4.5.2).

4.8.2 Expression Blend

Expression Blend je grafický editor pro vytváření grafického uživatelského rozhraní pro Siliverlight, WPF nebo pro WP7 aplikace. Tento editor díky své jednoduchosti, ale zároveň mimořádné propracovanosti nemá konkurenci.

Jeho hlavní výhoda je, že dokáže snadno a rychle vytvořit kvalitní grafiku pro jakoukoliv aplikaci postavenou na výše zmíněných technologiích. Umí velmi jednoduše vytvářet animace. Stačí jen na časové ose vybrat polohu a vybranému atributu nastavit vlastnost. Potom vybrat jiné místo v čase a

opět nastavit hodnotu. Je možnost ještě nastavit nějaké parametry a celá animace je hotová. Zvládá jednoduše i 3D transformace a jiné složitější úkony.

Je možné mít spuštěný Blend i VS najednou. V Blendu se starat o grafickou část psanou v XAML (veškerá grafika, vazby, příkazy, animace, chování (*behaviors*) atd.) a ve VS obstarávat všechen kód v C# nebo Visual Basicu jazycích. Stačí pak jen přecházet mezi okny editorů a projekt se automaticky aktualizuje. Tím, že se využijí nejsilnější stránky každého editoru, se práce velmi urychlí.

Další velmi silnou stránkou (o které ne každý nováček ví) je, že Blend umí zobrazovat data v režimu grafického návrhu (*design time data)* a nejenom to, on je umí i sám generovat. Jak dosahovat v projektu vysoké míry přizpůsobení Blendu z pohledu dat v grafickém režimu, popisuje John Papa v [22]. Pokud nechceme ale do projektu zanášet data pro Blend, můžeme si data v Blendu vytvořit. Vybereme třídu, kterou chceme naplnit daty a Blend vygeneruje XAML soubor, který data vytvoří z generátorů – to jsou propracovanější *Lorem Ipsum generátory* a je možné jim nastavovat počet slov, délku slov, nebo se může jedna to číselné hodnoty apod. Grafik tak může rovnou vidět, jak bude aplikace vypadat po naplnění daty. Taková data budou viditelná jen v design režimu (v Blendu) a po spuštění aplikace se použijí reálná data.

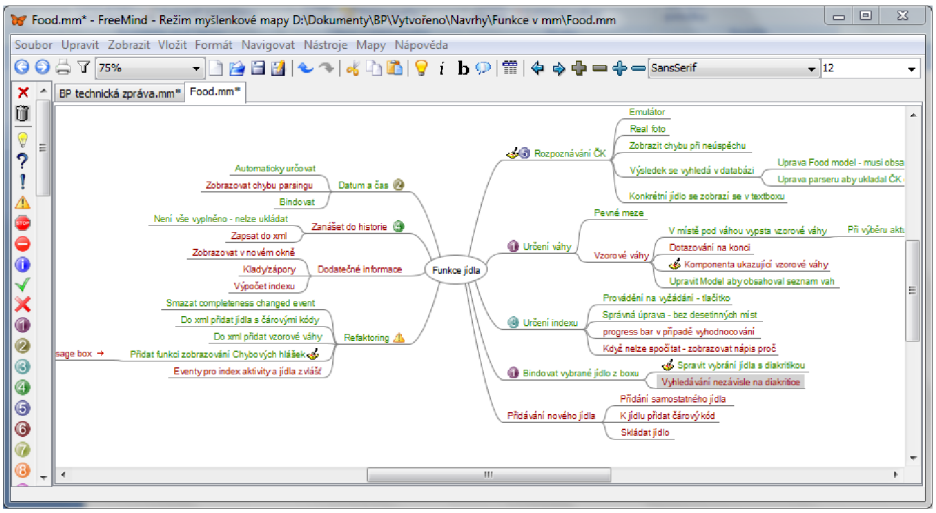
4.8.3 Ostatní nástroje

Ostatní nástroje mi neslouží přímo k vývoji aplikací, ale spíše pro podporu návrhu apod.

FreeMind

Myšlenkové mapy mi značně usnadňují záznam nápadů, myšlenek a poznámek oproti klasickému sekvenčnímu zápisu do seznamu. Pracoval jsem v editoru Freemind, což je jednoduchý, ale velmi intuitivní a rychlý editor, který splňuje všechny mé požadavky.

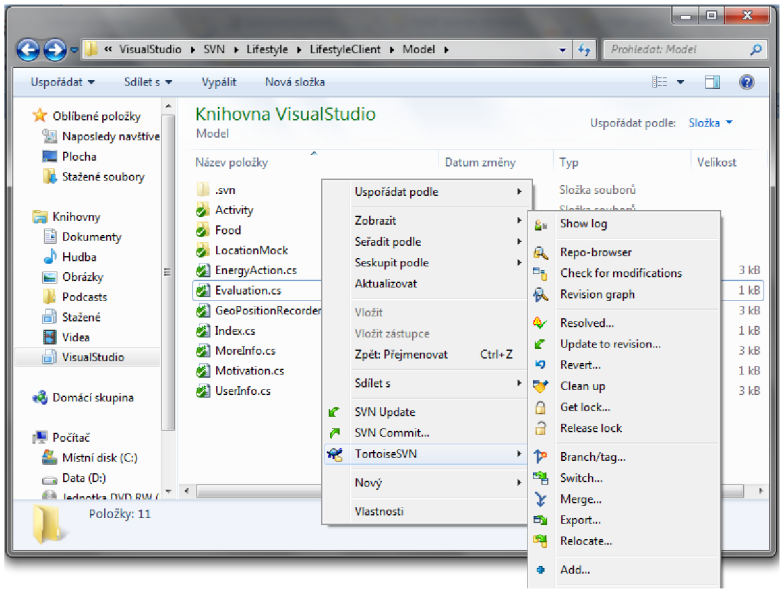
Myšlenkové mapy jsem používal při plánování v rámci PDCA (viz kapitola 4.2). Když jsem vybral vylepšení nebo další krok, který budu implementovat, začal jsem všechny myšlenky zaznamenávat do mapy. Zaznamenával jsem vše, co a jak bude potřeba udělat a také poznámky, kde by mohl být problém apod. – to odpovídá fázi *Plan*. Na začátku jsem si označil všechny položky/nápady červeně a pak už jsem jen implementoval jednu po druhé a průběžně značil hotové položky zelenou barvou. Někdy jsem používal i priority, abych zachoval posloupnost implementace, když jsem přerušoval práci. Při přerušení práce stačil jen velmi rychlý pohled na mapu a celá hlavní koncepce nově implementovaného rozšíření se mi dostala hned do hlavy. Díky tomu jsem mohl práci přerušovat častěji, aniž by výrazněji utrpěla produktivita.



Obr 4.24 Ukázka myšlenkové mapy v programu freemind. Zelené položky jsou splněné a červené ještě čekají na vyřešení

TortoiseSVN

TortoiseSVN je pravděpodobně nejznámější a nejrozšířenější SVN klient pro Windows. Je zakomponován přímo do kontextového menu v exploreru, takže odpadá potřeba se učit ruční příkazy. Je to velmi rychlé a přehledné. Jen občas se stane, že se pracovní kopie poškodí a potom nelze synchronizovat se serverem. S detailní znalostí příkazů, by se pravděpodobně problém dal snadno vyřešit, ale protože jsem se tomuto tématu více nevěnoval, bylo někdy nutné provést nový checkout celého projektu a při úpravách pak postupovat opatrněji.



Obr 4.25 Ukázka integrace TortoiseSVN do průzkumníka Windows.

MiniScrum

MiniScrum je online služba zdarma pro plánování metodou Scrum. Je jednoduchá, rychlá a přehledná. Pro mě osobně úplně dostačující, ale umím si představit, že s větším týmem a složitějším vývojem by tento nástroj nebyl dostatečný a bylo by nutné použít nějaký třeba i komerční nástroj.

Tato služba je provozována na adrese http://miniscrum.com/.

4.9 Vize do budoucna

V této práci je ukázána první verze aplikace, ale vývoj bude pokračovat i po odevzdání bakalářské práce. Již je připraven marketingový plán, je rozmyšlený postup jak oslovit velké partnery, dietologické poradny a další potencionální kandidáty na spolupráci.

Hlavní cíl do budoucna je naplnit databázi daty, která budou kompletně pokrývat český trh s potravinami a zároveň ke každé potravině budou dostupné informace pro rozhodování o její vhodnosti pro zdravý životní styl. Kromě čistého plnění databáze daty je nutné zapracovat na vzorcích pro výpočet hodnocení potravin, aktivit i denních souhrnů.

Po příchodu updatu Mango je nutné zapracovat co nejvíce nové funkčnosti a vydat druhou verzi aplikace.

Všechna vylepšení vedoucí k co největší uživatelské spokojenosti jsou hlavní prioritou. Jedná se především o skládání oblíbených jídel, učení aplikace a přizpůsobování se danému uživateli. Nejdůležitější bude samozřejmě spokojený zákazník.

Po těchto akcích je v plánu přidat sociální funkce a propojit aplikaci s facebookem, pořádat různé hry a soutěže k nalákání co největšího počtu zákazníků a zviditelnění aplikace.

Současně s těmito vylepšeními je nutné podnikat marketingové kroky a snažit se program dostat do médií, najít sponzory apod.

Z důvodů konkurence nebudu rozšíření více rozebírat.

5 Závěr

Vytvořil jsem aplikaci, která využívá moderní mobilní technologie – v současné době nejnovější platformy Windows Phone 7. Navíc její hlavní záměr je pomoci lidem řešit problémy s obezitou, kondicí a vzhledem.

V této práci jsem popsal postup při vývoji od obecného úvodu do problému s nezdravým způsobem života, přes popsání technologie WP7 až po konkrétní architekturu aplikace, použité knihovny a programy při vývoji.

Vývoj pro WP7 je snadný, rychlý a jediné překážky, na které jsem narazil, vyplívají ze Silverlight architektury a to především izolace aplikací. V případě této konkrétní aplikace se ale nejedná o nijak zásadní problém. Jiné problémy souvisí hlavně s tím, že WP7 je úplně nová platforma a nedosahuje zatím takových možností jako ostatní. To by se mělo ve velmi krátkém čase změnit a poté si myslím, že zařízení s WP7 se stanou velmi oblíbenými. To vyplívá i z odhadů společnosti Gartner, která v roce 2015 předpovídá, že WP7 bude oblíbenější než iPhone. Na Android ale prý nedosáhne.

Bylo vzrušující zkoumat takto mladou platformu i vyvíjet přínosnou aplikaci tak velkému počtu uživatelů. Věřím, že až aplikace vstoupí na český trh, bude se jí dařit i po finanční stránce.

Bibliografie

1. **Francesco Branca, Haik Nikogosian, Tim Lobstein.** World health organization. [Online] [Citace: 14. 5. 2011.] http://www.euro.who.int/en/what-we-do/health-topics/disease-prevention/physical-activity/publications/2007/challenge-of-obesity-in-the-who-european-region-and-the-strategies-for-response-the.

2. **Ústav zdravotnických informací a statistiky ČR.** Evropské výběrové šetření o zdravotním stavu v ČR - EHIS CR. [Online] 2010. [Citace: 14. 5. 2011.] http://www.uzis.cz/rychle-informace/evropske-vyberove-setreni-zdravotnim-stavu-cr-ehis-cr-index-telesne-hmotnosti-fyzic.

3. **Roche s.r.o.** Obezita v ČR i ve světě. *Obezita.cz.* [Online] [Citace: 14. 5. 2011.] http://www.obezita.cz/obezita/v-cr-a-ve-svete/.

4. **zun.** Každý pátý Čech je už obézní. *Novinky.cz.* [Online] [Citace: 14. 5. 2011.] http://www.novinky.cz/zena/zdravi/218125-kazdy-paty-cech-je-uz-obezni.html.

5. **Mineralfit.cz.** Light potraviny – škodí, nebo pomáhají? *MineralFit.* [Online] [Citace: 14. 5. 2011.] http://www.mineralfit.cz/domaci-lekar-clanek/light-potraviny-skodi-nebo-pomahaji-1012.

6. **Microsoft.** Msdn. *Push Notifications Overview for Windows Phone.* [Online] [Citace: 14. 5. 2011.] http://msdn.microsoft.com/en-us/library/ff402558(v=vs.92).aspx.

7. —. Execution Model Overview for Windows Phone. *Msdn.* [Online] [Citace: 14. 5. 2011.] http://msdn.microsoft.com/en-us/library/ff817008(v=vs.92).aspx.

8. —. Differences Between Silverlight and Silverlight for Windows Phone. *MSDN.* [Online] [Citace: 14. 5. 2011.] http://msdn.microsoft.com/en-us/library/ff426930(VS.95).aspx.

9. —. Launchers and Choosers Overview for Windows Phone. *MSDN.* [Online] [Citace: 14. 5. 2011.] http://msdn.microsoft.com/en-us/library/ff769542(v=vs.92).aspx.

10. —. Přehled systému Windows Phone 7. [Online] [Citace: 14. 5. 2011.] http://www.microsoft.com/cze/windowsphone/meet/windows-phone-7.aspx.

11. **Griffiths, Grant.** *Posilování pro muže.* Praha : Fragment, 2006. ISBN 80-253-0223-7.

12. **Austrac.** Continuous improvement. *Austrac e-learning.* [Online] [Citace: 14. 5. 2011.] http://www.austrac.gov.au/elearning_amlctf_programcourse/mod5/module_5_implementing_23.html.

13. **Šochová, Zuzana.** *Zuzi's blog.* [Online] [Citace: 14. 5. 2011.] http://soch.cz/blog/.

14. **Beck, Kent.** *PROGRAMOVÁNÍ ŘÍZENÉ TESTY.* Praha : GRADA, 2004. ISBN: 80-247-0901-5.

15. **Martin, C. Robert.** *Čistý kód.* Praha : COMPUTER PRESS, 2009. EAN 9788025122853.

16. **Llopis, Noel.** When Is It OK Not To TDD? *Games from Within.* [Online] [Citace: 14. 5. 2011.] http://gamesfromwithin.com/when-is-it-ok-not-to-tdd.

17. **Dajbych, Václav.** MVVM: Model-View-ViewModel. *dajbych.net.* [Online] 21. 4 2009. [Citace: 14. 5. 2011.] http://dajbych.net/model-view-viewmodel.

18. **Tiffany, Rob.** Building a WCF REST + JSON Service. *Windows Phone 7 Line of Business App Dev.* [Online] [Citace: 14. 5. 2011.] http://robtiffany.com/windows-phone-7/windows-phone-7-line-of-business-app-dev-building-a-wcf-rest-json-service.

19. **Bugnion, Laurent.** *MVVM Light Toolkit.* [Online] [Citace: 14. 5. 2011.] http://www.galasoft.ch/mvvm/.

20. —. *GalaSoft Laurent Bugnion.* [Online] [Citace: 14. 5. 2011.] http://www.galasoft.ch/.

21. **Torr, Peter.** Mock Location APIs from my Mix10 Talk. *Peter Torr's Blog.* [Online] [Citace: 14. 5. 2011.] http://blogs.msdn.com/b/ptorr/archive/2010/03/25/mock-location-apis-from-my-mix10-talk.aspx.
22. **Papa, John.** Kung Fu Silverlight at PDC 2010: Patterns and Practices with MVVM and RIA Services. [Online] [Citace: 14. 5. 2011.] http://johnpapa.net/pdc10kungfu.
23. **Microsoft.** Windows Phone 7 Design Templates. *CodePlex.* [Online] [Citace: 14. 5. 2011.] http://wp7designtemplates.codeplex.com/.
24. **Bray, Greg.** Windows Phone 7 Silverlight ZXing Barcode Scanning Library. *CodePlex.* [Online] [Citace: 14. 5. 2011.] http://silverlightzxing.codeplex.com/.
25. **Pecinovský, Rudolf.** *Návrhové vzory.* Brno : Computer Press, a. s., 2007. ISBN 978-80-251-1582-4.
26. **JetBrains.** *ReSharper.* [Online] [Citace: 14. 5. 2011.] http://www.jetbrains.com/resharper/index.html.
27. **Microsoft.** *MIX 2011.* [Online] [Citace: 14. 5. 2011.] http://www.microsoft.com/events/mix/.
28. **Petzold, Charles.** *Programming Windows Phone 7.* Redmond, Washington : Microsoft Press, 2010. ISBN: 978-0-7356-4335-2.

Seznam příloh

Příloha 1. CD/DVD obsahující:

- projekt pro Microsoft Visual Studio 2010 včetně zdrojových textů a potřebných knihoven pro překlad;
- návod na otevření projektu, kompilace projektu ve Visual Studiu 2010 a spuštění na emulátoru i reálném zařízení;
- tuto technickou zprávu ve formátu PDF;
- plakát propagující aplikaci.

Příloha 2. Tištěný plakát propagující aplikaci.

Příloha č. 2

Hubni
pomocí
moderních
technologií
Windows
phone
Zhubni!
Nová mobilní aplikace pro WP7