

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## STUDIE UŽIVATELSKÉHO ROZHRANÍ PRO ZADÁVÁNÍ JÍDEL

BAKALÁŘSKÁ PRÁCE

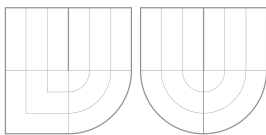
BACHELOR'S THESIS

AUTOR PRÁCE

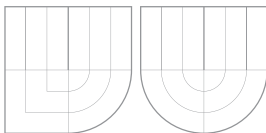
AUTHOR

MARIE KRATOCHVÍLOVÁ

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ



FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# STUDIE UŽIVATELSKÉHO ROZHŘANÍ PRO ZADÁVÁNÍ JÍDEL

STUDY OF USER INTERFACE FOR IDENTIFICATION OF MEALS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARIE KRATOCHVÍLOVÁ

VEDOUCÍ PRÁCE

SUPERVISOR

Doc. Ing ADAM HEROUT, Ph.D.

BRNO 2014

## Abstrakt

Tato práce dokumentuje vývoj aplikace pro jednoduchý a intuitivní výběr a zadávání jídel. Práce přibližuje jednotlivé etapy vývoje aplikace od analýzy existujících aplikací, přes specifikaci požadavků potencionálních uživatelů a návrhů prototypů rozhraní či samotnou implementaci až po testování. Popisuje výběr technologie pro implementaci aplikace. Výsledkem práce je elegantní, rychlá a intuitivní aplikace.

## Abstract

This thesis describes the development of mobile application for Windows Phone. Application is focused on choice of meals and identification of meals. The work contains description the different stages of development, starting with analysis of existing applications, specification of the customer requirements and design prototype interface and finally to its implementation and testing. The work describes the choice of technology for the implementation. Result of my work is an elegant, fast and intuitive application.

## Klíčová slova

Windows Phone, C#, Entity Framework, LINQ, mobilní aplikace, mobilní platforma, mobilní zařízení, Windows Store, WPF, UX, Uživatelské rozhraní, GUI, prototypy, Mockup, návrhové vzory, MVC, Observer, etapy vývoje, životospráva, zadávání jídel, výběr jídel, iterativní vývoj

## Keywords

Windows Phone, C#, Entity Framework, LINQ, Mobile Application, Mobile Platform, Mobile Device, Windows Store, WPF, UX, User Experience, UI, User interface, GUI, prototype, Mockup, Design Patterns, MVC, MV-VM, Observer, Stage of development, Diet, Identification of Meals, Choice of Meals, Iterative Development

## Citace

Marie Kratochvílová: Studie uživatelského rozhraní pro zadávání jídel, bakalářská práce, Brno, FIT VUT v Brně, 2014

# Studie uživatelského rozhraní pro zadávání jídel

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracovala samostatně pod vedením pana Doc. Ing. Adama Herouta Ph.D.

.....  
Marie Kratochvílová  
13. května 2014

## Poděkování

Především bych ráda poděkovala vedoucímu této bakalářské práce panu Doc. Ing. Adamu Heroutovi Ph.D. za jeho cenné rady při konzultacích a inspiraci při tvorbě této práce. Dále chci poděkovat své rodině za pevné nervy a podporu věnovanou během studia, přátelům, kteří mi byli oporou při psaní této práce, a Mgr. Aleně Hornákové za provedení jazykové korektury. Můj dík patří i těm, kteří se účastnili testování dílčích verzí aplikace.

© Marie Kratochvílová, 2014.

*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

<b>1 Úvod</b>	<b>2</b>
<b>2 Výběr vhodné technologie</b>	<b>3</b>
2.1 Počátky vývoje Windows Phone	3
2.2 Grafické uživatelské rozhraní Windows Phone	5
2.3 Katalog aplikací (Windows Store)	6
2.4 Windows Presentation Foundation (WPF)	8
2.5 User experience (UX)	8
2.6 Uživatelské rozhraní	8
<b>3 Analýza existujících aplikací</b>	<b>10</b>
3.1 Cal Track	10
3.2 Nutrition	11
3.3 Diet Planner	11
3.4 Eat Burn Learn	12
3.5 Calorie Counter Pro	12
3.6 MyFitnessPal	12
<b>4 Analýza a specifikace požadavků</b>	<b>16</b>
4.1 Dotazník	17
4.2 Prototypy	19
<b>5 Implementace a testování aplikace</b>	<b>24</b>
5.1 Návrhové vzory	24
5.2 Návrh úložiště dat	26
5.3 Použité technologie	28
5.4 Etapy vývoje aplikace	29
5.5 Zhodnocení a budoucnost aplikace	38
<b>6 Závěr</b>	<b>39</b>
<b>A Obsah DVD</b>	<b>41</b>

# Kapitola 1

## Úvod

V této závěrečné práci si kladu za cíl vytvořit jednoduché a intuitivní uživatelské rozhraní pro výběr a zadávání jídel, používané ke sledování životosprávy. Správná životospráva je důležitá pro lidský organismus. Dodává potřebnou energii tělesným buňkám, kterou potřebují pro svoji činnost a regeneraci. Správnou výživou se regeneruje i lidský organismus a zároveň se zvyšuje jeho odolnost proti škodlivým faktorům.

O správnou výživu jsem se začala zajímat v červnu minulého roku, kdy jsem měla zdravotní problémy s váhou. A na tomto základě jsem vymyslela aplikaci, která mi měla pomoci se zdravě stravovat.

Uživatelské rozhraní jsem tvořila s důrazem na jednoduchost a rychlost ovládání. Při jeho tvorbě jsem se snažila získat zpětnou vazbu uživatelů a na tomto základě jsem navrhla několik prototypů rozhraní, viz kapitola 4. Dále jsem si kladla za cíl vyvarovat se chyb u návrhů aplikací s podobnou tematikou. Tato část je popsána v kapitole 3.

Pro implementaci této práce jsem zvolila platformu Windows Phone, její filozofie a hlavní důvody výběru jsou blíže specifikovány v kapitole 2. Samotný vývoj ovlivňovaly rovněž požadavky uživatelů. Jednotlivé etapy vývoje spojené s průběžným testováním na uživateliích a použité technologie při implementaci jsou obsaženy v kapitole 5.

Budoucnost aplikace a její hlavní přínosy v této práci jsou shrnuty v kapitolách 5.5 a 6.

## Kapitola 2

# Výběr vhodné technologie

V této kapitole bych se ráda věnovala vybrané technologii, kterou jsem použila k realizaci této závěrečné práce. Pro realizaci jsem si vybrala platformu Windows Phone, dále jen WP, která v současné době zaujímá třetí místo mezi existujícími platformami, další informace poskytuje kapitola 2.1. Jeden z hlavních důvodů, proč jsem zvolila právě tuto platformu, je, že poskytuje široký a nevyužitý prostor pro vývoj různých aplikací a zároveň malý výskyt konkurenčních aplikací.

Mezi mé další důvody se řadí i:

- Běžný uživatel vlastní chytrý telefon a má ho stále u sebe.
- Platforma WP dává k dispozici nevyužitý prostor, co se týká vývoje různých druhů aplikací.
- Snadná implementace pomocí jazyka C# s použitím technologie .NET.
- Technologie .NET nabízí příjemné vývojové prostředí a spoustu rozsáhlých knihoven.
- Platforma WP je poměrně rychlá a přehledná oproti běžným zařízením s platformou Android.
- Společnost Microsoft, díky své stabilitě na trhu, chce i nadále podporovat tuto platformu.
- Prostedí Microsoft Visual Studio 2010, ve kterém je tato závěrečná práce vyvíjena.

### 2.1 Počátky vývoje Windows Phone

Platforma Windows Phone si prošla dlouhou cestou vývoje. Její původní název byl Windows Mobile, ale od verze Windows Mobile 6.5 se její pojmenování mění na platformu Windows Phone. Samotná historie pozdějšího operačního systému s názvem Windows Phone začala v roce 2000. V tomto roce společnost Microsoft uvedla na trh svůj první operační systém pod názvem Pocket PC 2000 pro mobilní zařízení. Tento systém běžel na jádře Windows CE. Byl dodáván spolu se základní sadou podporovaných aplikací, které byly vyvinuty prostřednictvím Microsoft Windows API. Pomocí Microsoft Windows API se jeho vzhled a funkce podobaly desktopové verzi Windows.

První mobilní zařízení měla k dispozici tzv. **stylus**, což bylo dotykové pero, se kterým mohl uživatel snadno zadávat příkazy za pomoci dotyku pera s obrazovkou mobilního zařízení. V pozdějších mobilních zařízeních stylus nahradil dotyk prstu uživatele.

Se zvyšujícími se nároky koncového uživatele se společnost Microsoft rozhodla vydat další verzi tohoto operačního systému. Verze Pocket PC 2002 spatřila světlo světa v říjnu roku 2001. Stejně jako její předchozí verze byla postavena na jádru Windows CE. Zároveň s touto verzí společnost Microsoft vydává první verzi operačního systému Windows Mobile, která se začíná používat pro chytré telefony, jako byl Smartphone.

Po první verzi Windows Mobile následovaly její další verze, které mimo jiné přinesly i další podporu nových technologií, jako je technologie Wi-fi a Bluetooth. Poslední verze Windows Mobile 6.5 je vydána již pod změněným názvem Windows Phone.

## Windows Phone

Windows Phone je snadno ovladatelný a intuitivní operační systém pro mobilní zařízení, který se od většiny své konkurence vyjímá svým pojetím. Jeho grafický vzhled nevychází z myšlenek vzhledu desktopového operačního systému Windows. Windows Phone má přinést novou éru v oblasti mobilních operačních systémů společnosti Microsoft.

Windows Phone disponuje minimálními nároky na HW. Přesto, že je tento systém novým nástupcem Windows Mobile, tak s tímto systémem není zpětně kompatibilní. Celá platforma je rovněž omezena na instalaci mobilních aplikací, které jsou běžně dostupné z Windows Store. Všechny tyto aplikace jsou odsouhlaseny a certifikovány společností Microsoft.

Celý systém se rovněž od své konkurence vyjímal naprosto novým a unikátním uživatelským rozhraním, které se nepodobalo ničemu, co je dosud na trhu. Toto unikátní uživatelské rozhraní poskytuje jednoduchost, přineslo nové použití typografie a zaměření na informace a intuitivní způsob, jak se k těmto informacím dostat.

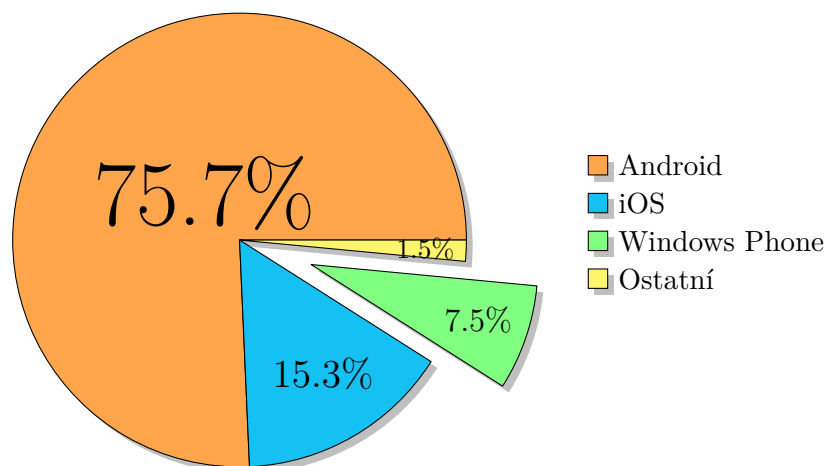
## Windows Phone 7

Operační systém Windows Phone 7 byl vydán 6. listopadu 2010. Cílem návrhářů bylo vylepšit dosavadní grafické rozhraní a co nejvíce zjednodušit dotykové ovládání tohoto operačního systému. Od této doby vydala společnost Microsoft několik aktualizací [4]. První aktualizace byla pojmenována jako aktualizace "**NoDo**". Tato aktualizace přinesla podporu kopírování a vkládání textů, větší rychlost a další vylepšení Marketplace. Další velkou aktualizací byla aktualizace označovaná jako "**Mango**". Windows Phone 7 se začal označovat jako Windows Phone 7.5. Tato aktualizace přinesla například integraci internetového prohlížeče Internet Explorer 9, integraci sociálních služeb jako je Twitter, mimo jiné i podporu češtiny.

V říjnu roku 2012 vydala společnost Microsoft nový operační systém Windows Phone 8. Zcela zásadní novinou u tohoto systému je jeho nové jádro, respektive podpora více jádrových procesorů. Tento systém převzal jádro ze systémů Windows 8. Hlavním přínosem této podpory by měla být větší výdrž baterie a větší výkon mobilního zařízení.

Vzhledem k tomu, že je to zatím poslední verze Windows Phone, která způsobila nárůst velkého zájmu koncových uživatelů, snažila jsem se i svou závěrečnou práci udělat kompatibilní s touto verzí systému.





Obrázek 2.1: Statistika nejvíce dominujících operačních systémů v Evropě (1. dubna 2014) dle [7].

Jak můžeme vidět z obrázku 2.1, tak v dnešní době je Windows Phone na vzestupu. V současné době zaujímá Windows Phone třetí příčku a je hned po platformě Android a iOS stále nejpoužívatelnějším mobilním operačním systémem. Důsledkem toho je i více příležitostí vývoje nových mobilních aplikací pro tuto platformu a dalším důvodem, proč jsem si právě tuto platformu vybrala.

## 2.2 Grafické uživatelské rozhraní Windows Phone

Jak bylo řečeno v kapitole 2, systém Windows Phone byl postaven na konceptu unikátního uživatelského rozhraní. Cílem bylo nechat se inspirovat navigačními tabulemi umístěnými na letištích nebo v metrech. Odtud vznikl název pro grafické uživatelské rozhraní Windows Phone jako Metro UI, které se později přejmenovalo na Modern UI. Modern UI je nové grafické uživatelské rozhraní s technologií více-dotykového ovládání, které vsází na jednoduchost a eleganci. Hlavní důraz je kladen na samotný obsah a na vhodné použití typografie, jenž může plnit funkci i jako skvělý designový prvek.

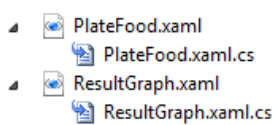
Úvodní obrazovka se skládá z dlaždic, které mají podobu čtverců nebo obdélníků. Dlaždice se dají přesouvat, zvětšovat, zmenšovat, přidávat nebo odebírat z úvodní obrazovky. Dále je možné měnit jejich barvu a pozadí (bílé nebo černé). Úvodní obrazovka má podobu základní matice, která je složena ze dvou sloupců a s neomezeným počtem řádků. Jednotlivé dlaždice slouží jako zástupce nějaké aplikace a zároveň mohou obsahovat různé notifikace (např. počet nepřečtených zpráv, zmeškaných hovorů aj.).

Modern UI je zaměřeno více na uživatele. Návrh probíhá tak, že se vezme skupinka náhodných uživatelů, zjistí se, k čemu by daný produkt chtěli používat, sepiší se jejich názory a poté se optimalizuje prostředí tak, aby uživatel dosáhl cíle co nejrychleji a nejjednodušeji. Při návrhu uživatelského rozhraní se odstraní přebytečné prvky a zachovají se pouze klíčové informace a funkcionalita.

Základním konstrukčním prvkem k tvorbě uživatelského rozhraní a vzhledu aplikací

je značkovací jazyk **XAML**<sup>1</sup>. Značkovací jazyk je postaven na standardech a principech obecných značkovacích jazyků XML<sup>2</sup>. Mají vždy stejný návrh struktury, tvorby a fungování. Zahrnuje to i využití jazyka XAML. Pro implementaci chování celé aplikace se používají jazyky z rodiny .NET<sup>3</sup>.

Výhodou jazyka XAML je oddělení programové části od uživatelského rozhraní. Ve WPF ([8]) aplikaci se každé okno skládá ze dvou souborů. První z nich má koncovku **"\*.xaml"** a druhý koncovku **"\*.xaml.cs"**. V prvním souboru jsou definovány v jazyce XAML jednotlivé prvky uživatelského rozhraní. Druhý soubor pak implementuje celou aplikační logiku, například využití databáze, zpracování dat, zobrazení výsledků aj. Výhoda oddělení těchto částí se uplatňuje například při práci v týmech na rozsáhlejších projektech, kde vývojáři mohou mít různá uplatnění. Oddělení aplikační logiky a grafického rozhraní lze vidět na obrázku 2.2.



Obrázek 2.2: Oddělení aplikační logiky a grafického rozhraní do dvou souborů

Jazyk XAML byl prvotně navržen pro část Microsoft .NET frameworku, která byla určena ke specifikaci uživatelského rozhraní pro Windows aplikace, tzv. Windows Presentation Foundation, zkráceně WPF. Později byl integrován v zjednodušené a pozměněné podobě do platformy Silverlight.

## 2.3 Katalog aplikací (Windows Store)

V katalogu aplikací, který bývá též označován jako Windows Store<sup>4</sup>, se nachází všechny dostupné aplikace pro Windows Phone. Celý katalog je rozdělen podle států, u nichž lze najít konkrétní aplikace. Z tohoto místa je možné aplikace stáhnout nebo přímo zakoupit.

Bohužel nelze instalovat jinou oficiální aplikaci než takovou, která je dostupná ve Windows Store. Výjimku tvoří telefon, který je zaregistrovaný jako vývojářský. Až po této registraci je možné aplikaci stáhnout a nahrát ji prostřednictvím programu Application Deployment do mobilního telefonu.

Pokud se vývojář rozhodne svou aplikaci umístit na Windows Store, tak jeho aplikace musí projít ověřovacím procesem. Nástroj WACK<sup>5</sup> zkontroluje, zda aplikace odpovídá podmínkám pro publikování na Windows Store, neobsahuje nežádoucí obsah a to,

<sup>1</sup>XAML, nebo-li označení pro Extensible Application Markup Language

<sup>2</sup>XML je označení pro Extensible Markup Language

<sup>3</sup>Do této rodiny jazyků .NET patří jazyk C# a jazyk Visual Basic. Více informací je možné nalézt na stránkách Microsoftu s článkem o .NET frameworku na <http://msdn.microsoft.com/en-US/vstudio/aa496123>.

<sup>4</sup>Katalog aplikací byl dříve označován jako Windows Phone Marketplace.

<sup>5</sup>WACK (z angl. Windows App Certification Kit) je nástroj pro ověření nezávadnosti aplikace před jejím umístěním na Windows Store. Více informací na <http://blogs.msdn.com/b/vyvojari/archive/2012/08/15/test-windows-store-aplikace-pomoc-237-wack.aspx>.

že je nezávadná. Výhoda tohoto procesu je, že si uživatel nemůže do telefonu nainstalovat vadnou aplikaci. Po tomto procesu je potřeba se zaregistrovat a vyplnit formulář na <http://dev.windowsphone.com/en-us> a nahrát danou aplikaci ve formátu XAP<sup>6</sup>. Tento formát je podobný formátu "zip". Je to archiv, který obsahuje soubory, knihovny nebo obrázky, které jsou potřeba ke spuštění programu (aplikace).

The screenshot shows the 'App info' page in the Windows Phone Dev Center. The page title is 'App info' and the app name is 'FoodTracker'. Below the title, there is a description: 'The info on this page is used to refer to your app here in the Dev Center, and also controls how it appears in the Store.' The form contains several fields:

- App name:** A text input field containing 'FoodTracker'.
- App alias\*:** A text input field containing 'FoodTracker'. Below it, a note states: 'This name is used to refer to your app here on Dev Center. The name your customer sees is read directly from your package.'
- App category:** A section with a note: 'The Category and Subcategory determine where the app will be listed in the Store. [Learn more.](#)'
- Category\*:** A dropdown menu with 'lifestyle' selected.
- Subcategory:** A dropdown menu with 'food + dining' selected.

Obrázek 2.3: Formulář pro vyplnění informací o dané aplikaci.

Aplikaci je možné publikovat na Windows Store ihned po úspěšně schválené certifikaci, nebo lze publikovat aplikaci později, tzn. jednoduše se vybere pozdější datum publikace. Aplikace může být zveřejněna po nějakou omezenou dobu nebo může být volně ke stažení jako zkušební verze. Což se většinou provádí u aplikací, které jsou dostupné za nějaký poplatek. Tady má uživatel možnost si danou aplikaci stáhnout, vyzkoušet a sám se rozhodnout, zda-li mu daná aplikace vyhovuje, nebo naopak nevyhovuje.

Já jsem využila možnosti jednoduchého umístění na Windows Store, do budoucna bych chtěla přejít na zkušební verzi, která by později byla za příplatek.

<sup>6</sup>XAP nebo AAPF, je archiv, která obsahuje potřebné soubory ke spuštění aplikace. Více informací na <http://msdn.microsoft.com/library/windowsphone/help/jj206723%28v=vs.105%29.aspx>.

## 2.4 Windows Presentation Foundation (WPF)

Windows Presentation Foundation (dále jen WPF) je grafický framework<sup>7</sup> pro psaní Windows klientských aplikací. WPF je nástupcem starší verze technologie Windows Forms. Vznik WPF způsobila potřeba psát složitější grafické aplikace a sjednotit způsob návrhu napříč platformami. WPF se snaží poskytnout co nejrozšířitelnější objektový model, vývojár není nijak vázán na pevnou sadu komponent s pevným vzhledem, ale může si libovolně upravit jakoukoli její část či si vytvořit zcela novou. Mezi výhody WPF [5] patří:

- Komplexnější model - nabízí řadu událostí, vlastností a parametrů.
- Vázání dat - pomocí *binding* lze vázat vlastnosti datového objektu na uživatelské prostředí, a tak jím lze zajistit jednodušší logiku uživatelského prostředí bez nutnosti psaní kódu.
- Grafika - celé WPF je navrženo prostřednictvím stylů, kterými lze měnit vzhled a funkčnost prvků.
- Vektorové transformace - lze zmenšovat, zvětšovat, otáčet a jinak transformovat elementy, neboť celé WPF je vektorové.

## 2.5 User experience (UX)

Při návrhu uživatelského rozhraní se obvykle vychází z uživatelských prožitků. User experience, zkráceně UX, vyjadřuje celkovou míru spokojenosti uživatele s produktem. Je zaměřený na uživatele, tj. na jeho potřeby a cíle i na jejich omezení. UX nemá přesnou definici a špatně se překládá i do češtiny. Ale podle předních odborníků, například podle Jeffa Johnsona [2], který se návrhu uživatelského rozhraní věnuje od roku 1978, je user experience zhruba toto: "Uživatelský prožitek je přesně to, co název napovídá: všechno, co uživatel vidí a s čím se potká, když stránku navštíví a chce ji vyzkoušet. Nenáleží sem pouze struktura stránky a její obsah, ale také to, jak uživatel stránku najde, zda funguje v jeho prohlížeči nebo mobilním zařízení, zda stránka poskytuje pomoc těm, kdo se setkají s problémem atd. Vše musí fungovat dobře, jinak nebude stránka z uživatelského hlediska úspěšná. Pokud nefunguje, navštíví uživatel stránku jinou."

Tzn. to, jak celou aplikaci uživatel vnímá, ovlivňuje jeho zájem o ni. Více informací o prvním návrhu této práce najdete v kapitole 3.

## 2.6 Uživatelské rozhraní

Uživatelské rozhraní je velice široký pojem, který by našel řadu definic. Dnes se s pojmem uživatelské rozhraní setkáváme běžně. Ať už by se jednalo o návrh uživatelského rozhraní pro mobilní platformu, nebo o návrh uživatelského rozhraní pro webové stránky. Podle mého názoru je jenom jedno uživatelské rozhraní, jehož cílovým aspektem není jen jeho kontext, ale také celkový úhel pohledu.

---

<sup>7</sup>Framework je obecné označení například pro strukturu knihoven, jejichž cílem je ulehčit práci při programování aplikace. Tzn. méně psaní, přehlednější kód a rychlejší vývoj.

Uživatelské rozhraní by se dalo obecně definovat jako souhrn prostředků pro komunikaci mezi uživatelem a zařízením.

Uživatelské rozhraní definuje soubor postupů, jak pracovat s daným rozhraním, objektem nebo danou věcí. Za zařízení, objekty nebo věci můžeme považovat například automobil, pračku nebo mobilní telefon aj.

Ve smyslu věci se za uživatelské rozhraní dají pokládat například tlačítka, ikony a menu mobilního telefonu. U automobilu bychom za uživatelské rozhraní mohli definovat hlasovou navigaci GPS. Zde si dovolím citovat paní Součkovou [11], podle ní je uživatelské rozhraní tvořeno souborem vstupních a výstupních zařízení, procesů a programového vybavení, kontrolující tyto procesy. Nebo naopak podle pana Papíka [6] se uživatelské rozhraní skládá z fyzické a konceptuální části. Fyzická část zde představuje vstupní a výstupní zařízení a nástroje pro zpětnou vazbu. V konceptuální části se nachází systémy pracující s ikonami, s prvky inteligentního rozhraní nebo menu apod. Při návrhu uživatelského rozhraní této práce, kterému se budu více věnovat v kapitole 4, jsem se nechala inspirovat Stevem Krugem, který popisuje v knížce "Don't make me think" [3], jakým způsobem běžný uživatel přemýšlí. Přitom se nemusí jednat jenom o webovou stránku, ale může jít i o stránku aplikace.

## Počátky vývoje uživatelského rozhraní

Na začátku 50. a 60. let 20. století probíhala komunikace mezi uživatelem a zařízením prostřednictvím světelných signálů, různých přepínačů a později i pomocí děrných štítků. Právě na těchto prvcích se zrodilo první uživatelské rozhraní. Avšak nevýhodou těchto rozhraní byla jejich vysoká náročnost na odborné znalosti samotného uživatele. Z tohoto důvodu byla tato rozhraní ovládána vědci, z nichž převážnou část tvořili různí matematici a pak to byli i odborníci. Další nevýhodou byla i skutečnost, že nepodporovali tzv. multitasking (tj. proces zpracování více úloh současně).

V pozdějších letech, tj. v 60. letech 20. století, vznikla další nová rozhraní, byla to příkazová řádka (z ang. command line interface), kdy mezi uživatelem a zařízením probíhá komunikace v textové podobě prostřednictvím nějakého skriptovacího jazyka. Komunikace probíhala podle předem určených příkazů v daném jazyce. Byl zde kladen důraz na znalosti samotného uživatele ovládat dané příkazy v konkrétním jazyce. Vstupní zařízení tvořila klávesnice.

V 60. letech se objevila technika direct manipulation - tzv. přímá manipulace. Tato technika byla poprvé použita v roce 1982 Benem Shneidermanem [10]. Vstupním prvkem bylo ukazovací zařízení, které umožňovalo manipulaci s objekty umístěnými na obrazovce. Vstupní prvek představovalo světelné pero, s jehož pomocí se dalo manipulovat s objekty, takže se mohla měnit jejich velikost či se daly různě posouvat.

Do rozhraní přímé manipulace řadíme i techniku WYSIWYG - what you see is what you get. Jedná se o způsob manipulace s texty a objekty, otvírání a jejich výběr.

V dalších letech 21. století se objevují nová uživatelská rozhraní, kdy mezi jejich trendy patří vylepšování stávajících funkcí, grafického rozhraní a snaha přizpůsobit se lidským potřebám, propojení uživatelského rozhraní s dalšími aplikacemi, jako jsou například online obchody.

## Kapitola 3

# Analýza existujících aplikací

V této kapitole bych se ráda věnovala analýze existujících aplikací s podobnou tématikou zadávání životosprávy, jejich představení a srovnání. Analýzu aplikací jsem prováděla před návrhem této práce a dále v průběhu akademického roku. Velmi často se myšlenka aplikací opírá o návrhový vzor a přichází s různým vylepšením ovládacích prvků, vzhledu včetně přidání různých pasáží s textem o dané problematice. Častým výsledkem je jednoduchá a vylepšená aplikace, která se dostane k uživateli. Míra návrhu aplikace je individuální pro každou aplikaci, kterou navrhuje vývojář. Vybrané aplikace jsem se snažila používat po nějaký čas, abych mohla odhalit jejich nedostatky a vyvarovat se jim ve své práci. Mezi těmito nedostatky jsem zaznamenala:

- Náročné ovládání prvků aplikace a proklikávání se mezi různými částmi aplikace.
- Časté mrhání volným místem na obrazovce.
- Nelogické rozmístění ovládacích prvků a náročná funkcionality.
- Špatný návrh grafického rozhraní.
- Pochopení účelu používání aplikace.

Veškeré analyzované aplikace jsou jinak orientované na danou tematiku oproti mému řešení. Jejich převážná část obsahovala seznam jídel s uvedenou kalorickou tabulkou včetně prezentování informací o nutričních hodnotách, avšak žádná z testovaných neposkytovala intuitivní a rychlé zadávání jídla. Dále zmíněné aplikace lze najít v katalogu aplikací (Windows Store), které jsou kompatibilní s platformou Windows Phone.

### 3.1 Cal Track

Aplikaci Cal Track<sup>1</sup> lze vidět na obrázku 3.1 je volně dostupná ke stažení ve Windows Store. Aplikace umožňuje uživateli sledovat množství přijatých kalorií za celý den. Jako první krok musel uživatel zadat potřebné údaje k vypočítání bazální hodnoty metabolismu. Bazální, nebo-li klidový metabolismus vyjadřuje přibližné množství energetického příjmu, který je

---

<sup>1</sup>Cal Track se nalézá ve Windows Store na adrese <http://www.windowsphone.com/cs-cz/store/app/cal-track/b2a33cee-0ffe-4270-b810-7dac95f1ec2c>.

potřeba k zachování důležitých základních životních funkcí (dýchání, trávení apod.). Ve druhém kroku si uživatel zvolí, jakou má pohybovou aktivitu, a aplikace mu spočítá celkový počet kalorií, které by za celý den měl přijmout. Při dalších krocích si uživatel zadává zkonsumované potraviny a sleduje svůj denní kalorický příjem. Tato aplikace neobsahuje seznam jídel s kalorickou tabulkou včetně dalších informací, tuto věc považují za hlavní nevýhodu této aplikace, neboť uživatel při zadávání jídla nemusí znát všechny informace. Další věc, která mě odradila, byla úvodní obrazovka, která zobrazuje zbytečné "rozcestí". V aplikaci by stačilo použít panoramatických prvků, které by umožnily uživateli krok po kroku projít celou aplikací. Naproti tomu se v této aplikaci musel uživatel proklikávat různými částmi, které dle mého názoru byly chaoticky rozmístěné. Výhoda této aplikace spočívá v přehledném seznamu jídel s přijatými kaloriemi a v hlídání dosažení celkové hladiny přijaté energie za celý den. Další výhodou je barevné rozlišení textů.

## 3.2 Nutrition

Aplikace Nutrition<sup>2</sup> je vidět na obrázku 3.2. Aplikace je dostupná za cenu 20 Kč, ale je možné si ji vyzkoušet ve zkušební verzi. Aplikace mne zaujala s možností nastavit si takové informace, které mě u jídla zajímají, například zobrazení sacharidů, tuků aj. Aplikace poskytuje uživateli vyhledat si zvolenou potravinu v seznamu jídel, který je značně rozsáhlý oproti jiným aplikacím. Každá potravina je rozdělena do kategorií a ve svém detailu zobrazuje velkou část nutričních hodnot. Co se mi na aplikaci moc nelíbí, je výpis nutričních hodnot dané potraviny, dle mého názoru by tvůrce mohl zauvažovat nad grafickým zobrazením. Celá aplikace je dle mého názoru užitečná pro ty uživatele, kteří chtějí vědět, kterým jídlům se raději vyhnout. Aplikace neobsahuje grafické prvky, pouze rozsáhlé informace. Není zde možnost zadat si vlastní jídlo, pouze je uživateli poskytnuto vyhledávací pole, jehož filtrováním si uživatel najde podobná slova, která obsahují jeho frázi. Zvolí si potravinu a může si ji uložit k nejčastěji vyhledávaným jídlům a vést si tak jejich přehled.

## 3.3 Diet Planner

Aplikace Diet Planner<sup>3</sup> je volně dostupná na Windows Store a lze ji zhlédnout na obrázku 3.3. Aplikace poskytuje docela příjemně sladěné grafické prostředí. V prostředí se nalézají různé kategorie jídel s ikonkou. Každá z těchto kategorií má roztříděný rozsáhlý seznam jídel, který poskytuje přehled nutričních hodnot každého jídla. Uživatel si prostřednictvím aplikace snadno spočítá množství přijaté energie. Aplikace nenabízí volbu množství u zvoleného jídla. Množství má zde výchozí hodnotu. Nevýhodou aplikace vidím v neposkytnutí vyhledávání potraviny, uživatel má pouze k dispozici seznam jídel, ve kterém pouze "scroluje". Tato aplikace celkově slouží pouze v poskytování nutričních hodnot, nikoli pro zadávání jídel.

---

<sup>2</sup>Nutrition se nalézá ve Windows Store na adrese <http://www.windowsphone.com/cs-cz/store/app/nutrition/97ffeafe-e6d5-df11-a844-00237de2db9e>.

<sup>3</sup>Diet Planner se nalézá ve Windows Store na adrese <http://www.windowsphone.com/en-us/store/app/diet-planner/acf911f7-b744-4e01-aa92-593cd0911ef7>.



### 3.4 Eat Burn Learn

Aplikace Eat Burn Learn<sup>4</sup> je vidět na obrázku 3.4. Aplikace je dostupná za cenu 40 Kč, ale je možné si ji vyzkoušet ve zkušební verzi. Zkušební verze poskytuje pouze základní ovládání. Aplikace umožňuje vést si přehled o přijatých kaloriích za celý den a používat ji ke spalování nadbytečných kilogramů. Uživatelské rozhraní disponuje jiným vzhledem pro každou sekci aplikace. Jsou tu k dispozici grafy, které sledují například pokrok úbytku na váze. Tyto grafy jsou vidět pouze v placené verzi aplikace. Aplikace neposkytuje přehled všech jídel a jejich nutričních hodnot, slouží pouze pro zaznamenání jídla. Nevýhodu vidím v tom, že uživatel musí znát příjem kalorií každého jídla, které chce zaznamenat. Další problém je, že aplikace je na ovládání příliš komplikovaná, neustálé zadávání jídla je pomalé a zdlouhavé. Tato aplikace ze všech analyzovaných aplikací na mě nepůsobila příjemným dojmem. Spíše mě odradila.

### 3.5 Calorie Counter Pro

Aplikace Calorie Counter Pro<sup>5</sup> je vidět na obrázku 3.5. Aplikace je dostupná za cenu 20 Kč. Tato aplikace je jedna z mála, která mě zaujala. Překvapila mě podobnou myšlenkou zadávání jídla. Oproti mé aplikaci se liší v tom, že každé jídlo má u sebe uvedenou kalorickou hodnotu. Jídlo si uživatel může zvolit sám, z jakého si seznamu jídel nebo si potravinu může přidat. Opět pro přidání musí znát jeho kalorickou hodnotu. Aplikace neposkytuje funkční seznam všech jídel včetně jejich nutričních hodnot. Aplikace umožňuje uživateli vést si pravidelný příjem kalorií z potravin a zobrazit jej v docela přehledném grafu.

### 3.6 MyFitnessPal

Aplikace MyFitnessPal<sup>6</sup> je ke zhlédnutí na obrázku 3.6. Ze všech testovaných aplikací se mi nejvíce líbila právě tato aplikace. Působila na mě velice příjemným dojmem a poskytovala jednoduché ovládání. Umožňuje vést si přehled o přijatých kaloriích každý den. Při zaznamenání nového jídla si uživatel může vybrat kategorii, kam dané jídlo zařadit, například kategorie snídaně nebo oběd. Aplikace disponuje poměrně rozsáhlou databází jídel včetně přehledu nutričních hodnot. Je zde vyhledávací pole pro snadnou manipulaci a ruční zadání porce jídla. Překvapila mě rovněž téměř stejná podobnost s mou aplikací. Aby si uživatel mohl vést úplný přehled, musí se v první řadě zaregistrovat vytvořením nového účtu. Toto bych viděla za malý problém aplikace. Další z věcí, kterou uživatel dle mého názoru na první kliknutí nenajde je možnost zadání nového jídla a také to, že při přidání nového jídla do stejné kategorie, například ke snídani, je uživatel nucen projít si celý proces zadávání jídla znova.

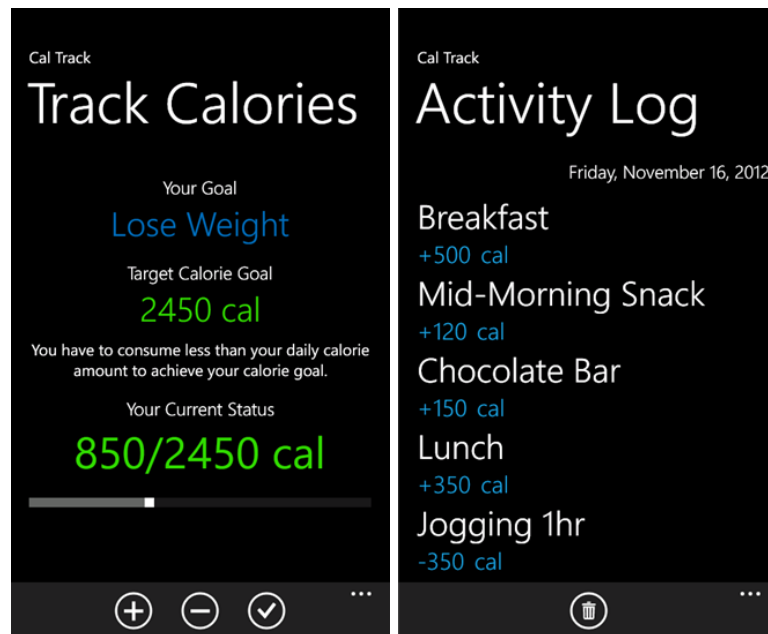
---

<sup>4</sup>Eat Burn Learn se nalézá ve Windows Store na adrese <http://www.windowsphone.com/cs-cz/store/app/eat-burn-learn/506dac4e-1f13-e011-9264-00237de2db9e>.

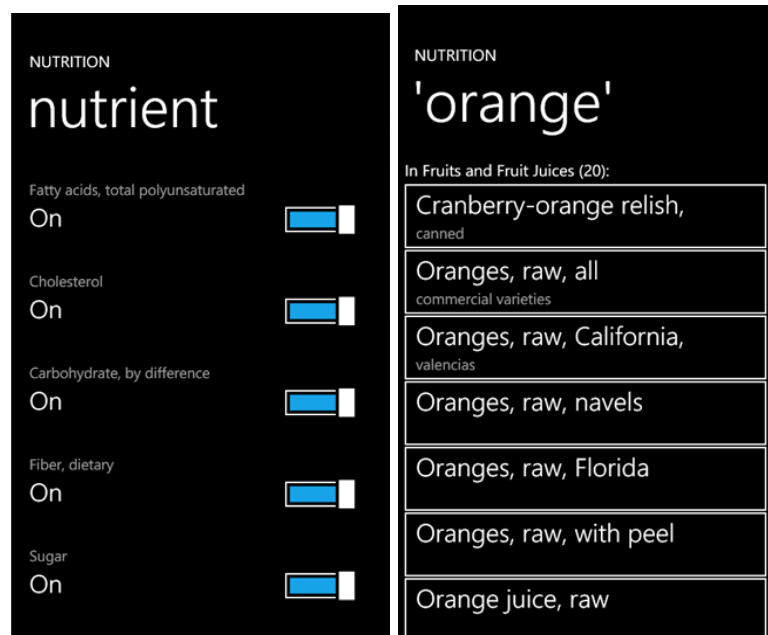
<sup>5</sup>Calorie Counter Pro se nalézá ve Windows Store na adrese <http://www.windowsphone.com/en-us/store/app/calorie-counter-pro/f959dc05-204e-40be-972f-16f5d2cb2187>.

<sup>6</sup>MyFitnessPal se nalézá ve Windows Store na adrese <http://www.windowsphone.com/en-us/store/app/myfitnesspal/22841d0a-389c-e011-986b-78e7d1fa76f8>.

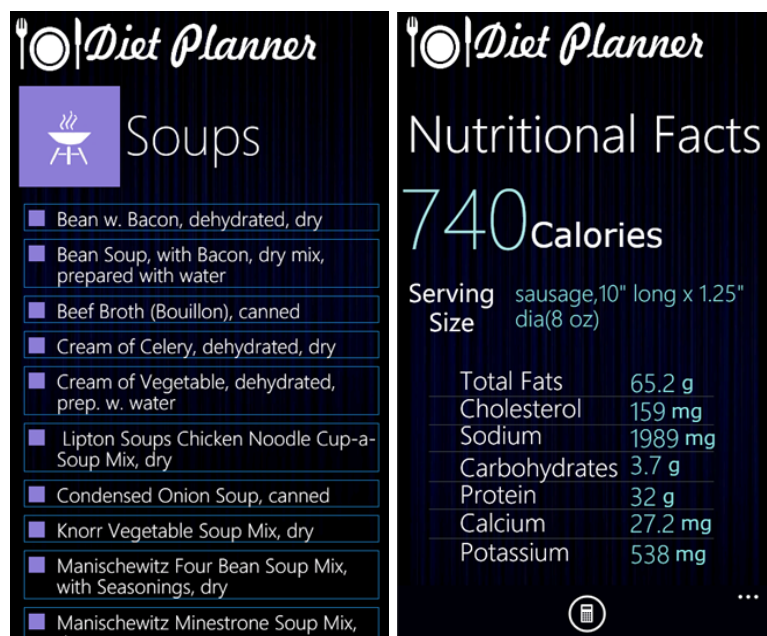




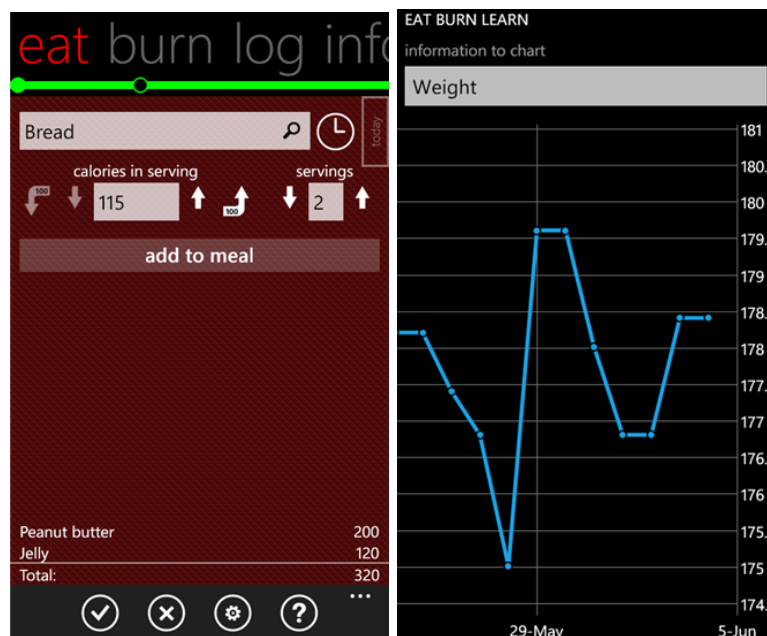
Obrázek 3.1: Aplikace Cal Track poskytuje přehledný seznam jídel s přijatou energií a má barevně rozlišený tech.



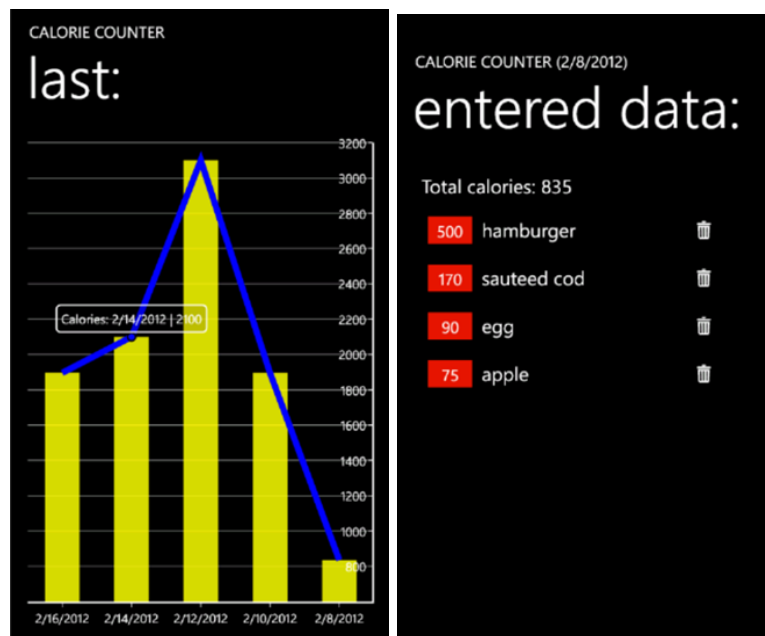
Obrázek 3.2: Aplikace Nutrition nabízí sice rozsáhlou databázi potravin, ale neposkytuje možnost sestavení si vlastního jídla.



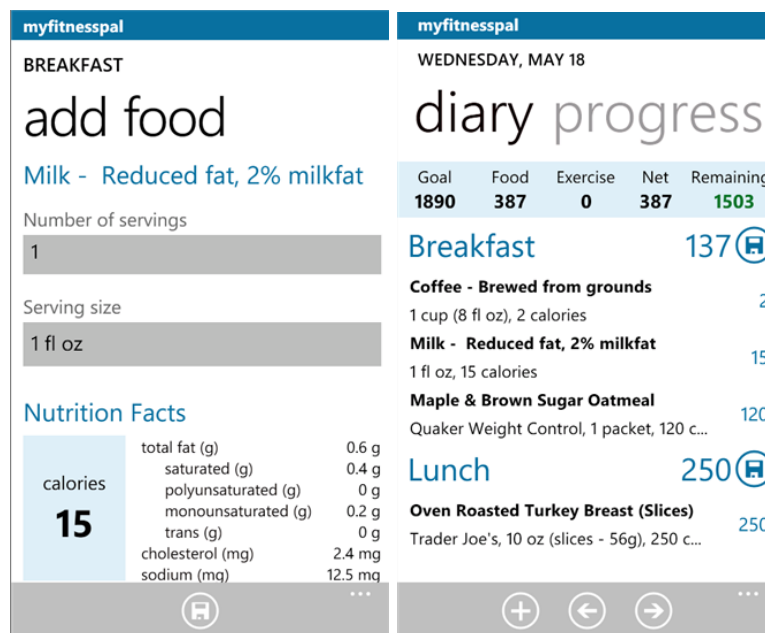
Obrázek 3.3: Aplikace Diet Planner má příjemné grafické rozhraní, ale uživatel v ní nemůže vyhledávat potravinu.



Obrázek 3.4: Aplikace Eat Burn Learn je příliš komplikovaná a uživatele spíše odrazuje a mate.



Obrázek 3.5: Aplikace Calorie Counter Pro neposkytuje žádný seznam jídel včetně nutričních hodnot. Poskytuje pouze přehled denní dávky energie.

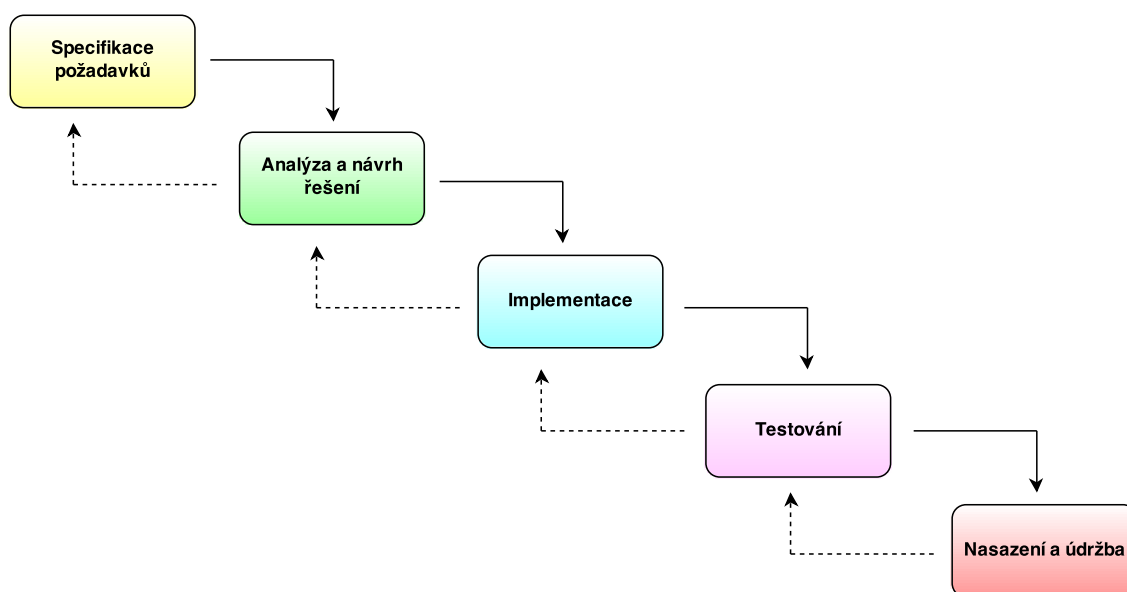


Obrázek 3.6: Aplikace MyFitnessPal mne zaujala téměř podobnou myšlenkou zadávání jídla.

## Kapitola 4

# Analýza a specifikace požadavků

Po analýze existujících aplikací a výběrů vhodné technologie je nutná analýza a specifikace požadavků pro implementaci této práce. Při návrhu této práce se snažím vyhnout nedostatkům, které jsem zaznamenala v analýze existujících aplikací v kapitole 3. Zároveň vycházím ze specifikace požadavků uživatelů a z jejich zpětné vazby. Abych si udělala ucelenou představu o požadavcích na výslednou aplikaci, rozhodla jsem se vytvořit dotazník a rozšířit ho mezi potenciální zákazníky. Na základě takto získané zpětné vazby jsem zrealizovala první etapu vývoje aplikace. K realizaci této práce jsem zvolila iterativní vývoj, ve kterém nejprve zjišťuji požadavky potenciálních uživatelů, analyzuji a navrhuji řešení, implementuji a zároveň testuji aplikaci. Model iterativního vývoje je vidět na obrázku 4.1 a více se o iterativním vývoji zmíním v kapitole 5.



Obrázek 4.1: Iterativní vývoj aplikace.

Jeho výhody lze shrnout do několika bodů:

- Neustálá komunikace s uživateli.

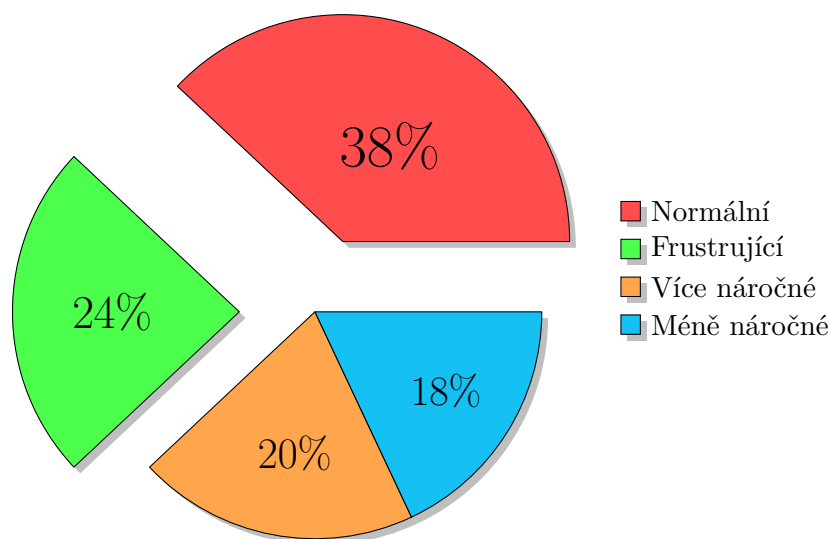
- Aktuální prototyp aplikace vyvíjený vždy po 2 týdnech.
- Včasné zachycení klíčových požadavků od uživatelů díky jejich zpětné vazbě.
- Eliminace zbytečných ovládacích prvků, vylepšování nadbytečných funkcí.

## 4.1 Dotazník

V dotazníku se ptám uživatelů na jejich zájem o vlastní životosprávu. Dále se zajímám o jejich zkušenosti s používáním aplikací s podobnou tematikou včetně toho, co by jim měla daná aplikace přinést. Celkově mi dotazník zodpovědělo 40 uživatelů. Výsledky dotazníku jsem rozdělila na dvě části. První část zobrazuje grafy s přehledem životosprávy a druhá shrnuté požadavky na výslednou aplikaci.

### Přehled životosprávy

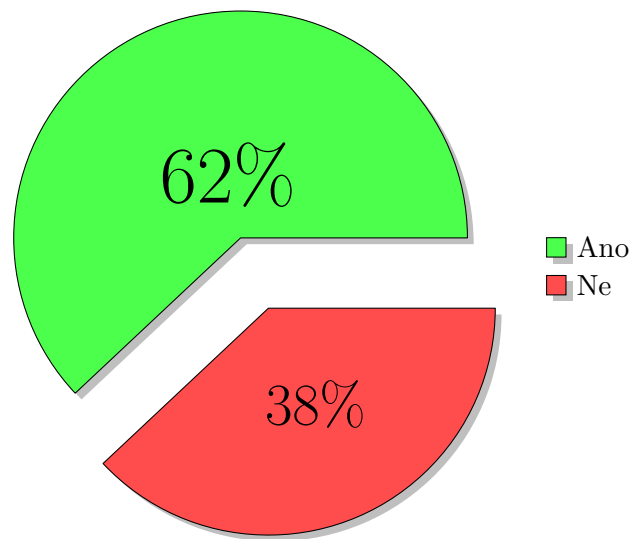
Ze čtyřiceti dotázaných uživatelů tvořili 52% muži a 48% ženy, kde průměrný věk byl 18-26 let. Nejvíce se průzkumu zúčastnilo 61% studentů, zbytek tvořili zaměstnanci (61%) nebo sportovci (9%). Většina z nich hodnotila svůj aktivní život jako normální nebo jako frustrující, což vyplývá z průzkumu na obrázku 4.2.



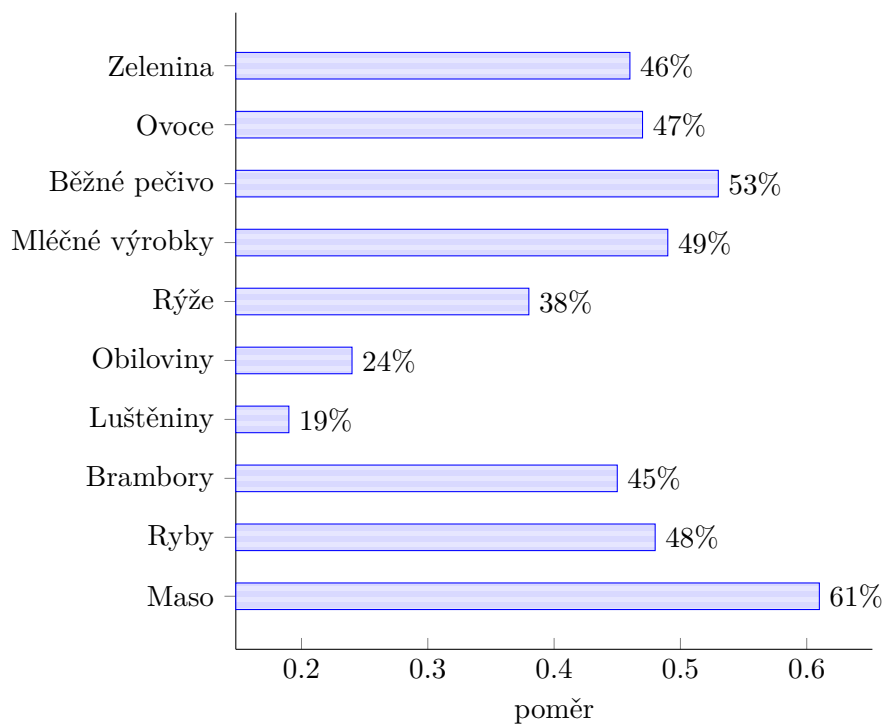
Obrázek 4.2: Uživatelé do aktivního života zahrnují práci nebo školu. To, jak na ně působí, lze zhlédnout na tomto obrázku.

Jak velkým zájem má uživatel o svou životosprávu, je vidět na obrázku 4.3. V nejběžnější čas, mezi 7. a 8. hodinou začíná každý den snídat 53% uživatelů. Zbytek uživatelů snídá nepravidelně nebo vůbec. 38% obědvá nejčastěji mezi 12. a 13. hodinou a 35% večeří mezi 18. a 19. hodinou. Zbylá procenta dotázaných uživatelů jíjí v jinou dobu.

Nejčastější konzumované jídlo podle statistiky na obrázku 4.4 je maso (61%).



Obrázek 4.3: Velká část dotázaných uživatelů jeví zájem o svou životosprávu.



Obrázek 4.4: Nejčastěji konzumované potraviny.

## Přehled požadavků

V této části dotazníku následovaly otázky s vlastní možností textových odpovědí. Uživatelé měli zodpovědět, jak by podle nich měla vypadat výsledná aplikace, a které vlastnosti se jim nelíbí na jejich aplikaci, kterou používají k zaznamenání přehledu životosprávy. Nejčastějšími požadavky na otázku, jak by měla vypadat výsledná aplikace, jsem shrnula do těchto bodů:

- Aplikace by měla být snadno ovladatelná a rychlá.
- Aplikace by měla zobrazovat výsledky ve formě grafů.
- Aplikace by měla poskytovat přehledné a jednoduché uživatelské rozhraní.
- Aplikace by měla umět zadávat požadavek ve třech krocích a zároveň ho jednoduše upravit.
- Aplikace by měla být méně náročná na výkon uživatelova mobilního telefonu.
- Notifikace na události uvnitř aplikace.

Stejný souhrn odpovědí jsem sestavila i na otázku, jaké nevýhody uživatel pozoruje na jiné používané aplikaci k zaznamenávání přehledu životosprávy. Nejčastější odpovědi byly:

- Aplikace poskytuje zbytečné složité ovládání.
- Aplikace nereaguje rychle na uživatelovy požadavky.
- Aplikace je náročná na výkon uživatelova mobilního telefonu.
- Aplikace obsahuje spoustu reklam.
- Aplikace disponuje spousty funkcemi navíc.
- Aplikace má spoustu stránek, textů, ovládacích prvků a málo obrázků.
- Aplikace není intuitivní a přehledná.

Body jsou shrnuty podle nejčastěji se objevovaných odpovědí dotazovaných uživatelů. Vzhledem k požadavkům na výslednou aplikaci jsem přehodnotila svá předchozí stanoviska a snažila se co nejvíce vyhovět požadavkům uživatelů. Některé z výše uvedených požadavků jsem měla i já.

## 4.2 Prototypy

**Prototyp**, nebo také **Mockup**, je návrh vyvíjené aplikace nebo představa interakčního návrháře [9]. Při návrhu dojde k vytvoření modelu z konceptů. Napodobuje ovládací funkce a chování výsledné aplikace. Vytváření návrhů prototypů tvoří iterativní proces. Po vytvoření prototypu proběhne testování na uživateli a na základě jejich zpětné vazby se provedou potřebné změny a celý proces se opakuje tak dlouho, dokud se nedosáhne požadovaných výsledků.

Na základě získané zpětné vazby z dotazníku jsem navrhla několik rozhraní. První prototypy návrhů jsem kreslila za pomoci tužky, pastelek a bílého papíru. Později jsem přešla na tvoření mockupů pomocí nástroje Mockup Bilder<sup>1</sup>. Nástroj umožňuje navrhovat vzhled aplikace nebo webových stránek včetně rozmístění ovládacích prvků.

Pomocí Mockup návrhů definuji:

- Představu o prioritách jednotlivých prvků, ovládacích tlačítkách aj.
- Počet prvků grafického rozhraní, počet textových polí a obrázků.
- Jaké druhy informací budou zobrazeny.
- Možnosti chování uživatelského rozhraní, například reakce formulářů na chyby, zobrazení chybových hlášek aj.

Po dohodě s vedoucím mé bakalářské práce jsem se rozhodla oddělit implementační část od návrhu rozhraní. Návrhů bylo více, za pomoci testování uživatelů a konzultace s vedoucím této práce jsem vybrala potencionálně nejlepší a pokoušela se ho zdokonalovat. Pomocí těchto návrhů jsem si ucelila své představy o aplikaci.

## Prototyp A

Můj první návrh jsem kreslila na papír, ale problém nastal při odhadování velikostí obrazovky mobilního telefonu, ale také jednotlivých ovládacích prvků. To jsem však vyřešila změnou obyčejného bílého papíru za papír čtverečkový. Díky tomu jsem věděla rozměry obrazovky mobilního telefonu. Další problém bylo vymýšlení nových věcí, které mě oddalovaly od původního návrhu. Navíc jsem nechtěla kreslit stejné prvky dokola ( např. obrys telefonu, který je stále stejný). To byl důvod, proč vznikly jednotlivé šablony telefonu pomocí nástroje Pencil<sup>2</sup>.

Na obrázku 4.5 je vidět první návrh úvodní obrazovky. Návrh je postaven na rozdělení jídla do kategorií. Tzn. neměly žádný popis, ale obrázek, který je charakterizoval. Toto řešení se objevilo v pozdějších návrzích uživatelského rozhraní aplikace.

V tomto návrhu lze vidět tlačítko, které slouží k vyhodnocení složení sestaveného jídla a zobrazení v podobě grafu.

Nebyla zde zakomponovaná myšlenka zobrazování chybových hlášek. V situaci, kdy si uživatel nevybere žádné jídlo a jako první si zvolí možnost vyhodnocení.

Výběr z uživatelem vybrané kategorie je zobrazen na obrázku 4.5. Jak je vidět z obrázku, hlavním prvkem uživatelského rozhraní je seznam jídel, které je možné vybrat jednoduchým zaškrtnutím políčka.

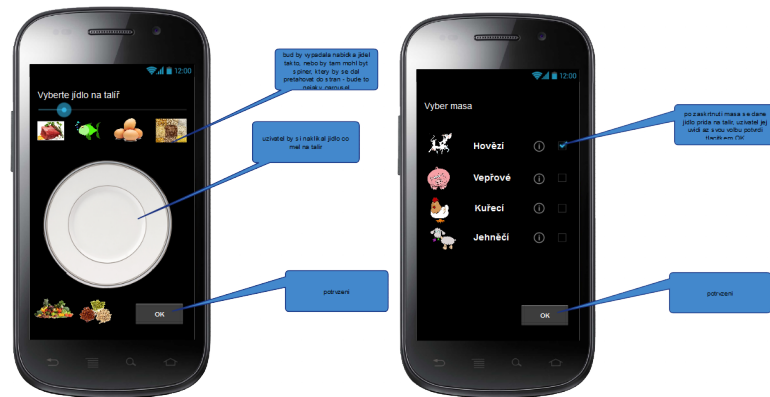
Po tomto výběru se uživatel dostane na předchozí obrazovku, na které přibylo vybrané jídlo. Tento výběr je vyobrazen na obrázku 4.6. Rovněž je vidět vyhodnocení vybraného prvku ve formě grafu.

---

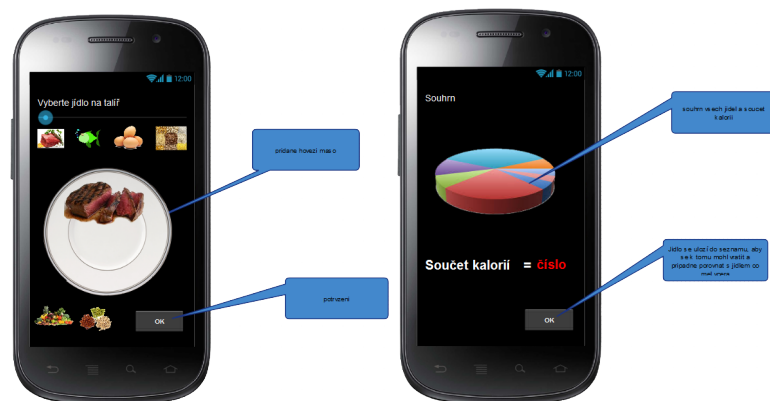
<sup>1</sup>Mockup Bilder je placená verze. Lze si ji vyzkoušet na stránkách <http://mockupbuilder.com/>

<sup>2</sup>Pencil je nástroj, pomocí kterého lze snadno navrhnout uživatelské rozhraní budoucí aplikace. Je k dispozici na stránkách <http://pencil.evolus.vn/>.





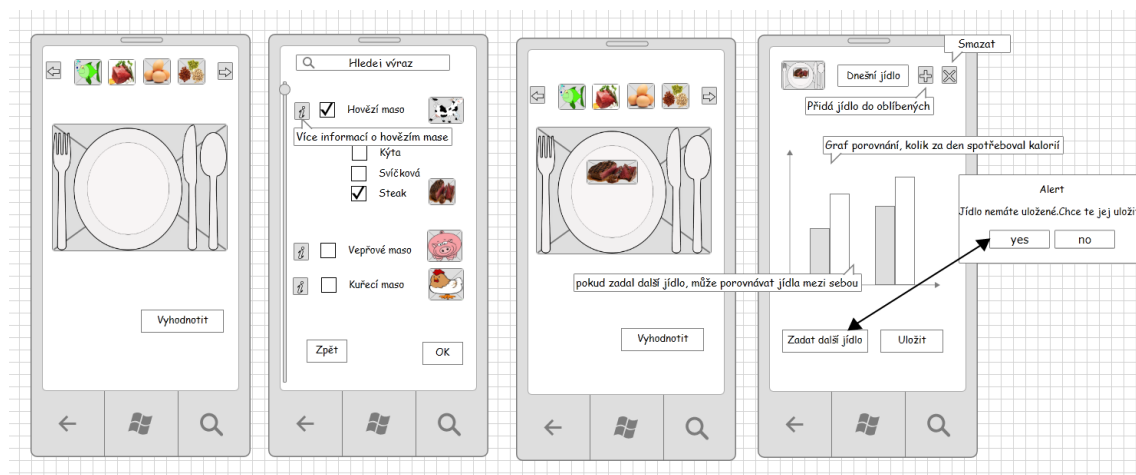
Obrázek 4.5: Úvodní obrazovka a kategorie zobrazující seznam prvků.



Obrázek 4.6: Zobrazení vybraného jídla na talíři a vyhodnocení v podobě grafu

## Prototyp B

V druhém návrhu jsem změnila prostředí Pencil za prostředí Mockup Builder, neboť mi neposkytovalo všechny potřebné prvky, které jsou běžné pro platformu Windows Phone. Nástroj Mockup Builder byl rovněž použit pro další návrhy aplikace. Jak je vidět na obrázku 4.7, od původního návrhu zde přibyla další tlačítka a provedly se i menší změny v návrhu.



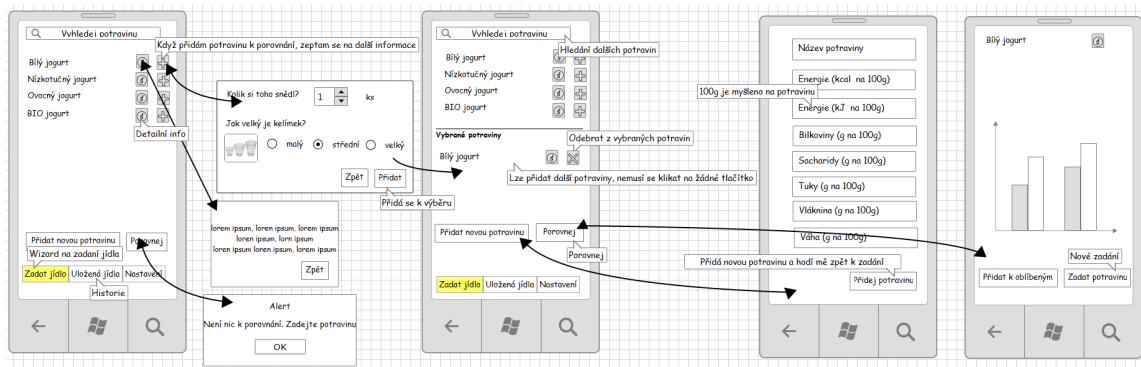
Obrázek 4.7: Druhý návrh aplikace

Změnil se výběr prvků v kategoriích. Na obrázku 4.7 je vidět, že každá kategorie má i svou podkategorii. Dále je zde možnost i vyhledávání konkrétního výrazu. Přibyla zde i zpětné tlačítko, kterým se uživatel vrátí o jeden krok dozadu. Kategorie lze pohybovat do stran. Dále se změnil i vzhled talíře a grafu. Graf zobrazuje na ose x daný produkt a na ose y jeho přijatou energii. V návrhu lze zadat nové jídlo. Stejně tak jsou zde vyobrazeny chybové hlášky. V tomto návrhu nelze smazat dané jídlo z talíře, uživatel si může zvolit uložení sestaveného jídla.

## Prototyp C

S dalšími požadavky od uživatelů a zpětnou vazbou vedoucího mé závěrečné práce byl návrh modelu aplikace složitější a rozsáhlejší. Tento návrh umožňoval zadávat pouze jeden produkt a o něm dostat konkrétní informace. Uplatnil by se v situaci, kdy uživatel jde do obchodu a v chladicím boxu vidí několik produktů. Z těchto produktů jej zajímá jeden. Ten produkt by si v aplikaci našel a podle informací, které by dostal, by se rozhodl, zda je pro něj přijatelný, či nikoli. Podobný návrh rozhraní existuje, proto jsem se raději zaměřila na průvodce zadáváním jídla na talíř zmíněného v předchozích odstavcích a vylepšení navrženého modelu aplikace. Náhled návrhu rozhraní se zadáním jednoho produktu je ke zhlédnutí na obrázku 4.8.

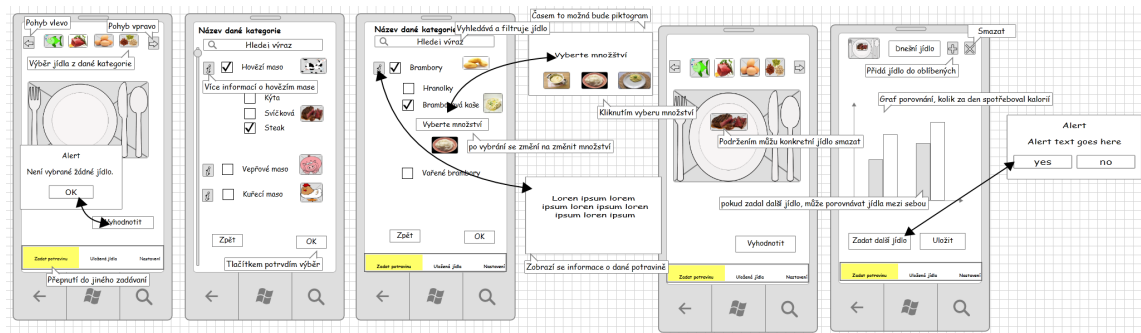
Toto rozhraní zobrazuje chybové hlášky na nepovolenou operaci, možnost zadat nový produkt, který v seznamu jídel není k nalezení, a možnost zvolení množství při přidání potraviny do seznamu vybraných jídel. Aplikace působila na uživatele příjemně, ale větší počet tlačítek uživatele mátl.



Obrázek 4.8: Zadávání jednoho produktu

## Prototyp D

Tento návrh má již konečnou podobu. Od předchozích návrhů má vyřešené posouvání kategorií do stran, náhled detailního zobrazení informace o daném prvku, mazání prvku z talíře, vybrání konkrétního množství, porovnávání prvků mezi sebou a vypisování upozorňujících hlášek.



Obrázek 4.9: Konečná podoba návrhu na uživatelské rozhraní aplikace

Na obrázku 4.9 je možné zhlédnout konečnou podobu návrhu uživatelského rozhraní této práce.

Na základě získané zpětné vazby uživatelů a vedoucího mé závěrečné práce jsem začala implementovat tento návrh rozhraní. Jak vypadá výsledná aplikace a jakými změnami v implementaci si prošla, je dále popsáno v kapitole 5.

## Kapitola 5

# Implementace a testování aplikace

Celá aplikace je postavena na třech návrhových vzorech [1], které jsou dále popsány v této kapitole. Při vývoji aplikace jsem vycházela z konkrétní agilní metodiky, konkrétně z iterativního vývoje. Pomocí této metodiky jsem vždy v rozmezí dvou týdnů vyvíjela prototypy aplikace, které jsem testovala na uživatelích. Hlavní myšlenkou iterativního vývoje je vyvinout prototyp aplikace s rostoucí sadou funkcí v určitém časovém rozmezí a otestovat ho na uživatelích. Často se stává, díky zpětné vazbě uživatelů, že se aplikace odvrátí od původního návrhu. Proto jsem se snažila navrhnout takový návrh aplikace, pomocí kterého bych se tomuto problému vyhnula. V této kapitole také přiblížím jednotlivé etapy vývoje aplikace. Podrobněji popíšu, co nového každá z etap přinesla, s jakými problémy jsem se setkala při testování aplikace na uživatelích a jaká byla jejich zpětná vazba. Dále se zmíním i o návrhu databáze včetně použitých technologií.

### 5.1 Návrhové vzory

Návrhový vzor se obecně dá definovat jako způsob popisu řešených problémů. Ukazuje jednotlivé vztahy a interakce mezi objekty a třídami, bez možnosti definice algoritmu, který by problém vyřešil. Zachycuje stejná a ověřená řešení návrhových problémů a urychluje práci při návrhu a vývoji aplikace. Návrhové vzory původně vycházejí z historické architektury<sup>1</sup>.

#### Model-View-Controller

Návrhový vzor Model-View-Controller, zkráceně MVC, odděluje logiku od výstupu. Dělí aplikaci na tři logické vrstvy. Na vrstvy **Model**, **View** a **Controller**. Tímto rozdělením aplikaci zpřehledňuje, usnadňuje a umožňuje testování jednotlivých částí zvlášť.

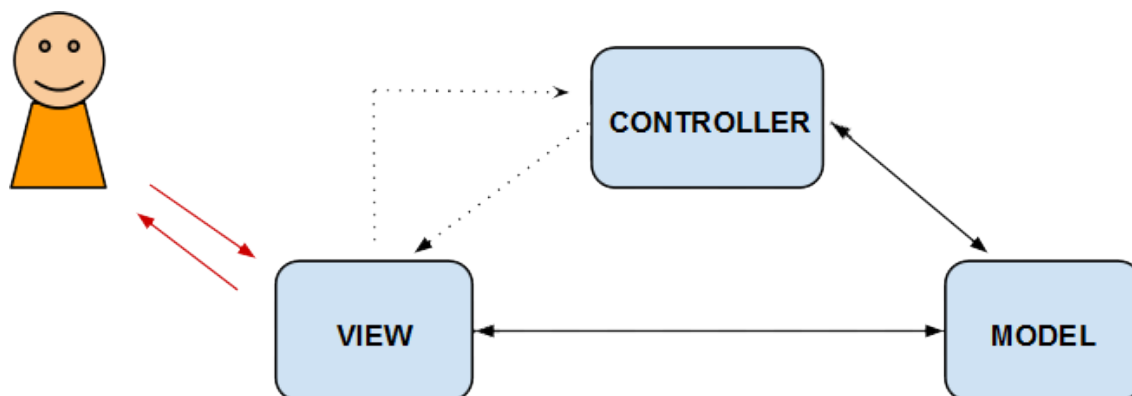
- **Model** - je datový a hlavně funkční základ celé aplikace. Obsahuje všechnu aplikační logiku aplikace. Jakákoli akce, jako jsou dotazy do databáze nebo výpočty, představuje akci modelu. Jeho funkce spočívá v přijetí parametrů zvenku a vydání dat ven. Změnu včetně zjišťování stavu lze docílit voláním funkcí tohoto rozhraní. Model neví o existenci View a Controlleru.

---

<sup>1</sup>První zmínka o návrhových vzorech se objevila v knížce *A Pattern Language* od Christophera Alexandra.

- **View** - View, nebo Pohled, představuje vrstvu, která se stará o zobrazení dat uživateli. Používá šablonovací systém a ví, jak se má daný výsledek získaný z modelu zobrazit.
- **Controller** - představuje prostředníka, se kterým komunikuje uživatel, model a view. Zpracovává požadavky uživatele a na jejich základě volá model a pak požádá view o vykreslení dat.

Na obrázku 5.1 je vidět životní cyklus aplikace. Uživatel zadá požadavek na vrstvu View. Zadání požadavku může představovat kliknutí na tlačítko. Vrstva View tento požadavek vezme a předá ho Controlleru. Controller pak může oslovit Model, který dále požadavek zpracuje, například vyhledá nějaký údaj v databázi a vrátí jeho výsledek. Controller pak ve výsledku zpracuje data z požadavku a předá je vrstvě View, která je vykreslí.



Obrázek 5.1: MVC životní cyklus aplikace.

## Model-View-View-Model

Návrhový vzor Model-View-View-Model, zkráceně MV-VM, odděluje logiku aplikace od uživatelského rozhraní. Prostřednictvím tohoto modelu se docílí kratšího a přehledného kódu. Tento vzor je zároveň vzorem pro WPF aplikace. MV-VM je postaven na vytvoření třídy, která uchovává stav aplikace. Tato třída se nazývá `ViewModel`. S třídou `ViewModel` komunikuje uživatelské rozhraní, které podle její specifikace vykresluje ovládací prvky, nebo naopak. Uživatel zadá nějakou informaci, která se poté automaticky přidá do této třídy. Ve WPF lze pomocí **data bindingu** propojit uživatelské rozhraní na třídu `ViewModel`. **Data binding** je důležitá vlastnost při vytváření XAML aplikací. Umožňuje navázat vlastnost XAML elementu na nějaký zdroj dat a zobrazit ho v dané XAML elementu. Navázání XAML elementu na nějaký zdroj dat lze udělat prostřednictvím **Bindingu**. Pokud bych chtěla navázat vlastnost `ItemSources` elementu `ListBox` na zdroj dat s názvem `PlateFoods`, vypadal by zápis následovně 5.1.

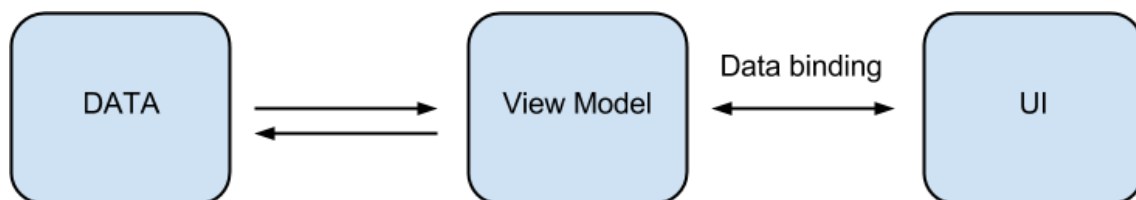
```

1 <ListBox x:Name="EnumItems" ItemsSource="{Binding Path=PlateFoods}" ItemTemplate="↔
  {StaticResource ListBoxFoods}" />
  
```

Zdrojový kód 5.1: Ukázka provázání XAML elementu se zdrojem dat.

Klíčové slovo **Binding** říká, že se daná vlastnost bude vázat na zdroj dat. Název vlastnosti, na kterou se váže, představuje slovo `PlateFoods`. Dále tu je samotný zdroj dat, kterým je `ListBoxFoods`.

Návrhový vzor MV-VM je ke zhlédnutí na obrázku 5.2.



Obrázek 5.2: Diagram návrhového vzoru MV-VM.

## Observer

Observer je návrhový vzor, který je použit všude v aplikaci. Obecně se dá popsat jako posluchač (nějaký program), který "sedí" na nějakém objektu a čeká na příjem nějaké události (event). Událostí může být kliknutí na nějaké tlačítko nebo stisknutí klávesy. Pokud posluchač přijme takovou událost, zavolá předem nadefinovanou funkci, která provede určitou akci. Jinými slovy pokud nastavím na nějakém tlačítku posluchače, aby při události `Click` zavolal funkci `ResultGraphBtn_Click`, bude to ekvivalent zápisu 5.2

```
1 <shell:ApplicationBarIconButton x:Name="ResultGraphBtn" IconUri="/icons/appbar.↔  
check.rest.png" Text="Vyhodnotit" Click="ResultGraphBtn_Click"/>
```

Zdrojový kód 5.2: Ukázka Observeru v praxi.

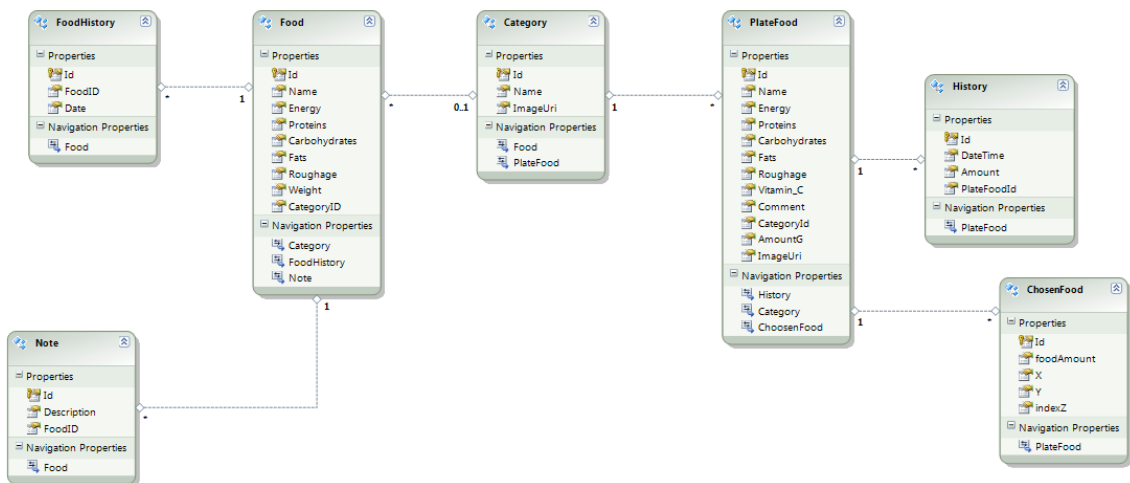
Kde jako výsledek funkce `ResultGraphBtn_Click` se zobrazí nová stránka s grafy pro vybraná jídla na talíři.

## 5.2 Návrh úložiště dat

Aplikace má k dispozici interní databázi, jejíž struktura je uložena v souboru `Database.sdf`. Pro přístup k jednotlivým tabulkám je použit jazyk `LINQ` 5.3. Pro vygenerování databázové struktury jsem použila `Entity Framework` 5.3. Struktura databáze se nachází ve jmenném prostoru `FoodTracker.WP.Models`. V něm je možné nalézt všechny třídy popisující jednotlivé tabulky. O připojení k databázi se stará třída `Database`. Tato třída se stará o inicializaci a aktualizaci základních dat v databázi. Ta jsou do tabulek vložena prostřednictvím třídy `DataContext`. Návrh databáze je vidět na obrázku 5.3.

Tabulky `Category`, `PlateFood` a `ChosenFood` využívám pro ukládání a výpis dat. Definice všech tabulek lze najít v souboru `FoodTrackerDataContext`. Struktura tabulek `texttt-Category`, `PlateFood` a `ChosenFood`:

- V tabulce **Category** jsou uloženy názvy a obrázky vystihující danou kategorii aplikace. Její atributy jsou:

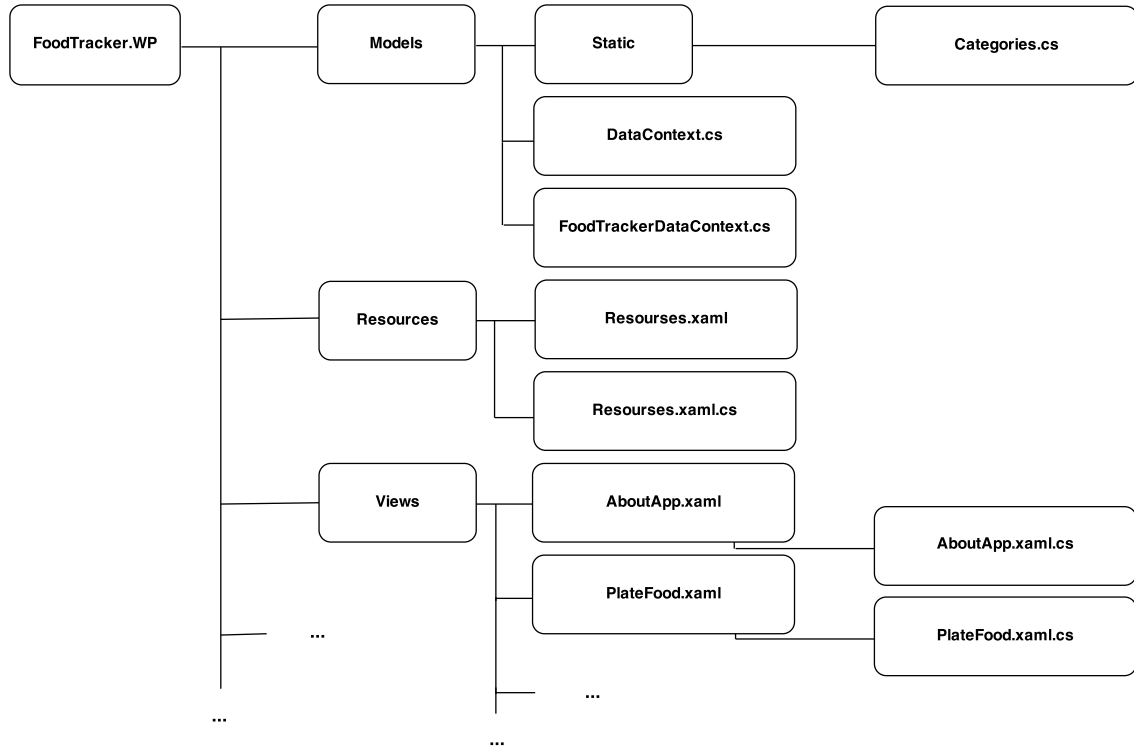


Obrázek 5.3: Databáze aplikace FoodTracker.WP

- *Id*
- *Name*
- *ImageUri*
- V tabulce **PlateFood** jsou uložena jídla včetně nutričních hodnot. Její atributy jsou:
  - *Id*
  - *Name*
  - *Energy*
  - *Proteins*
  - *Carbohydrates*
  - *Fats*
  - *Roughage*
  - *Vitamin\_C*
  - *Comment*
  - *CategoryId*
  - *AmountG*
  - *ImageUri*
- V tabulce **ChosenFood** jsou uložena jídla, které si uživatel vybral z dané kategorie na talíř. Dále sem ukládám i velikost porce jídla, umístění jídla na talíři a hloubku polohy jídla. Atributy tabulky jsou:
  - *Id*
  - *foodAmount*
  - *X*
  - *Y*
  - *indexZ*

## 5.3 Použité technologie

Struktura aplikace je rozdělena na dílčí jmenné prostory (namespace), viz obrázek 5.4.



Obrázek 5.4: Pohled na rozdělené dílčí jmenné části stromové struktury aplikace.

Jmenný prostor `FoodTracker.WP.Models` popisuje data, se kterými aplikace pracuje, nachází se zde datové modely. Ve jmenném prostoru `FoodTracker.WP.Resources` jsem si nadefinovala šablonu používanou pro komponenty z `FoodTracker.WP.Views`. Šablonu používám například pro výpis jídel v dané kategorii aplikace. Příklad výpisu jídel aplikace je zobrazen ve zdrojovém kódu 5.3.

```
1 <DataTemplate x:Key="ListBoxFoods">
2   <Border Padding="10">
3     <StackPanel Orientation="Horizontal" HorizontalAlignment="Left">
4       <Image Source="{Binding Path=ImageUri}" Width="50" Height="50" Margin="↔
5         0,0,20,0" />
6       <TextBlock Text="{Binding Path=Name}" TextWrapping="Wrap" d:LayoutOverrides=↔
7         "Width, Height" VerticalAlignment="Stretch" FontSize="29.333" FontWeight=↔
8         "Bold" Margin="0,10,0,0" HorizontalAlignment="Right" FlowDirection="↔
          RightToLeft" />
9     </StackPanel>
10  </Border>
11 </DataTemplate>
```

Zdrojový kód 5.3: Výpis jídel pro danou kategorii.

Jmenný prostor `FoodTracker.WP.Views` obsahuje šablony a kontrolery. Reprezentuje



uživatelské rozhraní v jazyce XAML. Může jít o okno aplikace nebo ovládací prvek. Bývá označován také jako UI nebo prezentační vrstva.

## Entity Framework

K vytvoření datového modelu aplikace je použitý Entity Framework. Je to objektově relační mapovací framework, který umožňuje pracovat s relačními daty jako s objekty a generovat databázové schéma za pomoci grafických nástrojů a diagramů. Zároveň s jeho pomocí lze spolupracovat s technologií LINQ, která umožňuje psát dotazy do databáze s podobnou syntaxí, jako má jazyk SQL<sup>2</sup>. Entity Framework umožňuje více se soustředit na aplikační logiku aplikace. Přístup k datům, tvorbu cizích klíčů, vazeb mezi tabulkami je možné generovat pomocí vazeb v grafickém rozhraní **Entity Designer Database Generation Power Pack**. Po dokončení návrhu databáze stačí vytvořit soubor s ve formátu "sdf", který uchovává data a jejich strukturu. V tomto souboru je následně vytvořeno databázové schéma. Soubor ve formátu sdf je možné pomocí nástroje SQL Metal převést do zdrojového kódu v jazyce C#. Pro platformu Windows Phone bohužel není výstup vygenerovaný nástrojem SQL Metal oficiálně kompatibilní. Kompatibilitu výsledného kódu je ale možné zajistit prostým smazáním dvou nadbytečných konstruktorů. K takové databázi je možné elegantně přistoupit pomocí jazyku LINQ.

Entity Framework je vyvinutý pro ADO.NET<sup>3</sup>, jenž je součástí .NET frameworku.

## LINQ

LINQ<sup>4</sup> je integrován do syntaxe jazyka C# a je součástí .NET frameworku již od verze 3.5. LINQ je deklarativní jazyk, jehož syntaxe je velmi podobná jazyku SQL. Dotazy do databáze se zapisují ve formě lambda výrazů. Výhodou integrace LINQ do jazyka C# je odhalení chyb při kompilaci programu. Ukázka dotazu při výběru kategorie z databáze je v kódu 5.4.

```
1 CategoryId = this.NavigationContext.QueryString["category"];
2 Category Category = (from row in Database.Categories where row.Id == System.<←
   Convert.ToInt32(CategoryId) select row).First();
3
4 //vybrana kategorie je ta, co jsem nasla v DB
5 Menu.SelectedItem = Category;
```

Zdrojový kód 5.4: Výběr kategorie z databáze.

## 5.4 Etapy vývoje aplikace

V této části kapitoly přiblížím vývoj aplikace. Na obrázcích aplikace ukážu změny oproti předchozím verzím vytvořených prototypů a shrnu výsledky testování na uživateli.

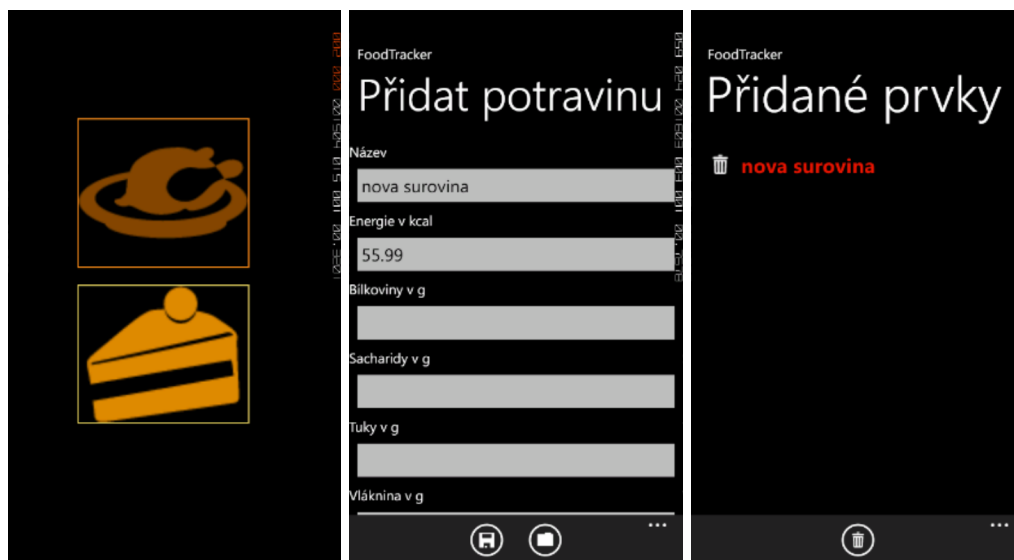
<sup>2</sup>Structured Query language, zkráceně SQL, je strukturovaný dotazovací jazyk, který se používá pro práci s daty v relační databázi.

<sup>3</sup>ADO.NET neboli Microsoft ActiveX Data Objects .NET představuje množinu tříd nabízejících služby pro přístup k datům a tvorbu databázových aplikací.

<sup>4</sup>LINQ neboli Linq Integrated Query je jazyk, který dokáže pracovat s entitami databáze jako s objekty.

## První etapa

S touto etapou jsem začala začátkem měsíce října 2013. Snahou bylo seznámit se s prostředím nástroje Visual Studio 2010 a s jazykem C#, v kterém jsem až do nedávna žádnou aplikaci neimplementovala. Po několika prvních pokusech vznikl první prototyp aplikace, jehož úvodní obrazovka je vidět na obrázku 5.5.



Obrázek 5.5: První etapa: Úvodní obrazovka ve smyslu "rozcestníku", přidání nového prvku a seznam všech přidaných prvků.

Umožňovala vybrat si mezi přidáním jednoho nového jídla a vybráním jídla z kategorie a následné přidání do seznamu vybraných jídel. Při přidání nového jídla (obr.5.5) si uživatel musel pamatovat nutriční hodnoty potraviny. Navíc aplikace neměla vlastní databázi, do které by se nově přidané prvky ukládaly a které by v aplikaci telefonu zůstaly i při nové instalaci. V tomto návrhu se ukládaly pouze do interní databáze telefonu, což ve výsledku bylo neefektivní, uživatel by musel stejný prvek přidat nově.

Při zvolení druhé možnosti zadání jídla se uživatel dostal na stránku s názvem "Talíř jídla", viz obrázek 5.7. V ní se nacházel první prototyp rozdělení typu jídel podle kategorií. Při výběru konkrétní kategorie jídel je používán soubor `Category.cs`, ve kterém je definovaná třída `Category`, která uchovává obrázek a název vybrané kategorie. Názvy kategorií včetně obrázků jsou ukládány do databáze umístěné mimo telefon. Po výběru určité kategorie se zobrazí příslušná stránka se seznamem jídel. Hodnota předávaná v url adrese je porovnávána s hodnotou umístěnou v databázi, pokud se najde jejich shoda, daná stránka je vykreslena. Získání hodnoty z url adresy a předání stránky je vidět ve zdrojovém kódu 5.5.

```

1 protected override void OnNavigatedTo(System.Windows.Navigation.↵
    NavigationEventArgs e)
2 {
3     base.OnNavigatedTo(e);
4     string CategoryId;
5
6     if (this.NavigationContext.QueryString.ContainsKey("category"))
7     {
8         //získám hodnotu z uri adresy
9         CategoryId = this.NavigationContext.QueryString["category"];
10        Category Category = (from row in Database.Categories where row.Id == ↵
            System.Convert.ToInt32(CategoryId) select row).First();
11        //vybrána kategorie je ta, co jsem našel v DB
12        Menu.SelectedItem = Category;
13    }
14 }

```

Zdrojový kód 5.5: Získání hodnoty z url adresy a její předání do Menu.

Stránka je implementovaná pomocí základní komponenty **Pivot**. Prostřednictvím této komponenty se na stránce zobrazuje několik sekcí (kategorií) vedle sebe a uživatel se na ně může podle libosti přesouvat. Komponenta je rovněž přizpůsobena velkému množství dat a zaplňuje celou šířku stránky. V následujícím zdrojovém kódu 5.6 je vidět jeho implementace.

```

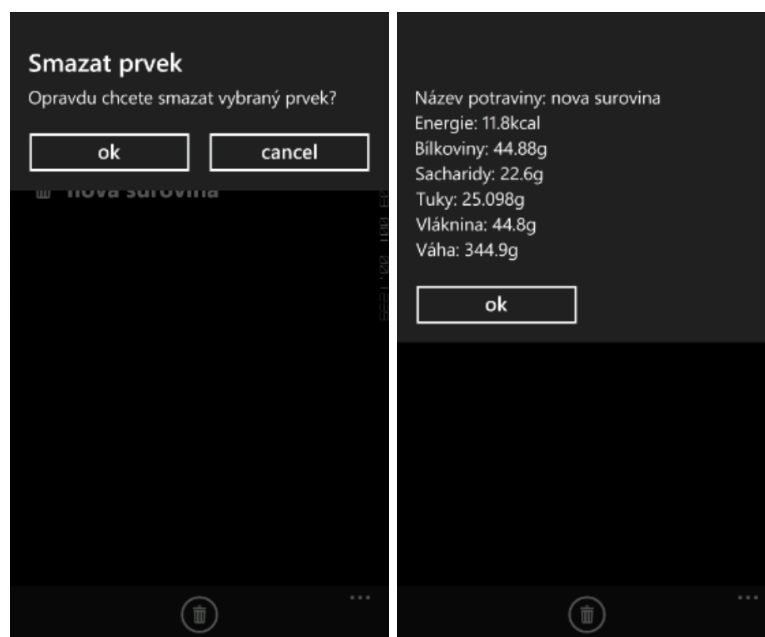
1 <StackPanel>
2     <ListBox ScrollViewer.VerticalScrollBarVisibility="Disabled" ScrollViewer.↵
        HorizontalScrollBarVisibility="Auto" Name="Categories" BorderThickness="↵
        0,0,5,0" SelectionChanged="Categories_SelectionChanged">
3         <ListBox.ItemsPanel>
4             <ItemsPanelTemplate>
5                 <StackPanel Orientation="Horizontal"/>
6             </ItemsPanelTemplate>
7         </ListBox.ItemsPanel>
8
9         <ListBox.ItemTemplate>
10            <DataTemplate>
11                <Border Padding="10">
12                    <StackPanel>
13                        <Image Source="{Binding Path=ImageUri}" Width="100" Height="80" Margin↵
                            ="0" />
14                        <TextBlock Text="{Binding Path=Name}" FontSize="32" ↵
                            HorizontalAlignment="Center" />
15                    </StackPanel>
16                </Border>
17            </DataTemplate>
18        </ListBox.ItemTemplate>
19    </ListBox>
20 </StackPanel>

```

Zdrojový kód 5.6: Komponenta Panorama.

Vybrané jídlo se poté zobrazí na předchozí stránce, kde je umístěno do seznamu jídel a lze ho odebrat z tohoto seznamu jednoduchým kliknutím.

Testování tohoto prototypu proběhlo pouze na pár nejbližších přátel a mé rodině, kdy do testování se zapojila i babička. Na základě jejich zpětné vazby jsem započala s druhou etapou vývoje aplikace. Výsledkem této etapy je jakási kostra, která obsahuje spoustu ovládacích prvků navíc a která se v dalším vývoji zdokonaluje. Rovněž zde vznikl první prototyp zadávání jídla na talíř a vybralo se jméno pro tuto aplikaci - **FoodTracker.WP**.



Obrázek 5.6: První etapa: Notifikace při smazání a detail nového prvku.

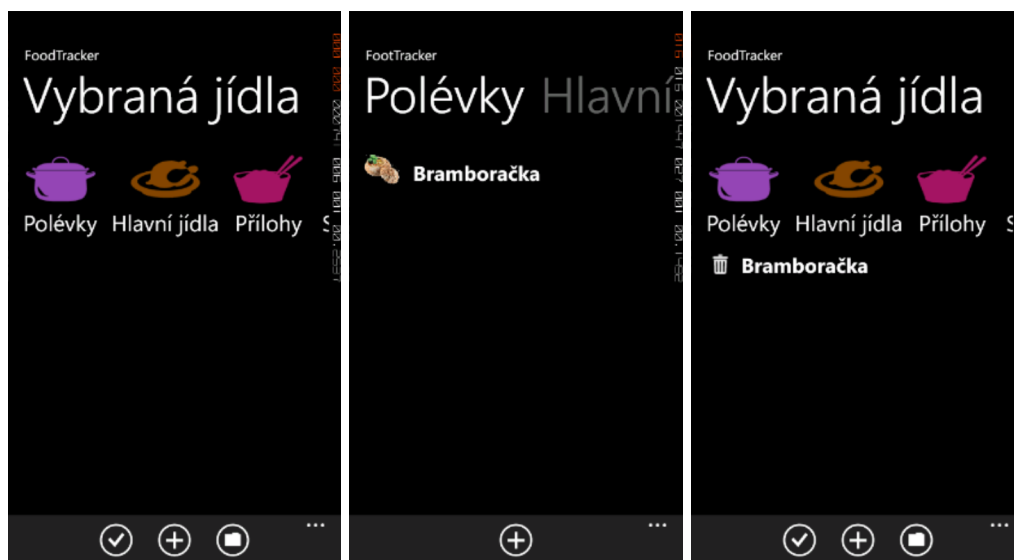
Na obrázku 5.6 lze zhlédnout první notifikaci při smazání jídla a výpis všech přidanych prvků, u kterých se po kliknutí na určitý z nich zobrazil detail s nutričními hodnotami.



Obrázek 5.7: První etapa: První verze obrazovky talíře a seznam jídel v kategorii, realizovanou komponentou panorama.

## Druhá etapa

V této etapě jsem si kladla za cíl vytvořit první uživatelsky příjemnou aplikaci, která nebude mít přebytečné funkce. Zároveň jsem se snažila co nejvíce se soustředit na jedno rozhraní zadávání jídel na talíř. V tomto prototypu aplikace jsem odebrala úvodní obrazovku, která představovala "rozcestník", protože na základě získané zpětné vazby se uživatelům zdál příliš matoucí a zbytečný. Dále jsem z aplikace odebrala možnost přidání nové potraviny a zaměřila se více na zadávání jídla na talíř.



Obrázek 5.8: Druhá etapa: Úvodní obrazovka talíře, seznam vybraných jídel s ikonami.

Při spuštění aplikace se zobrazila úvodní stránka talíře 5.8. Ve vybraných kategoriích se nachází seznam testovacích dat s obrázky, bez možnosti zobrazení náhledu detailu. Po zvolení prvku došlo k jeho umístění do seznamu vybraných jídel na předchozí stránce. Ikona charakterizující "koš" umožňuje smazat vybraný prvek. Seznam jídel s ikonami obrázků ve vybrané kategorii a seznam vybraných jídel na talíři je vidět na obrázku 5.8

Informace o konkrétním jídle jsou ukládány do vlastní databáze, což se ve výsledku projevuje jako výhoda při opětovné instalaci nového prototypu aplikace. V případě předčasného ukončení aplikace, je program schopen si pamatovat poslední známou činnost.

Během testování prototyp na uživatele působil přívětivěji než předchozí. Na základě zpětné vazby vedoucího mé bakalářské práce jsem se v další etapě zaměřila na vizualizaci uživatelského prostředí.

## Třetí etapa

Z testování a zpětné vazby vedoucího mé závěrečné práce vzešla řada podnětů na další vývoj aplikace. Obdobně jako u dotazníku, tak i zde jsem vybrané podněty shrnula do několika bodů:

- Více grafických prvků.
- Nechat uživatele rychle si sestavit vlastní jídlo, které se co nejvíce podobá reálnému jídlu na talíři.
- Intuitivní zobrazení a změna jídla (myšlena vlastní dávka porce, velikost jídla).
- Zobrazení nutričních hodnot každého jídla zvlášť, hlavně u grafů.
- Detailnější informace o zobrazeném jídlu.

Ostatní podněty jsem se rozhodla odložit na budoucí možný vývoj aplikace a více se raději zaměřit na tento seznam požadavků.

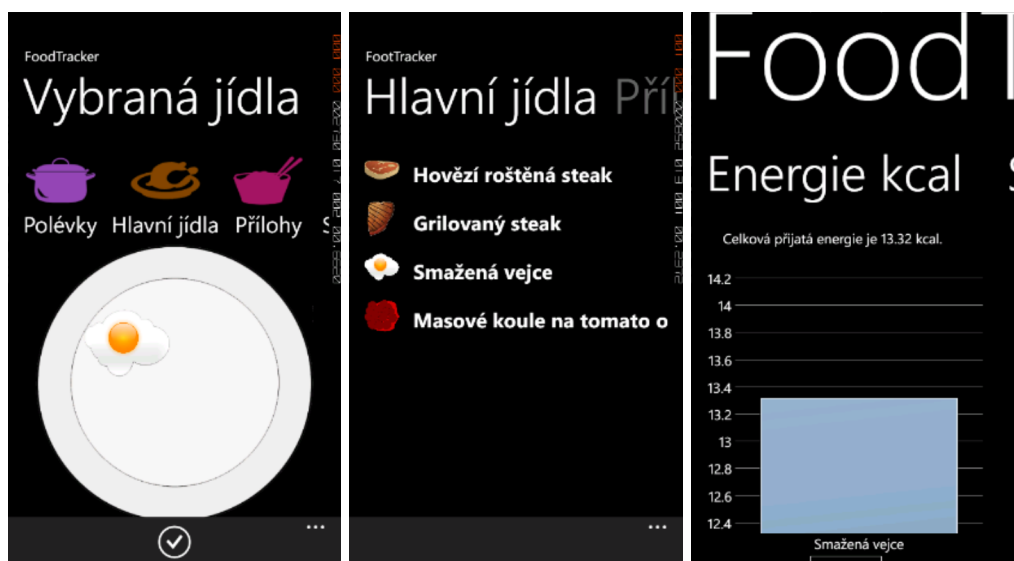
Jak je patrné z obrázku 5.9, došlo k menším změnám v grafice uživatelského rozhraní. Na úvodní obrazovce, kterou implementuje třída `PlateFood`, přibyl obrázek představující reálný talíř. Tomu předcházelo odstranění seznamu vybraných jídel, která se zde nově zobrazují v podobě obrázků. Vybraná jídla jsou náhodně rozmístovaná po talíři. Jejich náhodné zobrazení umožňuje zobrazovací prvek `Canvas`. Je umístěný v prostoru talíře, kde má definované i své rozměry. Vybraná jídla, jsou umístěna do tabulky `ChosenFood`, odkud jsou předávána `Canvasu` k jejich zobrazení na talíři. Obrázky jídel jsou vykreslovány v bitmapové podobě. Vykreslení obrázku jako bitmapa je vidět v kódu 5.7.

```
1 Image imgFood = new Image();
2 BitmapImage bitmapImg = new BitmapImage();
3 bitmapImg.UriSource = new Uri(item.PlateFood.ImageUri, UriKind.Relative);
4 //zobrazení jako bitmap obr
5 imgFood.Source = bitmapImg;
```

Zdrojový kód 5.7: Výběr kategorie z databáze.

Přibyl zde i graf udávající celkovou energii vybraných jídel. Graf je ke zhlédnutí na obrázku 5.9.

Jiné funkce tento prototyp neposkytoval. Výsledkem testování byla snaha uživatele přemístit náhodně zobrazený prvek na jiné místo, než je původní. Při kliknutí na daný prvek uživatel čekal, že se vykoná nějaká akce, například smazání prvku. Což mě donutilo více se zaměřit na sestavení jídla na talíři a prozatím opomenout další funkce, jako je detail prvku nebo graf.



Obrázek 5.9: Třetí etapa: Úvodní obrazovka, seznam jídel a graf přijaté energie.

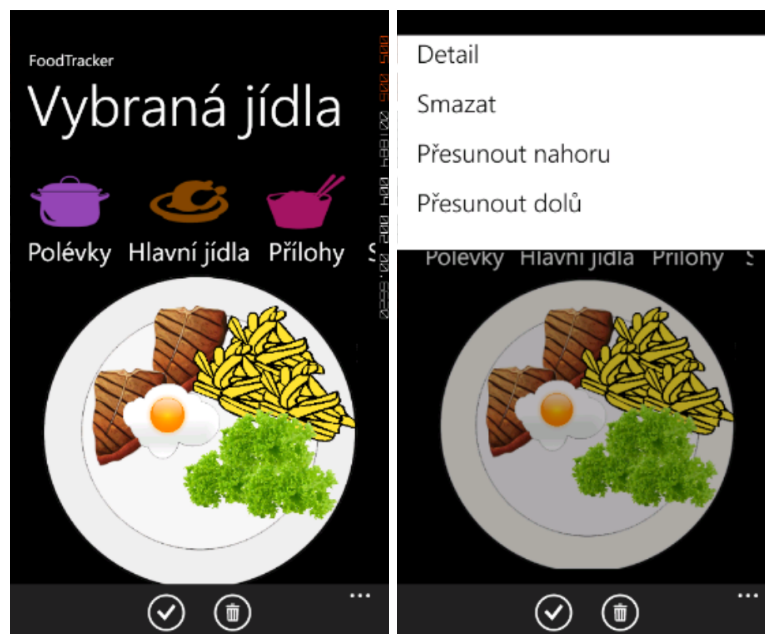
### Čtvrtá etapa

Tato etapa přinesla mnoho vylepšení oproti předchozím verzím. Představuje současnou podobu této práce. Ze třetí etapy vzešla celá řada podnětů, z nichž převážná část se implementovala do současné verze aplikace. Jak je vidět na obrázku 5.10, seznam jídel, který



Obrázek 5.10: Čtvrtá etapa: Rozšířený seznam jídel

implementuje třída `PivotPageList`, u každé kategorie se rozšířil o další prvky. Při výběru se jídlo zobrazí na talíři jako obrázek, který už není náhodně rozmístován. Jednotlivé prvky lze po talíři přesouvat pomocí funkce `onManipulationDelta()`. Při změně místa je ukládána do databáze nová pozice prvku. Prvkům na talíři lze měnit velikost, která zde představuje množství porce jídla, prostřednictvím funkcí `GestureListener_PinchStarted()` a `OnPinchDelta()`. Kliknutím na některý z prvků či při dlouhém podržení se zobrazí menu s dalšími možnostmi, viz obrázek 5.11. Vybranému prvku lze upravit hloubku třetí souřadnice "Z" ve 2D prostoru tak, aby bylo možné prvky vzájemně překrývat.



Obrázek 5.11: Čtvrtá etapa: Další nabídka možností ve vysouvacím menu.

Na obrázku 5.12 lze zpozorovat stránku s informacemi o vybraných jídlech, ale také oznamovací zprávu s možností smazání vybraného prvku. Stránka s dalšími informacemi o vybraných jídlech je implementovaná pomocí třídy `ListOfFood`. Na obrázku 5.13 je vidět stránka zobrazení výsledků ve formě grafů, kde má uživatel možnost zobrazit graf s nutričními hodnotami každého jídla zvlášť. Stránka je implementovaná pomocí komponenty `Panorama`, která je obdobou komponenty `Pivot`. Jednotlivé sekce jsou umístěny vedle sebe a uživatel se může libovolně mezi nimi pohybovat. Zobrazené grafy jsou implementované pomocí třídy `Charting` pro technologii .NET.

Implementace výše zmíněných grafů je ukázána ve zdrojovém kódu 5.8.



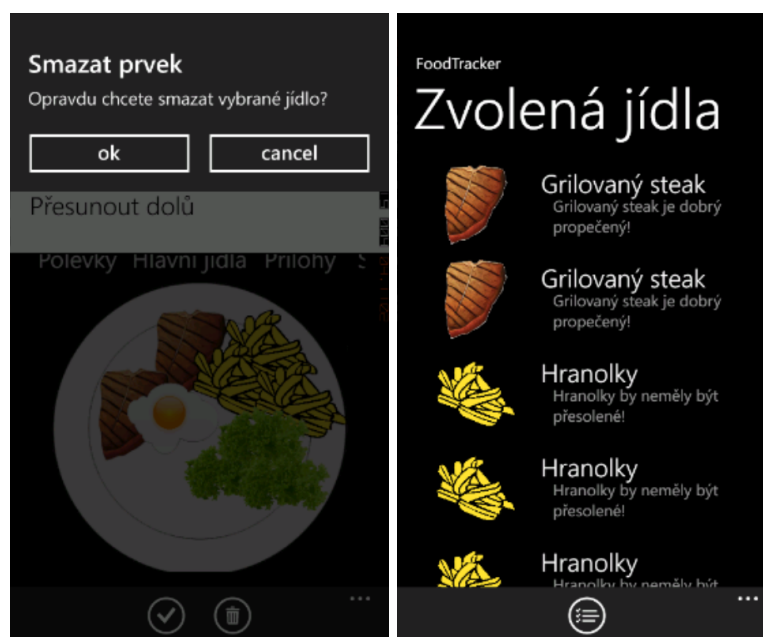
```

1 <cht:Chart x:Name="proteinsPie" Background="#FF73EB7C" Style="{StaticResource ↵
   PhoneChartStyle}" Template="{StaticResource PhoneChartPortraitTemplate}" Height↵
   ="498" VerticalAlignment="Top">
2 <cht:Chart.Series>
3 <cht:PieSeries Title="Vapnik" IndependentValueBinding="{Binding Path=Key}" ↵
   DependentValueBinding="{Binding Path=Value}" >
4 <cht:PieSeries.LegendItemStyle>
5 <Style TargetType="cht:LegendItem">
6 <Setter Property="Margin" Value="5 0 5 0"/>
7 </Style>
8 </cht:PieSeries.LegendItemStyle>
9 </cht:PieSeries>
10 </cht:Chart.Series>
11 </cht:Chart>

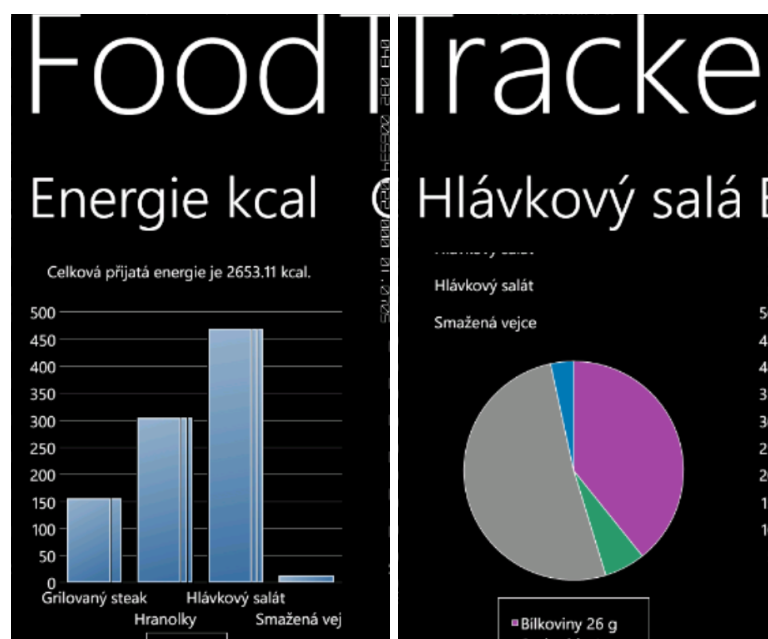
```

Zdrojový kód 5.8: Komponenta Panorama.

Testování jsem prováděla individuálně s každým uživatelem zvlášť. Přitom jsem zvolila klidné prostředí, kde se uživatel necítil nervózní a vyrušovaný. Jednotlivým uživatelům jsem poté zadala přesné jídlo, které měli k obědu, a to měl poté uživatel sestavit na talíř. Výsledkem testování bylo, že uživatel našel, co hledal, rychle a bez pomoci si jídlo snadno upravil. Valná většina testovaných subjektů by uvítala další vývoj této aplikace a její používání. Bohužel jsem se setkala i s takovými, kteří takovouto aplikaci nechtějí používat, neboť mají k dispozici jinou aplikaci, na kterou jsou zvyklí. Ale podle jejich zpětné vazby se jim tato aplikace zamlouvala.



Obrázek 5.12: Čtvrtá etapa: Stránka s informacemi o vybraných jídlech a varující zpráva o trvalého smazání vybraného jídla.



Obrázek 5.13: Čtvrtá etapa: Komponenta Panorama je použita jako stránka s grafy.

## 5.5 Zhodnocení a budoucnost aplikace

Aplikace FoodTracker.WP splňuje všechny mé očekávané vlastnosti, které jsem pro ni zamýšlela. Při testování na subjektech sklídila kladný ohlas, co se týče jejího používání. Proto bych se ráda odpoutala od tohoto základního návrhu a v budoucnu bych chtěla využít zpětné vazby uživatelů na vytvoření aplikace, která by měla šanci na komerční používání. V současné době mám plnou hlavu nápadů, jak aplikaci rozšířit o další funkce. Jako příklad bych uvedla vytvoření takového prototypu aplikace, který by uživateli umožnil ukládat jeho oblíbené restaurace, na jehož základě by si aplikace stáhla jejich denní nabídku jídel. Tato jídla by mezi sebou porovnála a jejich výsledky by poskytla uživateli. Rovněž by tento prototyp aplikace nabízel možnost si zvolit jeden z hlavních chodů, který by bylo možné uživatelem upravit do požadované podoby jako jídlo na jeho reálném talíři, které mu v restauraci donesli. Mimo jiné bych se ráda věnovala dalšímu možnému vývoji této práce i po složení státní závěrečné zkoušky.

## Kapitola 6

### Závěr

Cílem této práce bylo navrhnout jednoduché, rychlé a uživatelsky nenáročné rozhraní, které by umožňovalo zadávání a výběr jídel. V průběhu práce se mi povedlo vytvořit několik konceptuálních návrhů uživatelského rozhraní a specifikovat požadavky uživatelů na vývoj aplikace. Hlavní přínos této práce vidím v možnosti nechat uživatele si sestavit grafickou podobu reálného jídla, u níž je možná interaktivní změna velikosti porce jídla a v poskytnutí kvalitních informací s přesností 20%. Dalším přínosem bylo i testování na dobrovolných subjektech, kteří s výslednou aplikací byli spokojeni. Faktory, které ovlivňovaly implementaci celé práce, byly požadavky testovaných uživatelů, zpětná vazba vedoucího mé závěrečné práce a jistá omezení operačního systému Windows Phone. Jsem však ráda, že jsem tuto práci mohla vyvíjet právě pod touto platformou, neboť ve spolupráci s technologií .NET disponuje různými knihovnamí a technikami implementace již řešených problémů, které mi byly v této práci velice nápomocné. Aplikace je vhodná pro každého uživatele se zájmem o vlastní životosprávu. V této práci jsem se nejvíce zaměřovala na interaktivní upravování jídla umístěného na talíři, kdy jsem se snažila nejvíce ztvárnit jeho vizuální podobu s reálným jídlem. Výsledkem této práce je rychlá, intuitivní, jednoduchá a výkonově nenáročná aplikace pro Windows Phone, která nemá téměř žádnou konkurenci.

# Literatura

- [1] Bishopová, J.: *C# návrhové vzory*. Zoner Press, první vydání, 2010, iISBN 978-80-7413-076-2.
- [2] Fendrych, A.: User Experience – poznejte své uživatele [online]. <http://www.lupa.cz/clanky/user-experience-poznejte-sve-uzivatele/>, 2014-04-06 [cit. 2010-06-08].
- [3] Krug, S.: *Don't make me think! A common Sense Approach to Web Usability*. New Riders, druhé vydání, 2006, iISBN 0-321-34475-8.
- [4] Microsoft: Historie aktualizací Windows Phone 7 [online]. <http://www.windowsphone.com/cs-cz/how-to/wp7/basics/update-history>, 2014-04-06 [cit. 2014-04-06].
- [5] Noble, S.; Bourton, S.; Jones, A.: *WPF Recipes in C# 2008, A Problem-Solution Approach*. Computer Press, první vydání, 2008, iISBN 978-1-4302-1084-9.
- [6] Papík, R.: *Vyhledávání informací II. Uživatelské rozhraní a vlivy oboru „human-computer interaction“*. Národní knihovna ČR, druhé vydání, 2001, 81-90 s., iISSN 1214-0678.
- [7] Pavlíček, M.: Android a Windows Phone v Evropě rostou, iOS naopak ztrácí [online]. <http://mobilenet.cz/clanky/android-a-windows-phone-v-evrope-rostou-ios-naopak-ztraci-15231>, 2014-04-01 [cit. 2014-04-01].
- [8] Petzold, C.; Mikulaščík, J.; Farný, J.: *Mistrovství ve Windows Presentation Foundation*. Computer Press, první vydání, 2011, iISBN 978-80-251-2141-2.
- [9] Saffer, D.: *Designing for Interaction: Creating Innovative Applications and Devices*. New Riders, druhé vydání, 2010, iISBN 978-0-321-64339-1.
- [10] Shneiderman, B.: *Designing the user interface: strategies for effective human-computer interaction*. Addison-Wesley, Čtvrté vydání, 2010, iISBN 978-0-321-53735-5.
- [11] Součková, M.: *Aspekty vztahu člověk-počítač s důrazem na uživatelské rozhraní [Diplomová práce]*. Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví, první vydání, 2003, 89 s.

# Příloha A

## Obsah DVD

Příložené DVD obsahuje:

- `video/` - Video demonstrující návod k ovládní aplikace.
- `plakat/` - Plakát pro prezentaci aplikace.
- `BP-xkrato11.pdf` - Text bakalářské práce ve formátu PDF.
- `report/` - Zdrojové kódy bakalářské práce v TEX.
- `report/fig/` - Obrázky.
- `src/` - Zdrojové kódy aplikace v C#.