

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

POKROČILÝ BUDÍK PRO WINDOWS PHONE 7

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JOSEF ZAPLETAL

BRNO 2012



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

POKROČILÝ BUDÍK PRO WINDOWS PHONE 7

ADVANCED ALARM CLOCK FOR WINDOWS PHONE 7

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JOSEF ZAPLETAL

VEDOUcí PRÁCE

SUPERVISOR

Doc. Ing. ADAM HEROUT, Ph.D.

BRNO 2012

Abstrakt

Práce popisuje vývoj mobilní aplikace pro platformu Windows Phone 7 zaměřené na spánek a následné probuzení uživatele. Aplikace je velice jednoduchá na ovládání a ke své činnosti využívá vestavěných modulů moderních mobilních telefonů, jako je například kamera s vysokým rozlišením a akcelerometr. Práce popisuje hlavní rysy platformy Windows Phone 7, analýzu konkurenčních řešení, samotný návrh vlastní aplikace a její implementaci. Dále tato práce popisuje řešení různých problémů při vývoji a také uvádí příklady použitých metod a nástrojů. V neposlední řadě je zde také uveden seznam použitých knihoven.

Abstract

This bachelor thesis describes the development of mobile application for Windows Phone 7 focused on sleep and awakening the user. The application is very easy to use and uses built-in modules of advanced mobile phones, such as high resolution camera and the accelerometer. This thesis describes the main features of Windows Phone 7 platform, analysis of competing solutions, application concept and its implementation. Furthermore, this work describes the solution of various problems in the development and also provides examples of methods and tools. Finally, there is also a list of used libraries.

Klíčová slova

Windows Phone 7, WP7, budík, hodiny, spánek, postel, mobilní aplikace, zdřímnutí, Silverlight, C#

Keywords

Windows Phone 7, WP7, alarm, clock, sleep, bed, mobile application, nap, Silverlight, C#

Citace

Josef Zapletal: Pokročilý budík pro Windows Phone 7, bakalářská práce, Brno, FIT VUT v Brně, 2012

Pokročilý budík pro Windows Phone 7

Prohlášení

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

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

.....
Josef Zapletal
16. května 2012

Poděkování

Chtěl bych poděkovat svému vedoucímu doc. Ing. Adamovi Heroutovi, Ph.D. za věnovaný čas a také za cenné rady, které mi při tvorbě této práce poskytl.

© Josef Zapletal, 2012.

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

1 Úvod	2
2 Windows Phone 7	3
2.1 Hardwarová konfigurace	3
2.2 Styl Metro	5
2.3 Programové prostředí	5
2.4 Životní cyklus aplikace	7
2.5 Marketplace	9
2.6 Budoucnost WP7	10
3 Analýza existujících řešení	13
4 Návrh aplikace	17
4.1 ZXing Barcode Scanning Library	19
4.2 Coding4Fun Windows Phone Toolkit	19
4.3 Windows Phone 7 Toolkit Controls	19
5 Implementace aplikace Clever Alarm	21
5.1 Akcelerometr	24
5.2 Nastavení budíku	24
5.3 Řešení rovnice	27
5.4 Zdřímnutí	27
5.5 QR kód	28
5.6 Hlavní a čekací obrazovka	29
5.7 Testování	31
6 Marketing, hodnocení a budoucnost	32
7 Závěr	35
A Obsah DVD	38

Kapitola 1

Úvod

Mnoho lidí má problémy ráno vstát z postele do práce nebo třeba do školy. Večer si před spaním nastaví budík a ráno jakmile začne zvonit, tak jej pouze vypnou a spí dál. Často se tedy stane, že zaspí. Díky tomu přijdou pozdě na důležitou schůzku nebo test a to může být velký problém.

Cílem této bakalářské práce je vytvořit aplikaci, která donutí uživatele vstát z postele. Výsledná aplikace bude vyvíjena pro platformu Windows Phone 7 a bude co nejvíce využívat možností tohoto operačního systému a hardwarových vlastností telefonu. Aplikace bude obsahovat co nejjednodušší a uživatelsky přívětivé grafické rozhraní. Dále bude nabízet několik různých pokročilých funkcí, které by měly donutit uživatele vstát z postele.

Samotný text této práce je rozdělen do několika kapitol. Popis operačního systému Windows Phone 7 a základní specifikace vývoje aplikací pro tuto platformu jsou uvedeny v kapitole 2. Kapitola 3 analyzuje existující řešení a v kapitole 4 je popsán návrh výsledné aplikace a také použité knihovny při vývoji. V kapitole 5 je popsána vlastní funkčnost aplikace, její grafická podoba, samotná implementace a průběh testů. Kapitola 6 se zabývá marketingem, propagací a budoucností aplikace. Poslední kapitola shrnuje celý vývoj aplikace a dosažené výsledky.

Kapitola 2

Windows Phone 7

Platforma Windows Phone 7 (dále jen WP7) od firmy Microsoft je poměrně mladou mobilní platformou. Byla představena 11.10.2010. Nejedná se o další verzi Windows Mobile 6.5, ale WP7 se vydává jinou cestou. Na rozdíl od platformy Windows Mobile 6.x je zaměřena spíše na klasického uživatele mobilního telefonu a také na přívětivost uživatelského prostředí než na profesionální použití. Bohužel Microsoft v tomto odvětví přišel s novým operačním systémem poměrně pozdě. V době zveřejnění tohoto systému má jeho konkurence na trhu už stabilní postavení, proto musí hodně věcí dohánět. Jeho hlavním konkurentem je operační systém Android (od firmy Google) a iOS (od firmy Apple). Kvůli lepšímu průniku na trh Microsoft uzavřel v dubnu 2011 smlouvu o spolupráci na mobilních technologiích s firmou Nokia. Obě firmy si od tohoto spojení slibují nárůst prodeje a zisku. Dále na podzim roku 2011 Microsoft vydal první aktualizaci pro WP7 s názvem Mango a tato platforma se začala pomalu prosazovat a počet jejích uživatelů rostl. Také byl v červenci 2011 otevřen v ČR a dalších zemích Marketplace, který slouží pro prodej a distribuci mobilních aplikací pro tuto platformu. V současné době se připravuje další aktualizace s kódovým označením Tango.

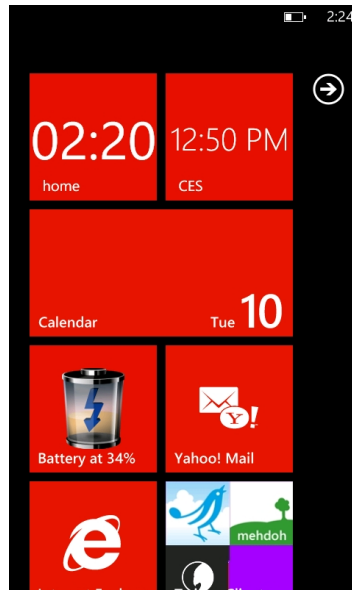
Platformu WP7 jsem si zvolil hlavně kvůli zkušenostem s vývojem aplikací pro Windows Mobile 6.x a také z důvodu zájmu o tuto platformu. V budoucnu bych se chtěl nějakým způsobem věnovat vývoji aplikací pro WP7.

První výrazná změna WP7 oproti ostatním operačním systémům je vidět hned na hlavní obrazovce. Jedná se o *Live Tile* neboli živé dlaždice (panely). Díky těmto dlaždicím je možné získávat různé aktuální informace přímo z hlavní obrazovky, aniž by se musela spouštět jakákoliv aplikace nebo někam kliknout. Také usnadňují přístup k vybraným akcím. Dlaždice odpovídají stylu Metro, takže jsou graficky jednoduché a přehledné. Jako dlaždice si můžete přidat skoro vše.

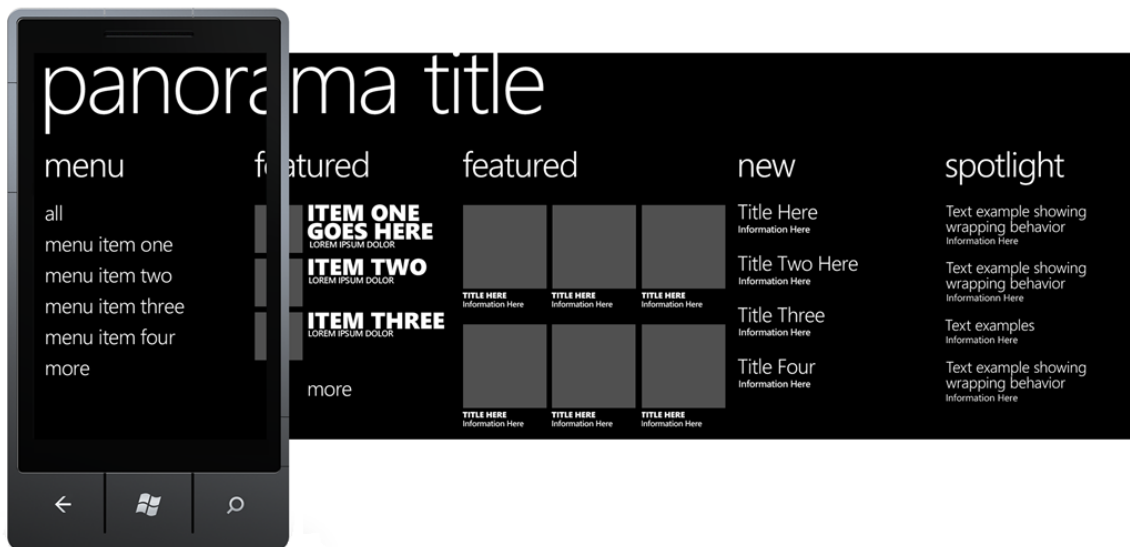
Další zajímavou novinkou WP7 oproti ostatním konkurenčním systémům je například používání systému panoramatických obrazovek, které nabízí unikátní pohled na data a jsou velice přehledné. Tento systém užívá jednotné pozadí pro celé panorama, které přesahuje hranice mobilního telefonu. Na obrazovce mobilního telefonu je vždy vidět pouze určitá část panoramatu a uživatel se posouvá mezi ostatními částmi.

2.1 Hardwarová konfigurace

Microsoft s příchodem WP7 udělal zajímavý krok s tím, že specifikoval minimální hardwarovou konfiguraci, kterou musí každý výrobce mobilního telefonu dodržet, aby jej mohl vydat s touto platformou. Tato specifikace je následující:



Obrázek 2.1: Hlavní obrazovka WP7 s živými dlaždicemi, které zrychlují přístup k vybraným aplikacím a informují o zvolených vlastnostech.



Obrázek 2.2: Ukázka panoramatické obrazovky, která se skládá z 5 různých stránek, mezi kterými uživatel může listovat.

- Kapacitní displej s rozlišením 480 x 800 pixelů podporující 4 a více doteků najednou,
- 7 tlačítek na telefonu (start, zpět, hledat, vypínací tlačítko, hlasitost, fotoaparát),
- minimálně 256 MB RAM a 8 GB paměť pro data,
- procesor alespoň o frekvenci 1 GHz,
- grafický čip s podporou DirectX 9,
- senzory: A-GPS, akcelerometr, kompas, světelný senzor a gyroskop,
- fotoaparát s rozlišením alespoň 5 Mpx a s přisvětlovací diodou,
- podpora Wi-Fi a internetu přes mobilní síť, Bluetooth a FM rádio.

Další vlastnosti jako jsou například HW klávesnice, přední kamera atd. jsou volitelné a závisí na výrobci telefonu. Hlavních výhod minimální HW konfigurace je hned několik. Programátor má velkou jistotu, že pokud mu aplikace poběží na jednom mobilním zařízení, tak s velkou pravděpodobností poběží na všech ostatních telefonech. Pro samotného uživatele je výhoda v tom, že operační systém WP7 bude na každém zařízení běžet velmi rychle.

V poslední době však výrobci mobilních telefonů volají po více různých HW specifikacích, aby mohli ceny mobilních telefonů s WP7 klesnout a zvýšila se tak prodejnost. Microsoft toto volání vyslyšel a aktualizace Tango počítá i s menšími HW požadavky.

2.2 Styl Metro

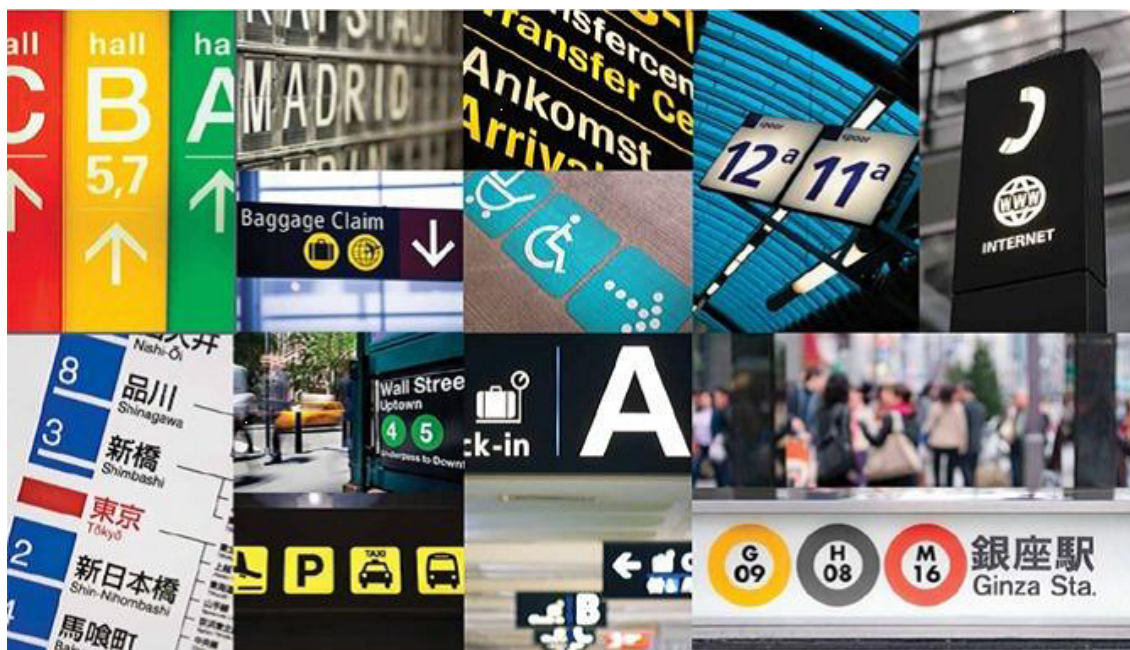
WP7 používá grafický styl nazvaný Metro [15]. Tento styl využívá například také přehrávač Zune. Metro se vyznačuje hlavně svojí jednoduchostí a přehledností. Je to moderní, netradiční styl, který se inspiruje informačními tabulemi na letištích a v metru. Hlavní výhody stylu Metro jsou:

- Čistota, čitelnost a rychlost,
- v popředí je informace, dekorace není důležitá,
- netradiční styl,
- jednoduchost.

2.3 Programové prostředí

Před začátkem práce na samotném vývoji aplikace pro WP7 je potřeba mít nainstalované Microsoft Visual Studio 2010 a také Windows Phone Developer Tools. Pokud chceme vytvářet vlastní grafické prvky, tak můžeme použít například Microsoft Expression Blend. Dalším důležitým nástrojem při vývoji aplikace je Windows Phone Emulator, který simuluje skutečný telefon. Je to velice kvalitní emulátor, podporující emulaci například i akcelerometru a GPS. Bohužel zatím všechny funkce telefonu simulovat neumí. Například k testování fotoaparátu je potřeba mít skutečný telefon.

V současné době (duben 2012) Microsoft vypustil novou verzi emulátoru, která má 256 MB RAM. Tato verze souvisí s příchodem aktualizace Tango a různými HW požadavky



Obrázek 2.3: Ukázka informačních tabulí, vývěsek a stojanů na letištích a v metru z nichž grafický styl Metro vychází.

telefonů. Takže v současné době programátor při testování aplikace volí, zda k testům použije emulátor s 256 MB RAM a nebo s 512 MB RAM. V samotné aplikaci by měl poté zjistit s jakým druhem telefonu pracuje a samotná aplikace by se měla přizpůsobit výkonnosti telefonu.

Pro samotný vývoj aplikací se využívá programového prostředí Silverlight, XNA a .NET Compact Framework 4. Silverlight se využívá především pro implementaci aplikací. Je podobný jako klasickému webovému Silverlightu, mobilní verze se liší jen v pár rozdílech. WP7 používá Silverlight verze 3 s některými funkcemi z verze 4.

XNA je docela nový framework, který se používá hlavně pro vývoj grafických aplikací (například her). Je primárně určen pro začátečníky, kteří se nechtějí zabývat mnoha detaily při vývoji samotné aplikace. XNA je vrstva nad DirectX, která už obsahuje implementované různé funkce, takže je programátor nemusí všechny psát. Hry napsané v XNA lze spustit i na konzoli XBOX 360. Je také možné oba přístupy (Silverlight a XNA) zkombinovat, kde část aplikace bude napsána v Silverlightu a část v XNA.

Pro tuto práci vyvíjím aplikaci s pomocí Silverlightu, takže další text tohoto dokumentu bude o vývoji aplikace s použitím Silverlightu. U XNA se použité postupy a metody mohou více či méně lišit.

Při implementaci samotné aplikace je rozdělen kód pro grafické uživatelské rozhraní (GUI) a pro klasický kód, který určuje funkčnost aplikace. Pro popis GUI se používá jazyk XAML, který vychází z jazyka XML a slouží pro vytváření tlačítek, tabulek, textů, atd. Na tento kód navazuje klasický kód napsaný v jazyce C# nebo Visual Basic. Hlavní výhodou tohoto přístupu je oddělení grafické stránky aplikace a její funkčnosti. Toto zabezpečuje jak přehlednost aplikace, tak i možnost současné práce více programátorů, kdy jeden programátor se může věnovat vzhledu aplikace a druhý psát vlastní funkčnost.

2.4 Životní cyklus aplikace

WP7 nepodporuje u aplikací třetích stran *multitasking*. Je to jednak z důvodu úspory baterie a také kvůli zajištění větší rychlosti aktuálně běžící aplikace. Ačkoliv verze Mango přináší podporu tzv. *Background Agents* [11]. Tito agenti mohou za určitých podmínek běžet v pravidelných intervalech na pozadí a vykonávat určitou činnost, ale jsou velice omezeni operačním systémem. O těchto agentech bude řeč později. S pojmem multitasking se pro platformu WP7 zavádí také pojem *tombstoning* (hibernace, uspání). Když je aplikace hibernována, tak byla vlastně ukončena, ale informace o jejím stavu zůstávají zachované. Programátor se tedy musí postarat o správné ukončení (uložení aktuálního stavu) a opětovné spuštění (obnovení stavu) aplikace při nějaké nečekané události (telefonát, příchod sms, atd.). Pouze originální aplikace, tedy aplikace přímo od Microsoftu nebo jeho partnerů (Nokia, HTC, atd.), jako je například přehrávač hudby mohou běžet současně s jinou aplikací.

Programátor může využít následující události pro zachycení spuštění, aktivaci, deaktivaci a vypnutí aplikace:

- *Deactivated* - volá se při deaktivování aplikace,
- *Activated* - volá se při reaktivování aplikace,
- *Closing* - volá se při vypnutí aplikace,
- *Launching* - volá se při spuštění aplikace.

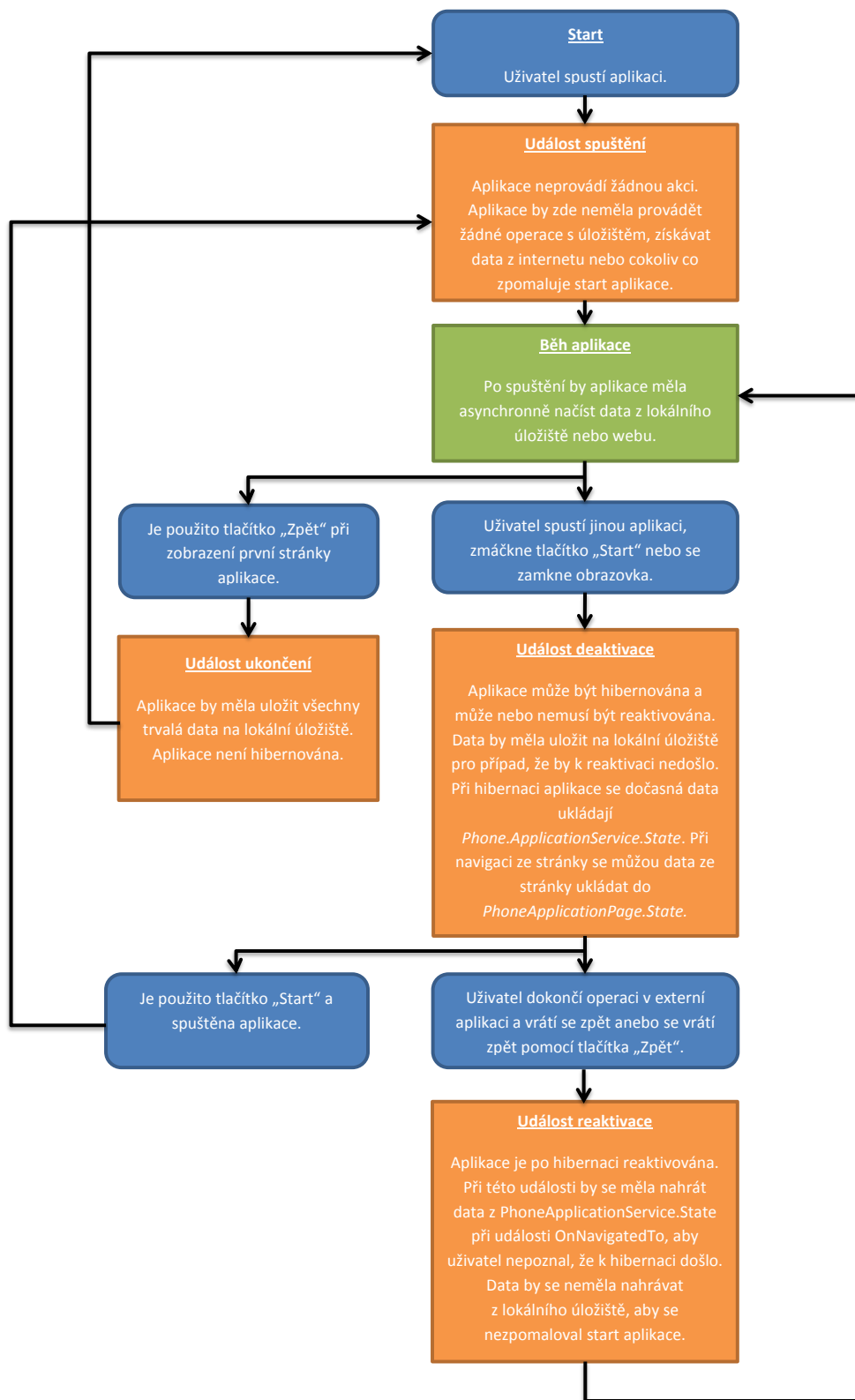
Data, která chceme mít k dispozici po obnovení aplikace, ukládáme pod zvolenými identifikátory v objektu typu *IDictionary* ve vlastnosti *PhoneApplicationService.Current.State*. Ukládaná data musí být serializovatelná, jelikož se ukládají do souboru.

Pokud vývojář potřebuje vykonávat nějakou akci na pozadí (synchronizování dat, upozornění uživatele, atd.), tak v aktualizaci Mango může využít již dříve zmíněných agentů. I když se nejedná o pravý multitasking, jelikož tito agenti mají různá omezení (jak často mohou běžet, jak dlouho, atd.).

Každá aplikace může používat pouze jednoho agenta, který bude zajišťovat periodické akce. Také je omezený celkový počet najednou spuštěných agentů. Existují dva typy těchto agentů:

- *Periodic Tasks* - tento typ agenta může běžet pouze krátký časový úsek (typicky 25 sekund) v pravidelných intervalech (typicky 30 minut). Používá se většinou pro malé úlohy jako je synchronizace dat, aktualizování živých dlaždic atd.
- *Resource-Intensive Tasks* - tento agent běží poměrně dlouho na pozadí (typicky 10 minut), ale pouze za určitých podmínek:
 - Baterie musí být nabitá alespoň z 90 %,
 - mobilní telefon musí být připojen k Wi-Fi,
 - mobilní telefon musí být zamčený,
 - uživatel nesmí právě telefonovat.

Využívá se většinou pro přenos dat, synchronizaci a výpočetně náročné operace.



Obrázek 2.4: Životní cyklus aplikace. Upraveno podle [10].



Obrázek 2.5: Rozdělení úloh běžících na pozadí. Oranžově jsou vyznačeny programátorem vytvořené úlohy. Zeleně jsou označeny předchystané a specializované úlohy a modře obecné typy úloh. Upraveno podle [3].

Pokud agenti splní všechny podmínky pro jejich běh, tak stále není zaručeno, že budou spuštěni. Operační systém je může například kvůli snížení spotřeby baterie nebo úspoře paměti kdykoliv deaktivovat.

Dále aplikace pro WP7 z důvodu větší bezpečnosti běží ve vyhrazeném prostoru (tzv. sandbox architektura), což je další rozdíl oproti Windows Mobile 6.x. Aplikace se tedy nemůže dostat k samotné fyzické struktuře souborového systému, ale je jí přiřazen izolovaný a omezený vlastní prostor, do kterého si ukládá data a má k němu přístup pouze tato aplikace. Díky použití této architektury se zvyšuje bezpečnost systému, ale také je díky ní omezen počet funkcí, které lze programově ovládat.

S vyhrazeným prostorem souvisí také pojem *IsolatedStorage*, který umožňuje aplikaci zapisovat a číst data z vyhrazeného prostoru.

2.5 Marketplace

Marketplace slouží uživatelům WP7 pro přístup k novým aplikacím a hrám. Aplikace jsou tříděny do několika kategorií, jsou zde i různé žebříčky a informace o novinkách. Pro každou podporovanou zemi je Marketplace mírně odlišný (jiný jazyk, odlišná nabídka aplikací, atd.). Marketplace je vlastně internetový obchod. Vývojáři vůbec nemusí při vývoji a testování aplikaci nahrávat na Marketplace, ale mohou na testování své aplikace použít program Application Deployment, který slouží k nahrání aplikace do mobilu.

Marketplace dále podporuje beta testování aplikace a trial mód před koupí aplikace. Beta testování funguje obdobně jako klasické zveřejnění aplikace. Rozdíl je hlavně v tom, že aplikace neprochází celým schvalovacím procesem a také že není veřejně dostupná. Aplikaci

```

IsolatedStorageFile myIsolatedStorage =
    IsolatedStorageFile.GetUserStoreForApplication();

using (StreamWriter writeFile = new StreamWriter(new
    IsolatedStorageFileStream("myFile.txt", FileMode.Create,
    FileAccess.Write, myIsolatedStorage))
{
    string someTextData = "Tento text bude ulozen v novem
        souboru v IsolatedStorage!";
    writeFile.WriteLine(someTextData);
    writeFile.Close();
}

```

Obrázek 2.6: Ukázka zápisu dat do vyhrazeného prostoru aplikace.

si může stáhnout a vyzkoušet pouze omezený počet uživatelů (maximálně 100) pomocí pozvánek, které uděluje vývojář. Samotná aplikace je dostupná pro beta testování pouze určitou dobu, poté je z Marketplace smazána.

Trial mód umožňuje uživateli vyzkoušet si aplikaci předtím, než ji zakoupí. Programátor musí pouze přidat do kódu tento řádek, pomocí kterého určuje, jaké funkce budou v trial verzi aplikace dostupné:

```

if (CurrentLicense.IsTrial().Equals(true))
{
    // nejaky kod ...
}

```

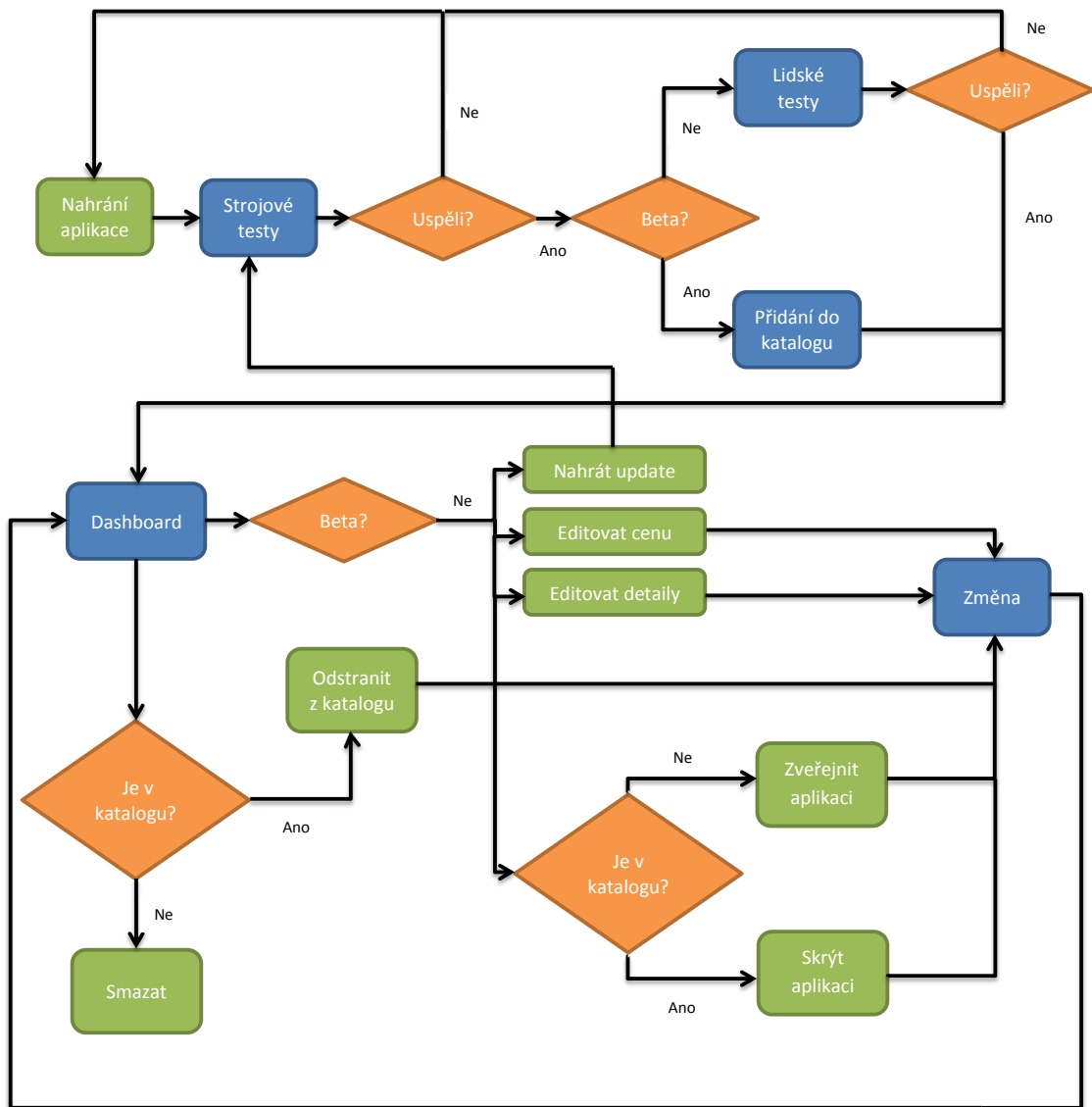
Obrázek 2.7: Ukázka podmínky, která zpřístupňuje určitou část programu v trial verzi aplikace.

Po zkompileování projektu vzniká soubor s příponou XAP. Jedná se v podstatě o obyčejný ZIP, ve kterém se nacházejí ikony, dll knihovny a také dva soubory *AppManifest.xaml*, který určuje, jaké knihovny se mají použít a *WMAAppManifest.xml*, ve kterém jsou zapsány aplikací využívané prostředky (mikrofon, kamera, atd.). Maximální velikost XAP souboru je omezena na 400 MB a přes mobilní síť není možné stáhnout do telefonu aplikaci větší než 20 MB.

Pokud má programátor k dispozici XAP soubor, tak musí aplikace před zveřejněním na Marketplace prvně projít schvalovacím procesem, který testuje, zda aplikace odpovídá pravidlům, které musí všechny aplikace dodržovat. Schéma schvalovacího procesu je popsáno na následujícím obrázku.

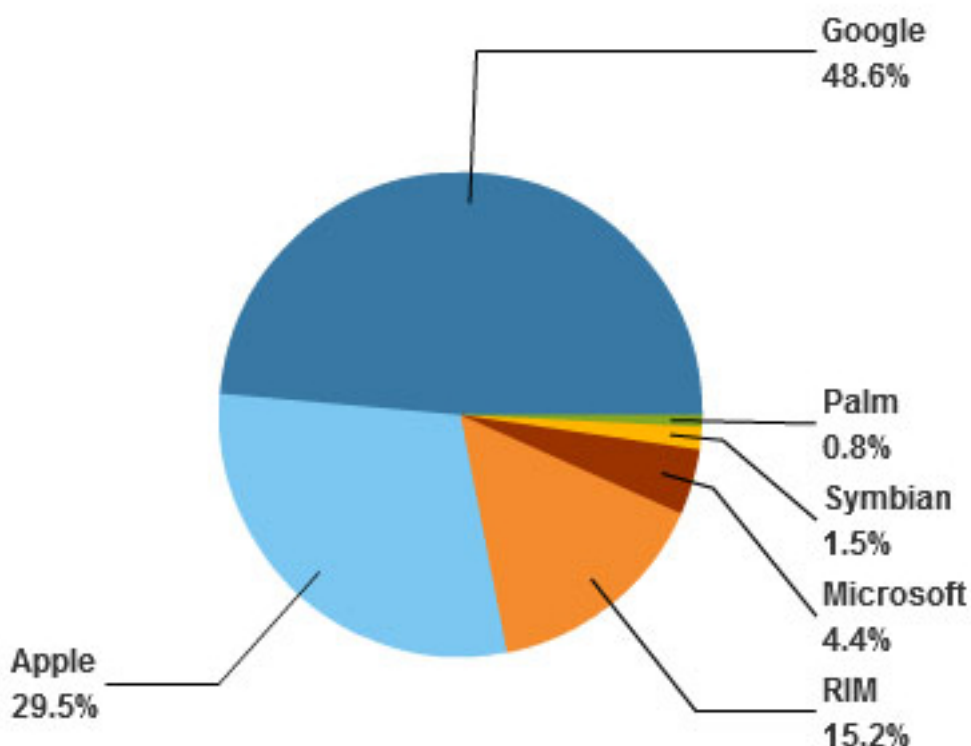
2.6 Budoucnost WP7

V současné době Microsoft postupně zveřejňuje menší aktualizaci Tango, která by měla přinést ještě větší podíl WP7 na trhu s mobilními telefony. V roce 2011 činil jeho podíl asi 0,5%, aktuálně je to asi kolem 4%. Pořád to není příliš vysoké číslo, ale samotný Microsoft věnuje velké finanční prostředky na propagaci WP7. Nedávno také firma Nokia vypustila



Obrázek 2.8: Schéma certifikačního procesu aplikace. Upraveno podle [1].

nový mobilní telefon Nokia Lumia 800 s WP7, který má zatím velice dobrou prodejnost a v recenzích je tento telefon hodnocen velmi kladně. Během letošního roku (2012) by také měla být uvolněna aktualizace Apollo (označovaná také jako Windows Phone 8), která by měla přinést například podporu pro NFC, paměťových karet microSD, dvoujádrové procesory a podporu displejů s HD rozlišením. Aktualizace Apollo bude využívat jiné jádro než aktuální verze WP7, které používají jádro z WinCE, ale bude používat jádro z Windows 8. Také se pořádně zatím neví, jak to bude se zpětnou kompatibilitou WP7 a aktualizací Apollo. Snad se Microsoftu podaří v budoucnu získat větší podíl na trhu, protože by byla škoda, kdyby takto zajímavý a rychle se rozvíjející operační systém zmizel.



Obrázek 2.9: Podíl firem na americkém trhu s chytrými telefony v lednu 2012. Celkový počet chytrých telefonů je 101,3 milionů.

Kapitola 3

Analýza existujících řešení

Při rozhodování o výběru druhu aplikace, kterou budu vyvíjet, jsem měl k dispozici mobilní telefon s WP7. S jeho pomocí jsem procházel a testoval různé druhy aplikací umístěných na Marketplace. Také jsem při používání telefonu využíval zabudované systémové aplikace. Například i budík, ale jelikož nenabízí mnoho možností nastavení a je velice jednoduchý, tak jsem zkusil na Marketplace najít nějakou jeho náhradu. Často mě právě tento budík nedonutil vstát z postele, pouze jsem ho vypnul a spal dál. Bohužel žádná z nabízených aplikací mě nezaujala a nesplňovala moje požadavky, tak jsem začal uvažovat o vytvoření vlastního pokročilého budíku pro WP7. Samotná výsledná aplikace by měla na základě mé prvotní představy splňovat hned několik základních požadavků:

- Měla by odpovídat stylu Metro,
- být uživatelsky přívětivá,
- přehledná, jednoduchá na ovládání,
- nabízet několik pokročilých funkcí, které uživatele donutí vstát z postele.

V této kapitole bude popsána analýza konkurenčních řešení, jejich vzhledu a funkčnosti, kterou jsem se rozhodl použít nebo přenést na platformu WP7.

Moje první kroky při zkoumání konkurenčních aplikací směřovali na Marketplace. Jelikož je WP7 mladou platformou a zatím ne příliš rozšířenou, tak mnoho kvalitních aplikací na téma buzení není. Většina budíků je placená (cena často 99 centů) a také vůbec neodpovídají stylu Metro. Jako první nabízená aplikace se mi zobrazil budík Free Talking Alarm Clock. Tato aplikace nenabízí příliš mnoho pokročilých funkcí a také její design se mi příliš nelíbí. Ze zajímavých funkcí, kterými tyto hodiny disponují, je možné zmínit používání animací při zobrazených hodinách, i když poté se hodiny často stávají nepřehlednými a dle mého názoru uživatele spíše ruší. Další funkce, která se příliš často nevyskytuje je buzení pomocí FM rádia. Uživatel si vybere místo nějaké hudby svoji oblíbenou rádiovou stanicí a ta se zapne v určený čas. Také je možné si nechat zobrazit informace o počasí, ale celkově mě tato aplikace příliš nezaujala a žádnou extra funkci neoslovila.

Další nabízená aplikace se jmenuje Night Stand Clock¹, která je zaměřena hlavně na samotný vzhled budíku. Tato aplikace patří podle mě mezi 5 nejlepších aplikací pro WP7 zaměřených na buzení. Nenabízí mnoho funkcí, ale její grafická podoba je opravdu povedená.

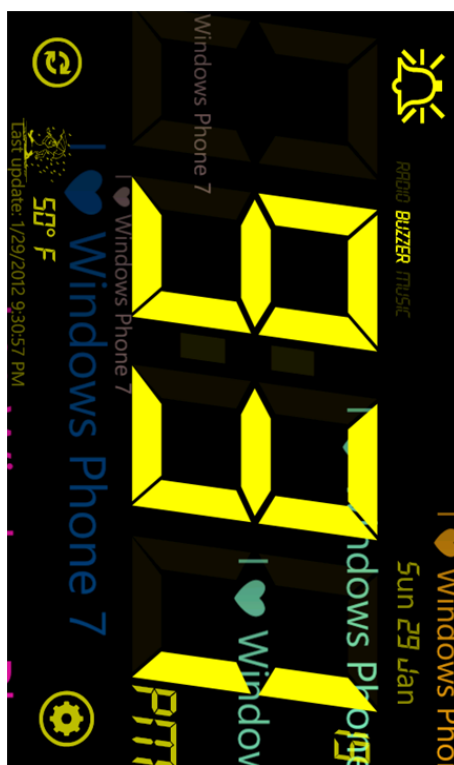
¹<http://www.windowsphone.com/en-us/apps/1ece760c-452a-e011-854c-00237de2db9e>

Na výběr je několik různých druhů hodin (digitální, analogové, atd.), které se budou zobrazovat. U této aplikace mě zaujali hlavně její digitální hodiny, které jsou hezké a přehledné. Tyto digitální hodiny mě inspirovali při tvoření vlastních digitálních hodin do aplikace. Obě výše zmíněné aplikace si je možné zdarma stáhnout anebo zakoupit. Rozdíl ve funkčnosti placené a neplacené verze jsem ani u jednoho programu nezaznamenal.

Dalších několik aplikací bylo docela jednoduchých a nic mě na nich příliš nezaujalo. Většina těchto aplikací není příliš graficky povedená a nabízí jenom základní funkce, které jsou mnohy stejné pro všechny tyto aplikace a odlišné jsou pouze díky jejich různé grafické podobě. Dále hodně aplikací nabízí výběr vlastních písniček z hudební knihovny, tuto funkci jsem se rozhodl také implementovat do vlastní výsledné aplikace.

Zajímavou aplikací, na kterou jsem narazil, je G-Alarm² a to byl dle mého názoru nejlepší budík pro WP7. Sice G-Alarm neodpovídá Metro stylu a jedná se o konverzi z jiné platformy, přesto je vcelku přehledný a jednoduchý na ovládání. Hlavní jeho přednost je možnost monitorování fází spánku a buzení uživatele v nejlepší možný čas. Dále aplikace nabízí několik dalších druhů buzení. Například nutnost vyřešení zadané rovnice uživatelem, aby se budík vypnul nebo vyřešení bludiště. Funkce vyřešení rovnice se mi velice líbila a rozhodl jsem se jí použít také ve svém budíku. Aplikace má také bohaté možnosti nastavení.

Další něčím zajímavé aplikace jsem už neobjevil. Hodně aplikací bylo podobných jako originální systémový budík, akorát měly navíc pár jednoduchých funkcí.



Obrázek 3.1: Hlavní obrazovka aplikace Free Talking Alarm Clock se zobrazeným aktuálním časem a zapnutými animacemi.

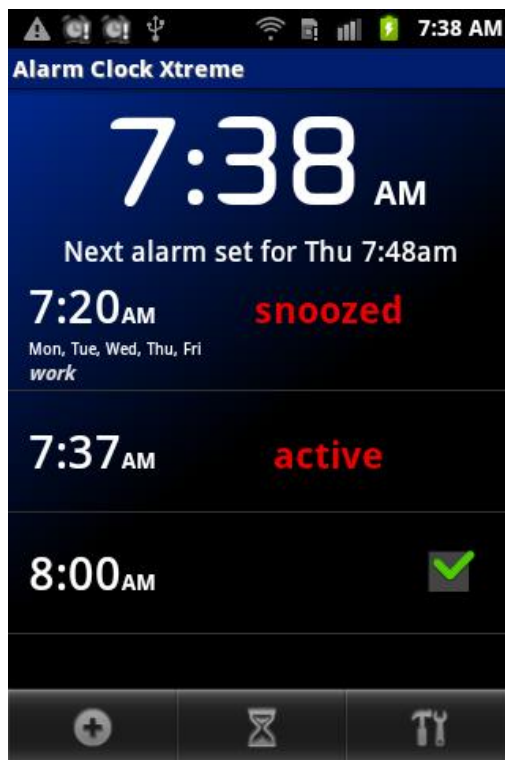
Jako další platformu jsem zvolil operační systém Android a jeho internetový obchod Google play. Tento operační systém má v současné době největší podíl na trhu s mobilními

²<http://www.windowsphone.com/en-us/apps/d8751f2f-a22c-e011-854c-00237de2db9e>

telefony a nabízí opravdu hodně aplikací, ale také mnoho z nich není příliš kvalitních. Jelikož systém Android pracuje na mnoha různých mobilních telefonech, tak se může stát, že na některém telefonu nepůjde aplikace spustit nebo nebude pracovat korektně.

První aplikace, která mi byla nabídnuta, měla názvem Alarm Clock Xtreme³. Samotná aplikace je samotnými uživateli velice kladně hodnocena, i když nenabízí nějaké zásadní funkce. Je tu opět možnost vypnutí budíku pomocí vyřešení rovnice a nebo zatřepáním s telefonem. Použití akcelerometru zabudovaného v telefonu pro vypnutí budíku se mi líbilo a také jsem se rozhodl tuto funkčnost použít. Aplikace má také docela pěkné uživatelské rozhraní.

Hodně nejstahovanějších budíků pro Android se mi zdálo docela podobných. Měli obdobné funkce, ať už se jedná o řešení rovnice, odložení budíku anebo grafické rozhraní. Velká část dalších aplikací byla graficky velice nepovedená, nenabízela žádné užitečné funkce a jejich ovládání bylo značně komplikované. Akorát aplikace Gentle Alarm⁴ nabízí nový druh buzení, kdy si uživatel musí zapamatovat zobrazenou informaci a poté ji zopakovat, ale graficky je tato aplikace opět nedodělaná.



Obrázek 3.2: Ukázka grafického prostředí aplikace Alarm Clock Xtreme.

Nakonec jsem zvolil operační systém iOS od firmy Apple. Bohužel jsem neměl k dispozici telefon s iOS, takže jsem aplikace mohl procházet pouze na PC a nemohl si je vyzkoušet. Pokud si uživatel chce na svém počítači najít nějakou aplikaci pro svůj mobilní telefon s iOS, tak si musí do počítače nainstalovat aplikaci iTunes, která mu zpřístupní internetový obchod s aplikacemi, hrami a písníčkami. Není možné aplikace vyhledávat prostřednictvím

³https://play.google.com/store/apps/details?id=com.alarmclock.xtreme.free&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5hbGFybWVsb2NrLnh0cmVtZS5mcmVlIl0.

⁴https://play.google.com/store/apps/details?id=com.mobitobi.android.gentlealarm&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5tb2JpdG9iaS5hbmRyb2lkLmdlbnRsZWFsYXJtIl0.

webového prohlížeče na internetu, jako to bylo u předchozích dvou platforem. Většina aplikací co jsem měl možnost vidět, byla graficky povedená, ale opět nenabízeli žádné pokročilé funkce. Narazil jsem na pár zajímavých aplikací, ale jejich funkce jsem už viděl na předchozích platformách. Pouze jedna aplikace měla možnost vypnutí budíku pomocí hlasu. Tuto funkci jsem ještě u žádné jiné předchozí aplikace neviděl. Uživatel si před vytvořením budíku nastavil určitou hranici hlasitosti zvuku a poté musel tuto hranici překonat, aby se budík vypnul.



Obrázek 3.3: Příklad elegantního grafického prostředí aplikace Alarm Tunes 2 pro iOS.

Po prozkoumání různých aplikací na odlišných platformách jsem objevil několik aplikací, které byly stejné na všech platformách. Z některých aplikací jsem si vzal poučení a pár zajímavých nápadů, které bych chtěl mít ve své aplikaci. Jedná se hlavně o možnost vypnutí budíku vyřešením rovnice a nebo zatřepáním s telefonem. Dále by měla výsledná aplikace disponovat výběrem vlastní hudby a také možností odložit zvonící budík. Pěkné a přívětivé grafické rozhraní je samozřejmost, proto aplikace musí využívat originální systémové komponenty.

Kapitola 4

Návrh aplikace

Po prozkoumání konkurenčních aplikací, zjištění jejich výhod a slabin jsem přemýšlel o tom, co by uživatel od budíku vyžadoval a potřeboval. Sám mám problém vstát, když mi budík začne zvonit, proto jsem se hlavně zaměřil na účinné metody jak uživatele donutit vstát z postele. Většina aplikací začne pouze zvonit a uživatel může jednoduše budík vypnout nebo jej odložit. Tomuto stavu jsem chtěl nějak zamezit, a proto jsem se rozhodl implementovat několik různých druhů buzení, u kterých bude uživatel muset buď přemýšlet anebo vykonat nějakou činnost.

Z existujících řešení, které jsem měl vyzkoušené z jiných aplikací, jsem se rozhodl implementovat vygenerování rovnice, kterou bude muset uživatel vyřešit. Při analýze existujících řešení jsem zjistil, že není možné nastavit obtížnost vygenerovaných rovnic, což mi vadilo, protože často tyto rovnice byly velice jednoduché. Navrhl jsem tedy několik různě obtížných druhů rovnic a tyto rovnice se snažil vyřešit. Zjistil jsem, kolik času potřebuji k jejich vyřešení a jak moc náročné na soustředění jsou. Z tohoto výzkumu jsem vybral určité druhy rovnic a rozdělil je do tří kategorií podle obtížnosti. Uživatel si bude moct zvolit obtížnost rovnic z těchto tří kategorií. Postupně by bylo možné přidat další obtížnosti. Samotné generování rovnic není příliš náročné na implementaci, pokud se tedy nepoužívají nějaké proměnné atd., ale přesto je to velice účinný druh buzení. Jako další dobrý nápad je využití akcelerometru, který se také objevilo v několika jiných aplikacích. Uživatel by si opět zvolil míru s jakou musí třepat s telefonem pro vypnutí budíku.

Dále jsem přemýšlel jaké nové vlastnosti vložit do mé budoucí aplikace. Postupně jsem zkoumal všechny možnosti telefonů s WP7. Jako první mě napadlo využít vestavěný GPS modul, kdy by si sám uživatel nastavil počet metrů, které musí ujít, aby se budík vypnul a aplikace by pomocí GPS modulu hlídala počet ušlých metrů. Bohužel jsem zjistil, že uživatelovu lokaci je možné zaměřit jenom venku, takže v bytě nebo domě by toto nebylo dobré řešení. Aplikaci by bylo možné upravit tak, že by budík zvonil, dokud by aplikace nezískala nějaká data z GPS satelitů. To znamená, že by uživatel musel mobilní telefon vystrčit třeba z okna nebo jít ven. Toto řešení se mi, ale nezdálo jako ideální, takže jsem použití GPS modulu zavrhl.

Jako další věc, která by mohla nějakým způsobem sloužit pro buzení je použití vestavěného fotoaparátu. Fotoaparát musí mít každý telefon s WP7, takže by toto řešení mohli používat všichni uživatelé. Většina těchto fotoaparátů by měla být kvalitní a musí mít automatické zaostřování, takže by nebyl problém pořídit nějaké detailní snímky. Prvotní věc, jak bych využil vestavěný fotoaparát, by bylo zachycení snímku a následné rozpoznání nějakého zadaného tvaru. Jenže jsem zjistil, že tato operace by byla příliš složitá a také nevím, jak moc úspěšné by bylo samotné rozpoznávání objektů. Proto jsem se rozhodl zkusit využít

vestavěného fotoaparátu jinak. Jelikož v práci pracuji se čtečkami čárových kódů, tak mě napadlo těchto zkušeností nějak využít ve své aplikaci.

Před dalším zkoumáním této možnosti jsem na Marketplace našel nějaké aplikace, které umožňovali čtení čárových kódů a poté zboží vyhledali na internetu. Aplikace jsem vyzkoušel v praxi a samotná úspěšnost čtení mě překvapila. Podařilo se mi načíst skoro všechny čárové kódy různých formátů. Dále jsem tedy zjišťoval, jestli existují nějaké knihovny pro čtení čárových kódů a které knihovny tyto aplikace používají. Jako první jsem našel knihovnu Windows Phone 7 Silverlight ZXing Barcode Scanning Library [12], která je nejpoužívanější a podle ohlasů by měla být také nejlepší. Mnoho dalších knihoven na skenování čárových kódů pro WP7 jsem už nenašel. Je možné vyzkoušet ještě ZXing 2.0 [9], ale tento port ještě není úplně doladěný. Po důkladném procházení různých fór a zkoumání vlastností knihovny WP7 Silverlight ZXing Barcode Scanning Library jsem se rozhodl ji využít ve svém projektu. Vlastní použití fotoaparátu a vybrané knihovny jsem vymyslel tak, že by uživatel při nastavování nového budíku načel nějaký čárový kód, který je vzdálen od postele a pro deaktivaci budíku jej bude muset opět načíst. Toto řešení by jej mělo donutit vstát z postele, pokud si ovšem čárový kód neumístí k posteli.

Při navrhování dalších vlastností aplikace jsem si vzpomněl, že v několika různých aplikacích se zobrazila nějaká informace uživateli, který si ji musel zapamatovat a zopakovat. Toto řešení je zajímavé, ale rozhodl jsem se jej nějak vylepšit a zjednodušit. Zkoumal jsem různé způsoby, jak by uživatel musel nějaký zadaný úkol jednoduše vyřešit nebo něco zopakovat. Zajímavou možností by bylo například vygenerování nějakého bludiště, které by musel uživatel projít. Jenže toto řešení by v Silverlightu naprogramovat nebylo vůbec jednoduché. Chtěl jsem nějak využít dostupné komponenty v Silverlightu a vytvořit jednoduchý, ale náročný druh buzení. Poté jsem si vzpomněl na hru, kterou jsme hráli jako děti, kde byl čtverec 3 x 3 s jedním prázdným políčkem. Jednotlivé dílky obsahovali čísla od 1 do 8 a museli se poskládat postupně od 1 do 8. Přemýšlel jsem jak princip této hry využít pro buzení. Napadlo mě, vytvořit matici z tlačítek 3 x 3, které by obsahovali různá vygenerovaná čísla a uživatel by musel v nějakém pořadí (vzestupně nebo sestupně) tato tlačítka zmáčknout. Pokud by nějaké tlačítko zmáčkne ve špatném pořadí, tak by začal zase znova.

Když jsem uvažoval nad vlastnostmi, které by měla aplikace splňovat, tak jsem dělal i různé další činnosti a také si občas udělal pauzu na několik minut. Jenže problém byl v tom, že jsem ztratil pojem o čase a místo pauzy na 10 minut jsem najednou ležel třeba půl hodiny. Tento problém jsem měl také při vykonávání nějaké činnosti, na kterou jsem si vyčlenil například 20 minut, ale dělal jsem ji dvakrát delší čas. Jelikož jsem ztratil pojem o čase, napadla mě další zajímavá funkce aplikace, kdy by uživatel nezadával nějaký konkrétní čas buzení, ale jenom časový úsek (například 10 minut) a budík by začal zvonit za 10 minut. Toto by bylo ideální řešení pro tyto problémy.

Při zkoumání různých možností jsem také narazil na problém, kdy samotný Silverlight pro WP7 neobsahuje všechny komponenty, které jsou použity ve vlastním operačním systému a které jsem potřeboval ve své aplikaci. Proto jsem musel najít nějaké další knihovny, které by mi zpřístupnily tyto komponenty. Při hledání různých specializovaných komponent na nějakou činnost (např. na výběr času nebo výběr ze zadaných možností) jsem narazil na několik knihoven, které poskytují tyto komponenty. Z těchto knihoven jsem se rozhodl na základě dobré dokumentace a mnoha příkladů použít knihovny Windows Phone 7 Toolkit Controls [13] a Coding4Fun Windows Phone Toolkit [14]. Více se o těchto knihovnách dočtete v následujících podkapitolách.

Použitím těchto knihoven se jednak ušetří čas při samotném vývoji aplikace, takže se může věnovat například testování a dále jsou tyto knihovny již prověřené a odladěné.

V neposlední řadě by konečná verze aplikace měla umožňovat určitou customizaci prostředí a také si pamatovat poslední zvolené nastavení, aby uživatel nemusel vše opět nastavovat. Dále by měla být lokalizována alespoň do dvou jazyků. Jako hlavní jazyk jsem zvolil angličtinu, protože největší skupina uživatelů WP7 ovládá tento jazyk.

4.1 ZXing Barcode Scanning Library

Jedná se o open-source knihovnu, která je původně vytvořena společností Google a napsaná v jazyce Java. Jedná se tedy o port. Tato knihovna umí číst 1D a 2D čárové kódy. Knihovna rozpoznává tyto formáty čárových kódů:

- UPC-A a UPC-E,
- EAN-8 a EAN-13,
- Code 39,
- Code 128,
- QR Code,
- ITF,
- Data Matrix (netestováno),
- PDF417 (netestováno).

Práce se samotnou knihovnou je velice jednoduchá díky různým příkladům.

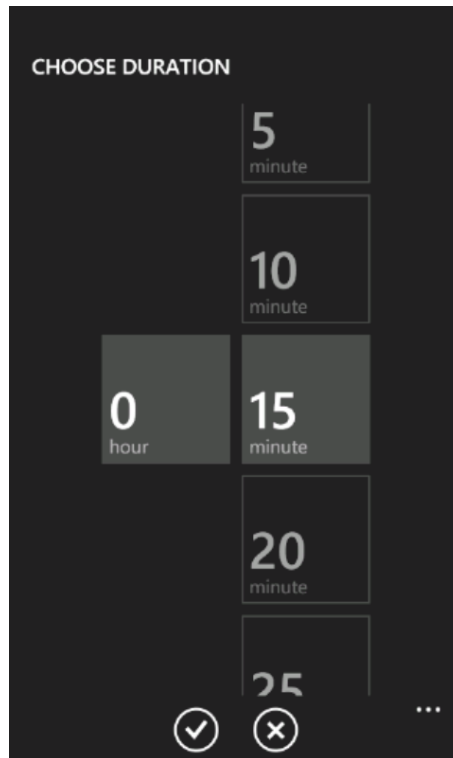
4.2 Coding4Fun Windows Phone Toolkit

Tato knihovna nabízí různé komponenty pro WP7. Většina těchto komponent jsou ovládací prvky, které usnadňují práci s WP7. Pro můj projekt jsem se rozhodl použít například komponentu TimeSpanPicker, která slouží pro vybrání určitého časového období. Dále tato komponenta umožňuje různá nastavení svých vlastností (například výběr vteřin atd.). Mimo TimeSpanPicker používám také komponentu ColorPicker, která slouží pro výběr barvy digitálních hodin.

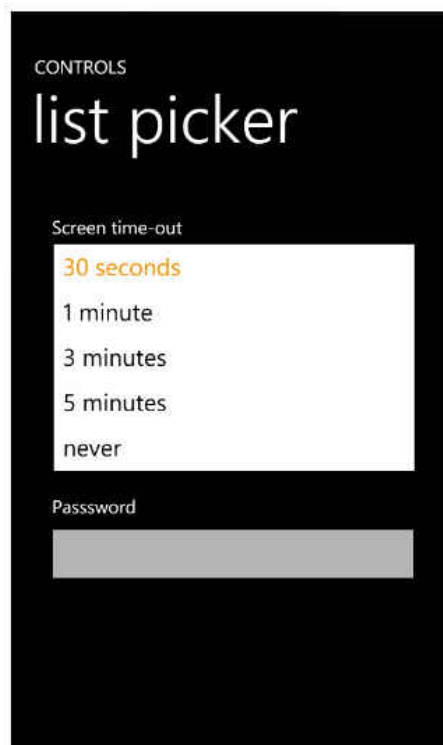
V dalších verzích aplikace chci používat také komponentu MessagePrompt, která je rozšířenou implementací klasického MessageBoxu. MessagePrompt dovoluje mnohem lepší nastavení různých vlastností než MessageBox.

4.3 Windows Phone 7 Toolkit Controls

Knihovna Windows Phone 7 Toolkit Controls nabízí mnoho základních komponent, které používá WP7 a práce s mobilním telefonem je díky nim velice jednoduchá. Z nabízených komponent jsem hodlal v projektu využívat například TimePicker pro výběr času buzení. Dále ToggleSwitch, což je vlastně vylepšený klasický CheckBox. A pro výběr nějaké vlastnosti z různých nabízených možností využít komponentu ListPicker, která má také bohaté možnosti nastavení. Například je možné využít celoobrazový režim zobrazení nebo pouze rolovací nabídku.



Obrázek 4.1: Ukázka komponenty TimeSpanPicker, která slouží pro výběr doby trvání.



Obrázek 4.2: Komponenta ListPicker s aktivovanou rolovací nabídkou a výběrem z několika nabízených možností.

Kapitola 5

Implementace aplikace Clever Alarm

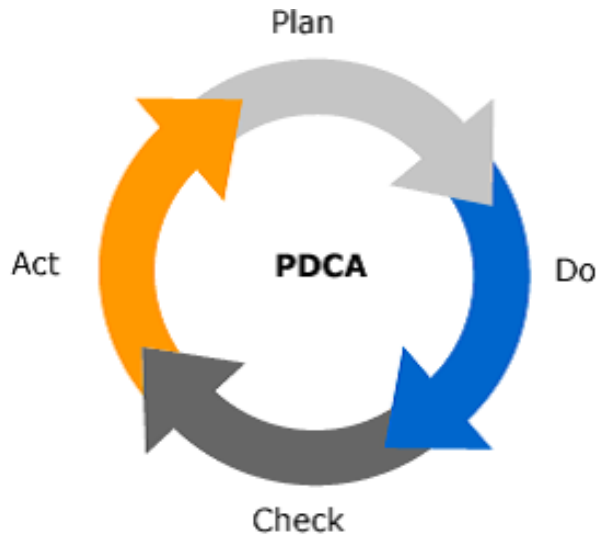
Před začátkem samotného vývoje jsem měl napsaných několik bodů, které by měla výsledná aplikace mimo jiné splňovat a v čem by se měla odlišit (viz. předchozí kapitola). Samozřejmě tyto body jsem postupně doplňoval dalšími požadavky, které mě napadly, případně vzešly z testování nebo z návrhů samotných uživatelů aplikace. Prvotní specifikace aplikace vypadala následovně:

- Jednoduchá aplikace na ovládání,
- využívání stylu Metro,
- několik různých druhů buzení (řešení rovnice, třepání s telefonem, čtení QR kódu, řazení, atd.),
- co nejméně různých nastavení aplikace,
- možnost používat aplikaci místo hodin,
- výběr vlastní hudby.

Dále jsem při vývoji udržoval seznam vlastností, které je potřeba doimplementovat. Tyto body jsem měl seřazené podle priority. Postupně jsem jednotlivé úkoly zpracovával, případně jsem přidával nové a měnil priority různých bodů.

Při implementaci vlastní aplikace jsem se snažil držet metodiky PDCA (Plan, Do, Check, Act). Metodika PDCA se často používá při procesu zlepšování. Cyklus PDCA se skládá ze 4 částí:

- *Plan* - shromáždění dat o problému, navrhnout možná řešení a zvolení nejvhodnějšího řešení,
- *Do* - realizace řešení,
- *Check* - zhodnocení výsledků testování, zjištění případných problémů,
- *Act* - vyhodnocení celého procesu, vytvoření konečného řešení.



Obrázek 5.1: Metodika PDCA

Tento postup jsem využíval při implementaci různých vlastností aplikace a jejich následných vylepšení.

V první fázi vývoje aplikace jsem vytvořil kostru programu. Tato kostra se skládala z hlavního menu, stránky se samotným nastavením budíku (čas buzení, délka odložení, atd.) a seznamu všech uživatelem vytvořených budíků. K vytvoření samotného budíku jsem používal API `Microsoft.Phone.Scheduler` a jeho třídu `Alarm`.

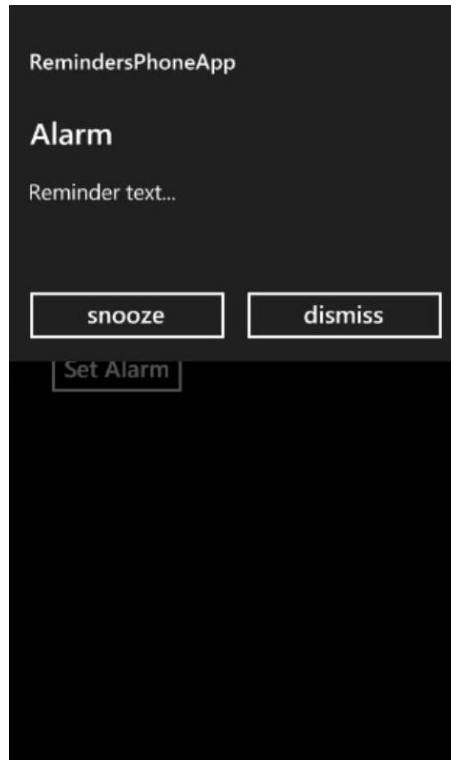
```
//vytvoreni budiku
Alarm alarm = new Alarm("Test Alarm");
alarm.Title = "My Alarm Title";
alarm.Content = "My Test alarm";
alarm.BeginTime = DateTime.Now.AddSeconds(10);
alarm.ExpirationTime = DateTime.Now.AddSeconds(15);
alarm.RecurrenceType = RecurrenceInterval.None;
alarm.Sound = new Uri("");
//pridani budiku do ScheduledActionService
ScheduledActionService.Add(myReminder);
```

Obrázek 5.2: Ukázka vytvoření budíku s pomocí předdefinované třídy `Alarm`.

Pokud se takto budík v aplikaci vytvoří, tak je možné aplikaci vypnout a s mobilním telefonem dále normálně pracovat. Samotná aplikace tedy nemusí být spuštěna a operační systém hlídá spuštění budíku. Bohužel jsem narazil na mnoho omezení tohoto postupu, díky kterým není možné zvolené řešení použít ve výsledné aplikaci.

Jako zásadní omezení je nemožnost upravit dialog, který se uživateli zobrazí v nastavený čas. Vždy se zobrazí pouze hláška se zadaným textem a dvěma tlačítky pro zrušení a odložení buzení.

Bohužel zatím neexistuje možnost jak tento dialog rozšířit, obejít nebo ovlivnit. Takže jsem hledal další možnosti zobrazení budíku v předem zadaný čas. Po pátrání na různých fórech, napsání několika dotazů a zkušeni nalezených tipů jsem nenašel žádnou vhodnou



Obrázek 5.3: Ukázka vytvořeného budíku pomocí třídy Alarm z API Microsoft.Phone.Scheduler

možnost, jak zajistit, aby aplikace běžela na pozadí. Toto je velké omezení platformy WP7.

Po zjištění této situace jsem tedy musel prvotní verzi aplikace přepracovat a zajistit, aby si aplikace sama hlídala, kdy má uživatele vzbudit. Toto znamená, že uživatel musí mít během spaní aplikaci zapnutou.

V aplikaci tedy musela přibýt další stránka, která zobrazuje čas a na pozadí běží časovač, který aktualizuje zobrazovaný čas a také hlídá, kdy má aplikace začít uživatele budit. Pokud aktuální čas odpovídá nastavenému času buzení, tak se zobrazí další stránka s pokročilou funkcí vybranou uživatelem, kterou musí splnit, aby se budík vypnul.

Při hledání řešení problému s během aplikace na pozadí, jsem našel alespoň postup [5], jak zajistit, aby se automaticky nezamykala obrazovka po určité době nečinnosti. Stačí do aplikace přidat jeden řádek kódu (viz. následující obrázek). Toto nastavení je v aplikaci konfiguračně závislé. Také jsem vymyslel řešení jak zabezpečit, aby aplikace běžela i pod zamčenou obrazovkou. Objevil jsem možnost, jak odchytit uzamčení a odemčení telefonu. Pokud nastane tato situace, tak se upraví časovač, který běží v aplikaci a poté dokáže běžet i pod uzamčenou obrazovkou.

```
PhoneApplicationService.Current.UserIdleDetectionMode =  
    IdleDetectionMode.Disabled;
```

Obrázek 5.4: Kód pro vypnutí automatického uzamykání obrazovky.

Při řešení předchozího a mnoha dalších problémů nebo různých nejasností během vývoje aplikace mi pomohla knížka Programming Windows Phone 7 [7].

```

public App() // konstruktor
{
    RootFrame.Obfuscated += Obfuscated;
    RootFrame.Unobfuscated += Unobfuscated;
}

private void Unobfuscated( object sender, EventArgs e )
{
    // kod, který se má vykonat pokud nastalo odemčení
    // obrazovky
}

private void Obfuscated( object sender, ObfuscatedEventArgs e )
{
    // kod, který se má vykonat pokud při uzamykání obrazovky
}

```

Obrázek 5.5: Příklad kódu pomocí kterého je možné odchytil zamčení a odemčení obrazovky.

5.1 Akcelerometr

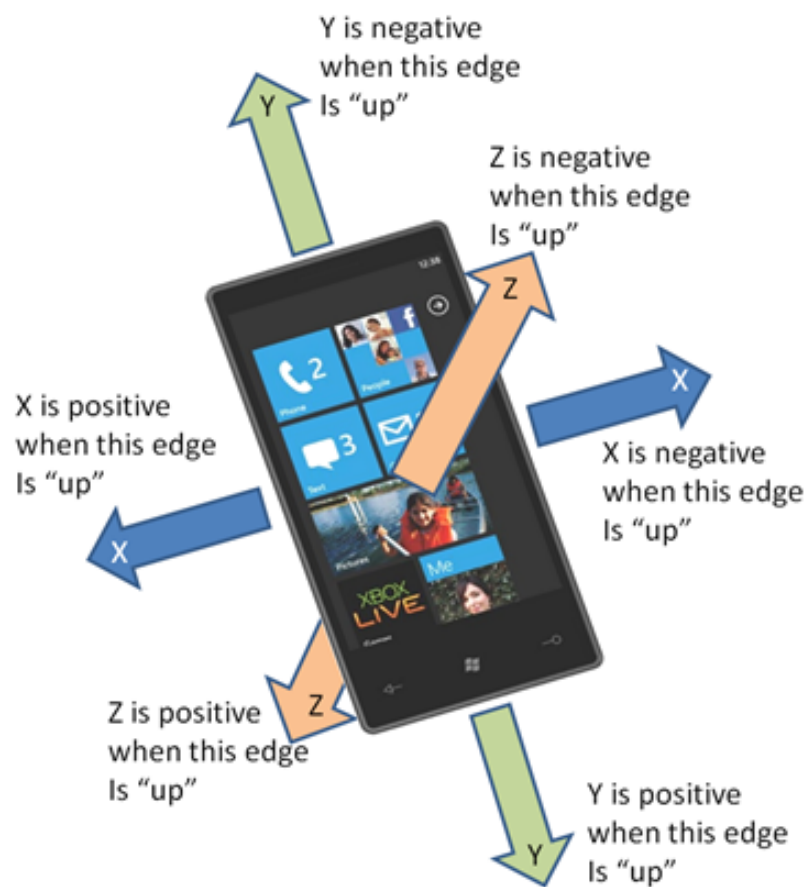
Akcelerometr měří zrychlení ve všech třech osách (X, Y, Z). Samotné hodnoty se pohybují v intervalu od -1 do +1. Pokud je hodnota na nějaké ose rovna 0, tak je telefon položen přímo na tuto osu.

Akcelerometr vrací data ze senzoru každých několik milisekund (tuto hodnotu je možné změnit). Je pouze potřeba nastavit metodu, která se zavolá, pokud přijdou nové hodnoty z akcelerometru. Nastavení této metody a zapnutí akcelerometru je zobrazeno na následující stránce.

V samotné aplikaci uživatel před spuštěním budíku nastaví, jak moc intenzivně musí s telefonem třepat, aby se budík vypnul. Tato nastavená hodnota se používá jako práh při samotném výpočtu síly třepání s telefonem. Funkce, která na základě předešlých a aktuálních dat a nastaveného prahu vrací, zda uživatel zatřepal s telefonem dostatečně intenzivně je znázorněna na pozdějším obrázku.

5.2 Nastavení budíku

Jako další věc jsem se rozhodl upravit obrazovku se samotným nastavením budíku, kde si uživatel zvolí typ buzení (rovnice, QR kód, atd.), dále čas buzení, délku odložení budíku a podle zvoleného typu budíku další nastavení. Tuto obrazovku jsem od prvních verzí příliš neměnil. Pro výběr různých hodnot používám komponentu `ListPicker`, která je součástí knihovny `Windows Phone 7 Control Toolkit`. Podle zvoleného typu budíku se uživateli zobrazují další možnosti. Například jestliže zvolí jako typ budíku vyřešení rovnice, tak se mu zobrazí další nabídka, kde si vybere obtížnost generované rovnice. Dále si na této stránce může zvolit, zda telefon bude zvonit a zároveň vibrovat nebo pouze zvonit či vibrovat. Dále si také aplikace pamatuje poslední zvolené nastavení, které ukládá do XML souboru. Po opětovném otevření tohoto okna se uložené nastavení načte z XML souboru a uživatel



Obrázek 5.6: Vektory sil akcelerometru. Převzato z [2].

```
Accelerometer acc = new Accelerometer();
Acc.ReadingChanged +=
new
    EventHandler<AccelerometerReadingEventArgs>(acc_ReadingChanged);
acc.Start();
```

Obrázek 5.7: Ukázka inicializace akcelerometru, nastavení funkce, která se bude volat při příchodu nových dat a samotné zapnutí akcelerometru.

```

private static bool CheckForShake(AccelerometerReading last,
    AccelerometerReading current, double threshold)
{
double deltaX = Math.Abs((last.Acceleration.X -
    current.Acceleration.X));
double deltaY = Math.Abs((last.Acceleration.Y -
    current.Acceleration.Y));
double deltaZ = Math.Abs((last.Acceleration.Z -
    current.Acceleration.Z));

return (deltaX > threshold && deltaY > threshold) || (deltaX
    > threshold && deltaZ > threshold) || (deltaY > threshold
    && deltaZ > threshold);
}

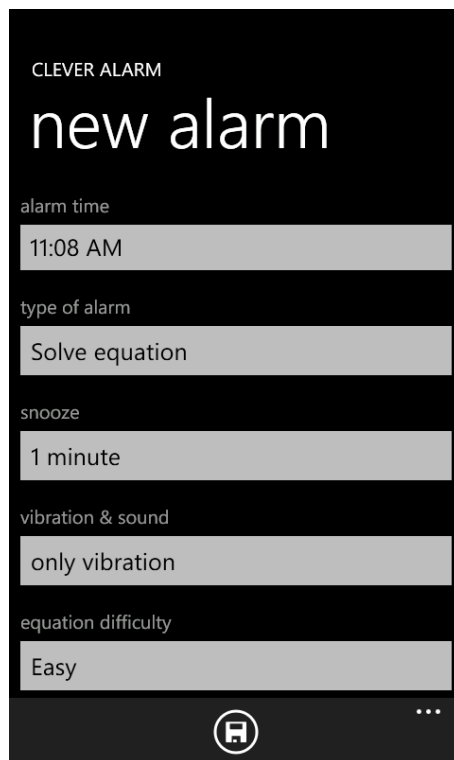
```

Obrázek 5.8: Funkce, která na základě nastaveného prahu, aktuálních a předešlých dat z akcelerometru vypočítá, zda došlo k dostatečně velkému zatřepání s telefonem.



Obrázek 5.9: Výsledný dialog, který se zobrazí uživateli pokud má začít třepat s telefonem.

nemusí opět nastavovat všechny hodnoty.



Obrázek 5.10: Upravená obrazovka s nastavením nově vytvářeného budíku.

5.3 Řešení rovnice

Další typ budíku, který jsem se rozhodl implementovat, byla možnost vypnutí budíku vyřešením zadané rovnice. Opět si uživatel v nastavení budíku zvolí obtížnost generované rovnice (tři úrovně) a v nastavený čas začne budík zvonit a aplikace zobrazí vygenerovanou rovnici, kterou musí uživatel vypočítat. Generované rovnice sestávají pouze ze znamének + nebo - a čísel v určitém intervalu, který se liší podle zvolené obtížnosti. Výsledná stránka s rovnicí je vidět na dalším obrázku.

5.4 Zdřímnutí

Dále jsem se začal pracovat na trochu jiném typu budíku. Kde uživatel nevybírání čas, kdy bude budík zvonit, ale vybírá za jak dlouho bude zvonit. Tato hodnota se vybírá pomocí komponenty `TimSpanPicker`, která je součástí knihovny `Coding4Fun`. Na základě zkoumání a účelu toho druhu buzení jsem se rozhodl hodnoty nabízet po 5-ti minutových krocích. Při používání jsem zjistil, že je často dobré vědět v jaký čas bude budík zvonit, takže jsem aplikaci vylepšil o zobrazení času, kdy bude budík zvonit. Dále jsem tento druh buzení oddělil od standardního budíku, aby to uživatele příliš nemátlo a celkově se ovládání zjednodušilo. Takže přímo na hlavní obrazovce si zvolí zda chce standardní budík nebo právě tento odlišný druh budíku. Dále má také tento typ budíku oddělené vyzvánění od hlavního budíku,



Obrázek 5.11: Stránka s vygenerovanou rovnicí, kterou musí uživatel vypočítat.

takže je možné mít pro oba budíky nastavené jiné buzení. Také si pamatuje uživatelské poslední nastavení. Vlastní obrazovka s nastavením zdřímnutí je podobná jako u standardního budíku. Navíc je pouze přidáno tlačítko pro změnu zvuku buzení a zobrazení času buzení podle nastaveného trvání.

5.5 QR kód

Po otestování funkčnosti zdřímnutí, jsem začal pracovat na další zajímavé možnosti vypnutí budíku a to pomocí načtení čárového kódu. Pro načítání čárových kódů mobilním fotoaparátém jsem využil knihovnu ZXing Barcode Scanning Library. Jako první bod jsem vytvořil testovací stránku, která bude sloužit pro načtení příslušného čárového kódu, který bude muset uživatel pro vypnutí budíku znovu načíst. Bohužel se mi často nepodařilo přečíst čárový kód napoprvé. Zkoušel jsem hledat v čem je problém, ale žádné řešení jsem nenašel. Je možné, že optika v telefonu nebyla ideální (poškrábaná, atd.). Takže jsem vyzkoušel místo čárových kódů použít QR kód. Čtení QR kódů se dařilo bez problémů, takže jsem se rozhodl používat v aplikaci QR kódy místo čárových kódů. Samotná implementace funkčnosti čtení QR kódů nebyla příliš složitá díky bohatým tutoriálům. Po načtení QR kódu v nastavení budíku se uživateli zobrazí načtený text a může potvrdit vytvoření nového budíku. V uživatelsky zadaný čas buzení se objeví podobná obrazovka jako při čtení QR kódu, která slouží pro opětovné načtení QR kódu a pokud je aktuální a předchozí načtený QR kód shodný, tak dojde k vypnutí budíku.


```

private void ScanPreviewBuffer()
{
    try
    {
        // pri inicializaci nove stranky se take vytvari trida
        QRcode (QRCodeReader _reader = new QRCodeReader();)
        _photoCamera.GetPreviewBufferY(_luminance.PreviewBufferY);
        var binarizer = new HybridBinarizer(_luminance);
        var binBitmap = new BinaryBitmap(binarizer);
        var result = _reader.decode(binBitmap);

        Dispatcher.BeginInvoke(() => DisplayResult(result.Text));
    }
    catch (Exception ex)
    {
    }
}

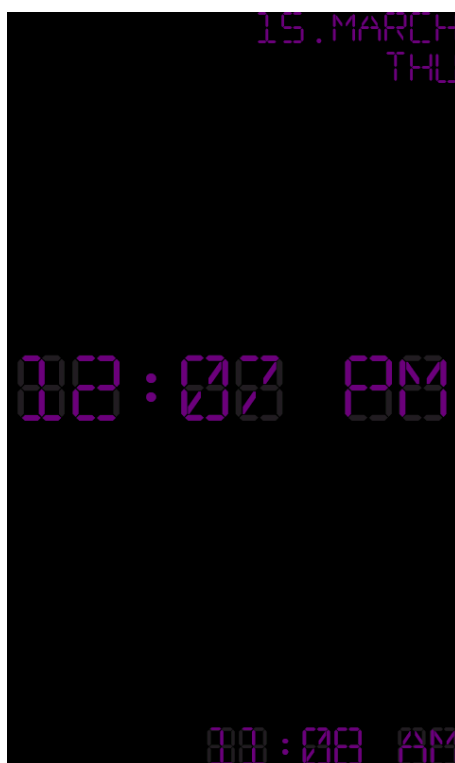
```

Obrázek 5.12: Ukázka funkce, která využívá metody knihovny ZXing Barcode Scanning Library pro čtení QR kódů. Funkce ScanPreviewBuffer() se volá v určitých intervalech a pokaždé se v obraze hledá QR kód.

5.6 Hlavní a čekací obrazovka

Po implementaci těchto základních prvků budíku jsem se rozhodl upravit samotnou stránku s aktuálním časem, která je zobrazena po vytvoření nového budíku až do nastaveného buzení. V prvních verzích aplikace jsem zde zobrazoval čas, který zbýval do buzení. Ale při používání aplikace mi toto řešení nepřišlo jako ideální. Hlavně z důvodu, kdy jsem měl zapnutý v noci budík, probudil jsem se a chtěl jsem vidět kolik je hodin a v kolik vstávám, jenže tato informace tam zobrazena nebyla. Takže jsem se rozhodl tuto obrazovku předělat. Po prozkoumání existujících řešení u jiných aplikací mi jako nejlepší řešení přišlo zobrazovat digitální hodiny uprostřed obrazovky a na krajích datum a čas buzení. Největší inspirací při implementaci této úpravy mi byly digitální hodiny zobrazené v knize [6]. Pokud uživatel otočí mobil do vodorovné polohy, tak se i tato obrazovka otočí o 90° a zvětší se font aktuálního času.

Dále jsem začal také upravovat hlavní obrazovku aplikace. V prvních verzích obsahovala pouze tlačítka pro různé testovací účely. Neměl jsem příliš velkou představu o tom, jak by tato obrazovka měla vypadat. Chtěl jsem, aby byla jednoduchá a uživatel se dostal jedním kliknutím k nastavení samotného budíku a také, aby zobrazovala aktuální čas. Zkoušel jsem si nakreslit různé možnosti rozložení komponent a jako nejlepší a nejhezčí řešení jsem zvolil použití spodní lišty s nastavením, nad ní dvě větší tlačítka a poté velké analogové hodiny. Pomocí tlačítek si uživatel vybere, zda chce použít standardní budík nebo si pouze na chvíli zdřímnout. Výsledná obrazovka byla velice jednoduchá a intuitivní. Při testování jsem, ale narazil na problém, když jsem měl tato tlačítka umístěná nad sebou, tak se občas stalo, že jsem se netrefil do správného tlačítka. Takže jsem udělal menší změnu a tlačítka dal vedle sebe do dvou sloupců, toto řešení se jeví jako ideální.



Obrázek 5.13: Upravená obrazovka se zobrazeným aktuálním časem, datem a časem buzení.



Obrázek 5.14: Hlavní obrazovka aplikace Clever Alarm.

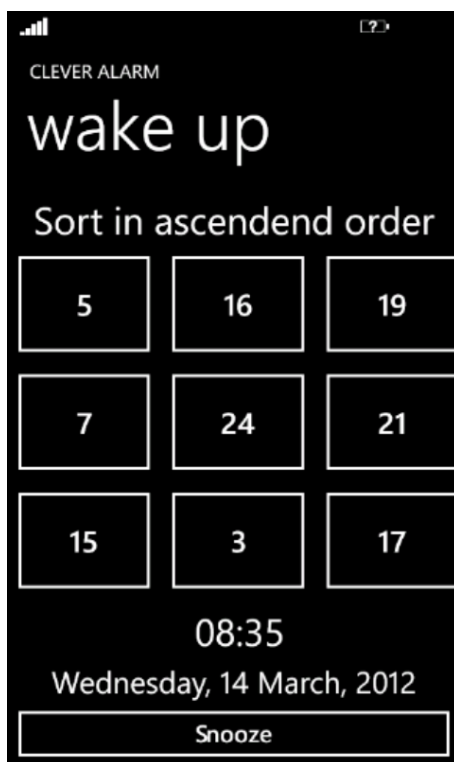
5.7 Testování

Prvotní verze aplikace jsem testoval s pomocí emulátoru. Bohužel je emulátor docela náročný na hardware a není s ním tak pohodlná práce jako s reálným telefonem. Takže jsem si pro pozdější verze zapůjčil různé mobilní telefony s WP7.

Aplikaci jsem testoval na mobilních telefonech značky HTC. Jednalo se o modely HD7, Mozart a Titan. Jakmile jsem měl první odladěnější podobu aplikace, tak jsem tuto verzi zkompiloval a umístil ji na Marketplace jako beta verzi. Poté jsem napsal na různá internetová fóra a hledal zájemce o vyzkoušení aplikace. Sehnal jsem asi 10 lidí. Díky tomuto testování jsem odhalil jednu zásadní chybu v aplikaci, která se občas projevovala při spuštění aplikace a několik menších chyb. Dále mi byly sděleny nějaké připomínky k aplikaci a část z nich jsem upravil. Bohužel ne všichni účastníci beta testování odepsali. Takže jsem se aplikaci rozhodl dále testovat sám na reálném telefonu a postupně doplňovat novou funkčnost a zrovna ji zkoušet.

Jakmile jsem měl všechnu funkčnost hotovou a otestovanou, tak jsem mobilní telefon půjčil několika osobám z mého okolí a nechal je aplikaci si vyzkoušet. Zjišťoval jsem jejich připomínky, potíže s ovládáním a postupně tyto problémy řešil.

Po skončení tohoto průzkumu jsem se rozhodl umístit na Marketplace první veřejnou verzi aplikace. Avšak ve schvalovacím procesu zjistili, že aplikace v několika bodech neodpovídá podmínkám uvedených na Marketplace, takže jsem ji ještě musel upravit a poslat na schválení znovu. Samotný schvalovací proces většinou trvá asi týden.



Obrázek 5.15: Příklad obrazovky, kdy uživatel musí seřadit vzestupně vygenerovaná čísla. Tato čísla se generují v intervalu od 1 do 30. Tento druh buzení je velice účinný.

Kapitola 6

Marketing, hodnocení a budoucnost

Po zveřejnění aplikace jsem se rozhodl podpořit její stahování vytvořením nových témat o aplikaci na různá česká a zahraniční diskuzní fóra. Díky tomuto rozhodnutí jsem získal i první zpětnou vazbu od různých uživatelů. Tyto náměty a připomínky jsem zpracoval do verze 1.1, kterou jsem vydal asi 14 dní po zveřejnění prvotní verze. Hlavní úpravy této verze se týkají především vzhledu, který se na různých telefonech mírně lišil a také jsem přidal čekací dialogy. Jedná se například o čekací dialog při načítání seznamu písniček z hudební knihovny, kdy tato operace trvala delší dobu, pokud měl uživatel mnoho písniček a nemusel vědět co se aktuálně děje.

Jakmile jsem napsal na všechna významná fóra, tak jsem už pouze čekal na připomínky uživatelů, ať už na samotných diskuzích nebo přímo na Marketplace. Za první týden měla aplikace celkově kolem 500 stažení a objevovala se na posledních příčkách při zadání klíčového slova „alarm“ na Marketplace. Ale postupně se začali objevovat první hodnocení aplikace a stahovanost aplikace se velice rychle zvedla. Během tří dnů si aplikaci stáhlo přes 2000 vlastníků mobilních telefonů s WP7. Také jsem si všiml, že na zahraničních serverech vyšly první recenze aplikace. Například na serveru AddictiveTips [4]:

Clever Alarm is one such app, and it is so much more than just an alarm clock. The app will make sure that you are properly awake before it gives up on trying to make you rise and shine. It makes use of mathematical equations, QR codes and number sorting for that purpose.

Další recenze pochází ze serveru WPCentral [8]:

Here's a simple alarm app for your Windows Phone that has plenty of options to help you avoid oversleeping. Clever Alarm has three preset alarm sounds plus the ability to use music files. Plus a pesky set of challenges to prevent the alarm from being shut off too easily.

Samotná aplikace je velice kladně hodnocena. Nejvíce stažení pochází ze Spojených států amerických¹, poté z Kanady² a Velké Británie³. K 17.4.2012 měla aplikace celkový počet stáhnutí přes 3 400. To je velice dobrý výsledek.

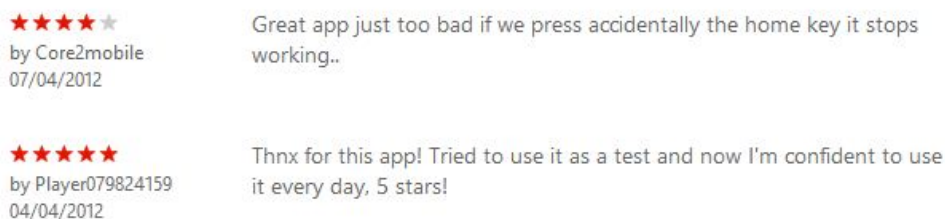
¹<http://www.windowsphone.com/en-US/apps/cdd28225-e361-49dc-8226-5353b0c1b1de>

²<http://www.windowsphone.com/en-CA/apps/cdd28225-e361-49dc-8226-5353b0c1b1de>

³<http://www.windowsphone.com/en-GB/apps/cdd28225-e361-49dc-8226-5353b0c1b1de>



Obrázek 6.1: Hodnocení aplikace uživateli na kanadském Marketplace.



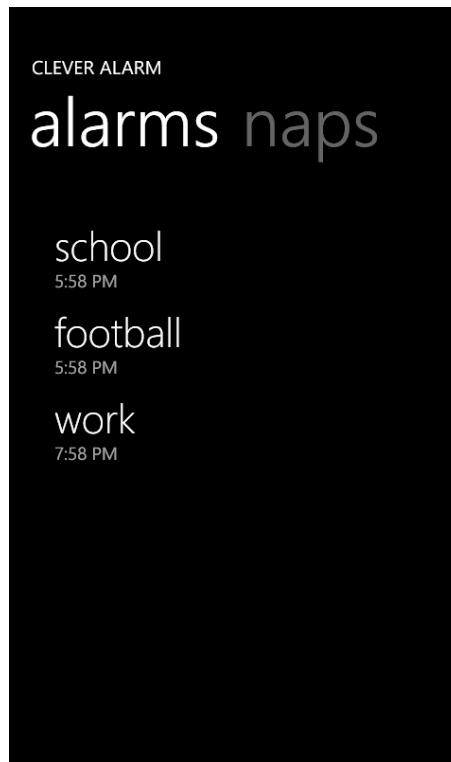
Obrázek 6.2: Hodnocení aplikace uživateli na britském Marketplace.

Díky těmto hodnocením se aplikace rychle rozšiřuje a postupně se zvyšuje její hodnocení na Marketplace a už se nenachází mezi posledními nabízenými aplikacemi s tematikou budíku, ale pomale se dostává mezi první aplikace.

Po sesbírání různých připomínek uživatelů jsem začal pracovat na verzi 1.2. V této verzi jsem jednak lehce optimalizoval kód aplikace, ale hlavně jsem přidal možnost ukládání často používaných budíků. Na tohle téma jsem obdržel několik dotazů a rozhodl jsem se tuto funkčnost implementovat. Při vytváření nového budíku přibyla ve spodní liště ikona, když na ni uživatel klikne, tak je vyzván k zadání názvu budíku a aktuální nastavení se uloží do seznamu budíků se zadaným názvem. Vytvořil jsem tedy dva seznamy (jeden pro budíky a druhý pro zdřímnutí). Na hlavní obrazovku jsem přidal ikonu, přes kterou je možné se dostat k těmto často používaným budíkům. Uživatelé tedy stačí dvě kliknutí pro opětovné nastavení a zapnutí budíku. Stránka s často používanými budíky využívá prvek Pivot, pomocí něj je možné přepínat mezi nastavenými budíky a často používanými časy zdřímnutí. Dále se zde používá komponenta ListBox pro zobrazení uložených budíků a listování v nich.

V budoucnosti se chci aplikaci stále věnovat a rozšiřovat ji podle dalších připomínek uživatelů. Mým cílem s touto aplikací je dostat se mezi 5 nejstahovanějších aplikací s podobnou tematikou. K dosažení tohoto cíle jsou mimo jiné důležité pravidelné aktualizace a dobré hodnocení od uživatelů. Samotný marketing by už měl zůstat v pozadí, pokud by tedy nepřibyla nějaká zásadní nová funkce. Mám připravený seznam úkolů, které chci do budoucích verzí zařadit. Jedná se hlavně o tyto body:

- Zobrazení aktuálního počasí (stupňů) u hodin v čekacím dialogu,
- možnost spuštění hodin bez nastavení budíku,



Obrázek 6.3: Nově přidaná funkčnost. Jedná se o seznam často používaných budíků s jejich názvy. Stránka využívá prvek Pivot a komponentu ListBox.

- výběr zda zobrazovat analogové či digitální hodiny,
- monitorování fází spánku,
- přidání dalšího typu budíku, kdy uživatel pomocí hlasu vypne buzení,
- možnost zapnutí hudby, která bude hrát před usnutím.

Kapitola 7

Závěr

Vytvořil jsem aplikaci, která využívá různých vlastností mobilního telefonu a operačního systému Windows Phone 7. Hlavním cílem aplikace je pomoci lidem ráno vstát z postele, aby už nikdy nezasпали. Aplikace nabízí několik různých pokročilých metod buzení. Jedná se například o nutnost seřadit vzestupně vygenerované číslice, načíst QR kód nebo vyřešit zadanou matematickou rovnici. Aplikace využívá Metro styl, nabízí přívětivé uživatelské rozhraní a další různá nastavení (výběr vlastní hudby, nastavení barvy hodin, ...). Také aplikace umožňuje kromě nastavení času buzení, nastavit i časovou délku, za kterou začne budík zvonit (například 10 minut). Aplikace je přeložena do tří jazyků (angličtina, němčina a čeština), plánuji lokalizaci rozšířit o další světové jazyky. Aplikaci je možné si stáhnout z adresy <http://www.windowsphone.com/en-US/apps/cdd28225-e361-49dc-8226-5353b0c1b1de> pod názvem Clever Alarm.

Aktuálně připravuji již třetí verzi aplikace, která přinese mimo jiné i monitorování fází spánku. Předchozí verze upravovaly například vzhled aplikace, přidaly možnost vyhledávání v hudební knihovně uživatele, dále také možnost uložit si často používané budíky a poté je spustit jedním kliknutím a mnoho dalších vylepšení, která vzešla z připomínek uživatelů.

V této práci jsem popsal postup při samotném vývoji aplikace od popisu WP7 a různých aspektů této platformy, přes analýzu existujících řešení, použité knihovny, návrh aplikace a její implementaci. Vývoj pro operační systém WP7 je poměrně rychlý a největší překážky při vývoji jsou různá omezení této platformy.

Samotná aplikace je velice povedená a to také dokládají různé recenze na internetu, četná uživatelská hodnocení z různých zemí a její postup v žebříčku aplikací na Marketplace. Clever Alarm má velký potenciál, aby se stal velice úspěšnou aplikací pro WP7.

Bylo zajímavé zkoumat tuto platformu a vyvíjet užitečnou aplikaci pro stále se rozrůstající počet uživatelů. Věřím, že aplikaci bude používat stále větší množství lidí, že se jí bude dařit a v budoucnu pro ni plánuji mnoho dalších vylepšení.

Literatura

- [1] DAJBYCH, V.: Jak to chodí s aplikacemi na Marketplace. *Czech MSDN Blog* [online]. Dostupné z: <http://blogs.msdn.com/b/vyvojari/archive/2012/02/24/jak-to-chod-237-s-aplikacemi-na-marketplace.aspx>, 2012-04-04 [cit. 2012-04-27].
- [2] EDSON, D.: Using the Accelerometer on Windows Phone 7. *The Windows Phone Developer Blog* [online]. Dostupné z: http://windowsteamblog.com/windows_phone/b/wpdev/archive/2010/09/08/using-the-accelerometer-on-windows-phone-7.aspx, 2010-09-08 [cit. 2012-05-02].
- [3] GALLOWAY, J.: Leveraging Background Services and Agents in Windows Phone 7 (Mango). *Jon Galloway* [online]. Dostupné z: <http://weblogs.asp.net/jgalloway/archive/2011/11/23/leveraging-background-services-and-agents-in-windows-phone-7-mango.aspx>, 2011-11-23 [cit. 2012-04-27].
- [4] KHALID, H.: Clever Alarm Makes You Solve Problems To Shut Out The Sound [WP7]. *AddictiveTips* [online]. Dostupné z: <http://www.addictivetips.com/mobile/clever-alarm-makes-you-solve-problems-to-shut-out-the-sound-wp7/>, 2012-04-05 [cit. 2012-04-27].
- [5] MARBUTT, J.: WP7 - Disabling the Lock Screen. *dotnet Catch* [online]. Dostupné z: <http://dotnetcatch.wordpress.com/2010/08/17/wp7-disabling-the-lock-screen/>, 2010-08-17 [cit. 2012-05-02].
- [6] NATHAN, A.: *101 Windows phone 7 apps*. Indianapolis, Ind.: Sams, c2011-, ISBN 06723311951, 78 s.
- [7] PETZOLD, C.: *Programming Windows Phone 7*. Redmond, WA: Microsoft Press, 2010, ISBN 978-0-7356-4335-2.
- [8] PONDER, G.: Windows Phone App Review: Clever Alarm [online]. *WPCentral*. Dostupné z: <http://www.wpcentral.com/windows-phone-app-review-clever-alarm>, 2012-04-04 [cit. 2012-04-27].
- [9] STEPHANSEN, H. M.: ZXing 2.0 - Windows Phone 7 library. *Henning.ms* [online]. Dostupné z: <http://www.m0sand.com/henningms/?p=324>, 2012-02-12 [cit. 2012-05-02].

- [10] Chapter 6: Application Lifecycle Differences Between Windows Phone 7 and the iPhone. *Windows Phone Interoperability* [online]. Dostupné z: <http://windowsphone.interoperabilitybridges.com/articles/chapter-6-application-lifecycle-differences-between-windows-phone-7-and-the-iphone>, 2011-04-26 [cit. 2012-04-27].
- [11] Background Agents Overview for Windows Phone. *MSDN* [online]. Dostupné z: [http://msdn.microsoft.com/en-us/library/hh202942\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/hh202942(v=vs.92).aspx), 2012-03-22 [cit. 2012-05-02].
- [12] Windows Phone 7 Silverlight ZXing Barcode Scanning Library. *Windows Phone 7 Silverlight ZXing Barcode Scanning Library* [online]. Dostupné z: <http://silverlightzxing.codeplex.com/>, poslední aktualizace 2010-12-16 [cit. 2012-05-02].
- [13] Windows Phone 7 Toolkit Controls. *Silverlight Toolkit* [online]. Dostupné z: <http://silverlight.codeplex.com/>, poslední aktualizace 2011-12-19 [cit. 2012-05-02].
- [14] Coding4Fun Toolkit for Windows Phone. *Coding4Fun Tools* [online]. Dostupné z: <http://coding4fun.codeplex.com/>, poslední aktualizace 2012-04-17 [cit. 2012-05-02].
- [15] Metro (design language). *Wikipedie* [online]. Dostupné z: [http://en.wikipedia.org/wiki/Metro_\(design_language\)](http://en.wikipedia.org/wiki/Metro_(design_language)), poslední aktualizace 2012-04-30 [cit. 2012-05-02].

Příloha A

Obsah DVD

Příložené DVD obsahuje tyto části:

- Program - obsahuje projekt pro Visual Studio 2010 včetně zdrojových kódů aplikace a potřebných knihoven pro překlad,
- Text - obsahuje PDF soubor bakalářské práce,
- Plakat - obsahuje PDF soubor s plakátem k aplikaci,
- Video - obsahuje video soubor s prezentací vlastní aplikace.