

# 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

## GTD PRO WINDOWS PHONE 7

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JIŘÍ MAŠEK

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

## **GTD PRO WINDOWS PHONE 7**

GTD FOR WINDOWS PHONE 7

**BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

**AUTOR PRÁCE**

AUTHOR

**JIŘÍ MAŠEK**

**VEDOUcí PRÁCE**

SUPERVISOR

**Doc. Ing. ADAM HEROUT, Ph.D.**

BRNO 2012

## **Abstrakt**

Tato práce se zabývá metodou Getting Things Done (GTD) a její implementací na mobilním zařízení s Windows Phone 7. V první části je podrobně popsána metoda GTD a operační systém Windows Phone 7. Dále jsou zde uvedeny diagramy použití a dotazníky, ze kterých vycházel návrh aplikace. Práce pokračuje popisem implementační části a je zakončena závěrem.

## **Abstract**

This thesis deals with a Getting Things Done method (GTD) and its implementation on a mobile device with Windows Phone 7. In the first part the GTD method and the Windows Phone 7 operating system are described. Thereafter the charts of use and questionnaires are introduced from which an application design has come. The thesis continues with a description of the implementation part and finishes with a conclusion.

## **Klíčová slova**

GTD, Getting Things Done, David Allen, Windows Phone 7, WP7, Záležitost, Inbox, Projekt, Kontext

## **Keywords**

GTD, Getting Things Done, David Allen, Windows Phone 7, WP7, Matter, Inbox, Project, Context

## **Citace**

Jiří Mašek: GTD pro Windows Phone 7, bakalářská práce, Brno, FIT VUT v Brně, 2012

# GTD 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.

.....

Jiří Mašek

13. května 2012

## Poděkování

Rád bych zde poděkoval vedoucímu práce Doc. Ing. Adamu Heroutovi Ph.D. za odborné vedení. Také bych rád poděkoval Pavlu Maximenko, který mi velmi pomohl s návrhem ikon aplikace a mému bratrovi Jaroslavu Maškovi za pomoc při překladu aplikace do anglického jazyka. Poděkování také patří mé rodině a přítelkyni za pomoc a podporu.

© Jiří Mašek, 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

<b>1 Úvod</b>	<b>2</b>
<b>2 Metoda GTD</b>	<b>3</b>
2.1 Smysl GTD . . . . .	3
2.2 Princip GTD . . . . .	4
2.3 Význam datumu a času v metodě GTD . . . . .	9
2.4 Současný stav . . . . .	9
<b>3 Windows Phone 7.5 (Mango)</b>	<b>11</b>
3.1 Uživatelské rozhraní Metro . . . . .	12
3.2 Pokyny pro vývoj . . . . .	12
3.3 Základní prvky UI . . . . .	13
3.4 Životní cyklus aplikace . . . . .	15
3.5 Hardwarové požadavky . . . . .	17
3.6 Marketplace . . . . .	17
<b>4 Návrh aplikace</b>	<b>18</b>
4.1 Use-Case diagramy . . . . .	18
4.2 Dotazník . . . . .	18
4.3 SketchFlow . . . . .	21
4.4 Návrh databáze . . . . .	22
4.5 Databáze na serveru . . . . .	23
4.6 Použité knihovny . . . . .	25
<b>5 Implementace</b>	<b>27</b>
5.1 Databáze . . . . .	27
5.2 Uživatelské rozhraní . . . . .	29
5.3 Jazyková lokalizace . . . . .	35
5.4 Synchronizace . . . . .	35
5.5 Testování . . . . .	36
<b>6 Závěr</b>	<b>37</b>
<b>A Obsah CD</b>	<b>40</b>
<b>B Dosažené výsledky</b>	<b>41</b>
<b>C Email</b>	<b>44</b>

# Kapitola 1

## Úvod

Pro dnešní uspěchanou dobu je již téměř nutností naplánovat si čas tak, abychom stihli všechny povinnosti a úkoly a mohli žít svůj život spokojeněji a klidněji. Každý člověk si během života vytvořil nějaký systém pro správu svého času. Ať už mu byl doporučen, našel ho na internetu, vyčetl v knížkách nebo si vymyslel zcela vlastní. Bez umění spravovat vlastní čas bychom nebyli schopni odvádět kvalitní práci či vést spokojený život. O správě času úkolů byly vydány spousty knih, článků, pořádáno mnoho konferencí a přednášek, ale dosud se nepodařilo objevit žádnou metodu správy času, která by byla univerzální.

V této práci se zabýváme metodou GTD, která sice též není zcela univerzální, ale dle slov David Allena se ještě nesetkala se špatným výsledkem. Metoda GTD je založena na sběru dat do co nejmenšího počtu míst. Většina lidí má dnes vlastní mobilní telefon s operačním systémem. Proto se v této práci věnujeme metodě GTD ve Windows Phone 7.

Aplikací implementující metodu GTD již existuje několik, ale naším cílem je vytvořit interaktivní aplikaci, která bude mít rozšířené možnosti zadávání jednotlivých záležitostí. V současných aplikacích lze zadávat záležitosti pomocí klávesnice. V této práci si klademe za cíl tyto možnosti rozšířit na kresbu prstem a nahrávání hlasu.

Metodu GTD, její princip a smysl popisujeme v [2](#) kapitole. V kapitole [3](#) píšeme o Windows Phone 7, kde popisujeme uživatelské rozhraní a jeho prvky, hardwarové nároky na zařízení a také životní cyklus aplikace. V práci se dále zabýváme návrhem aplikace, popsané v kapitole [4](#). V návrhu práce rozebíráme prvotní návrh Use Case diagramů, dotazník a návrh SQL databáze. V [5](#). kapitole popisujeme vlastní implementaci, kde vysvětlujeme funkčnost jednotlivých stránek, synchronizaci, databáze atd. A výsledky celé práce, její zhodnocení a budoucnost popisujeme v závěru, v kapitole [6](#).

## Kapitola 2

# Metoda GTD

Informace v této kapitole jsou čerpány z knihy *Mít vše hotovo: jak zvládnout práci i život a cítit se při tom dobře* [4] a z knihy *Aby vše klapalo: jak hravě zvládat pracovní i životní výzvy* [5] od Davida Allena.

GTD neboli Getting Things Done je nová metoda správy času a práce. Tato nová metoda byla vyvinuta Davidem Allenem, který po několikaletém působení a řízení zaměstnanců v malých i mezinárodních firmách získal informace a postupy pro metodu GTD. Postupy byly tvořeny v reálných situacích a díky tomu metoda GTD obsahuje z velké části člověku přirozené principy. Práce Davida Allen vyvrcholila vydáním knihy GTD, kde je metoda popsána a vysvětlena. Je to metoda, díky které dokážeme zvládnout veškeré své úkoly a povinnosti a mít ze života skvělý pocit.

### 2.1 Smysl GTD

Každý pracující člověk se během své kariéry určitě dostal do situace, kdy je na něho kladeno příliš mnoho požadavků v zaměstnání, současně přibývají starosti a problémy v rodině. Cítí se unavený, je ve stresu a má pocit, že již více nemůže zvládnout. Pokud se tyto situace v životě opakují častěji, život se tak stává nezvladatelný a vznikají problémy ve vztazích. Člověk stále myslí na úkoly, které nestíhá udělat, je ve velkém stresu a každá maličkost ho rychle vyvede z míry. V těchto chvílích nastávají krizové situace a je třeba hledat pomoc. O řešení takových situací je nepřeberné množství knih, seminářů, přednášek o pozitivním myšlení atd., avšak většina metod zabývajících se stejnou či obdobnou problematikou trpí jedním nebo kombinací následujících chyb.

1. Příliš zjednodušené
2. Neúplné
3. Nepřirozené

Metoda GTD těmito problémy netrpí v žádném ohledu. Jako důkaz zde budu citovat Davida Allena: „To, jak dobře GTD funguje, lze myslím doložit na překvapivém fenoménu jeho rychlého šíření mezi IT, internetovou a blogovou komunitou. V době, kdy píše tento text, přibývá každý den na blozích jen v angličtině průměrně padesát nových a jedinečných příspěvků, které se nějakým způsobem týkají metody GTD[5].“ Popularita metody GTD stále stoupá a má již mnoho uživatelů. A jak říká David Allen: „Za všechna ta léta od vydání *Mít Vše Hotovo* se ke mně nedostala žádná stížnost, že model nefunguje[5].“ Hlavní

smysl a přínos metody GTD je především v její funkčnosti, možnostech, flexibilitě, ale také v tom, že každý, kdo se s metodou setkal a zkusil ji používat alespoň nějakou dobu, pocítil příval pozitivní energie se zvýšenou koncentrací, opadla velká část stresu, či dostal mnohem větší sebejistotu v organizaci své práce.

## 2.2 Princip GTD

V následující sekci práce se pokusíme popsat principy a postupy metody GTD, ke kterým se snažíme v aplikaci co nejvíce přiblížit. Jak již bylo uvedeno, cílem GTD je snížit stres, docílit pozitivního myšlení a získat lepší pocit sám ze sebe. K dosažení těchto hodnot zavádí metoda GTD 5 jednoduchých kroků, které tvoří základní model metody.

### 2.2.1 Sběr

Lidský mozek je nástrojem, který dennodenně sbírá všechny úkoly, požadavky, představy, myšlenky, zkrátka vše co během dne přijde. Metoda GTD pro toto vše zavádí jednotný pojem *záležitosti*. K většině těchto záležitostí si člověk vytvoří vnitřní závazek, protože by je rád dovedl do cíle. Mozek obsahuje krátkodobou paměť, která funguje jako paměť RAM počítače. Tato paměť si uchovává všechny záležitosti, ke kterým si člověk vytvořil vnitřní závazek a které jsou nedokončené. Aby paměť RAM fungovala optimálně, není dobré ji přepřehňovat, ale většině lidí kapacita paměti nestačí. Záležitosti člověku v případě zaplněné paměti RAM vyskakují, vyrušují a nejhorší je to, že ho obtěžují v tu nejnevhodnější dobu.

Většina metod v tomto případě říká, запиšte si to. Ale právě zde tyto metody selhávají, jelikož nestačí si záležitost jenom zapsat, ale také vědět kam byla poznámka zapsána a kde ji v případě potřeby najdeme. Hodně lidí si věci zapisuje, ale na různá místa, různá zařízení a nakonec jsou stejně ve stresu, protože zapomenou, kam si potřebné poznámky udělali. Metoda GTD tomuto jevu předchází právě sběrem záležitostí. Jde o **vyprázdnění paměti RAM** do místa, kterému naprosto důvěřujeme a můžeme do něho kdykoli nahlédnout. Těchto míst může být i více, avšak jejich počet by měl být omezen. Příkladem sběrných míst je emailová schránka, telefon s Windows Phone 7 a poštovní schránka.

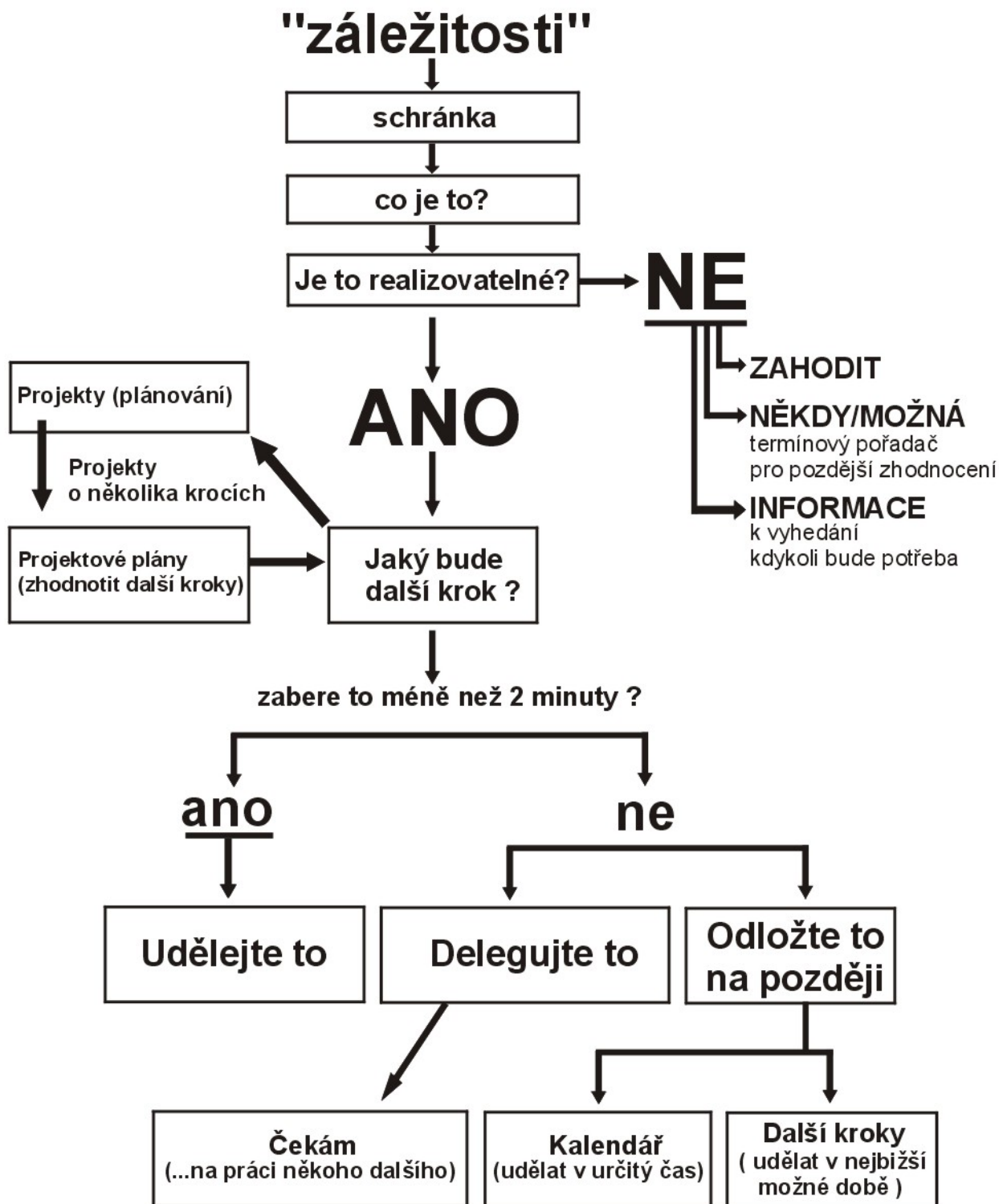
Metoda GTD sběrem záležitostí myslí **100%** sběr všech úkolů, které se během dne objeví, od maličkostí až po zásadní věci, od nákupu chleba po dovolenou, od telefonátu sekretářce po motivaci zaměstnanců. Teprve potom, až paměť bude prázdná a nebude člověka obtěžovat, bude ten následně schopen koncentrace a jeho produktivita se tak bude mnohonásobně zvyšovat.

### 2.2.2 Zpracování

Další částí modelu je zpracování sesbíraných záležitostí. Tento bod obsahují i další metody zabývající se Time Managemenem, avšak nedosahují velkého úspěchu proto, že zpracovávají nekompletní seznamy úkolů. GTD má však v tomto momentu **100%** všech záležitostí sesbíráno a jsme schopni je zpracovat.

Zpracovává se záležitost po záležitosti, což může uživatele odradit, protože si řekne: „to bude trvat dlouho“, avšak za chvíli pozná, že tomu tak není. Zpracování probíhá dle diagramu, který je uveden na obrázku 2.1. U každé záležitosti bychom si měli položit, na první pohled primitivní, otázku „**Co je to?**“. Málokdo se na chvíli pozastaví a ujasní si jaká byla původní myšlenka. Často si po zodpovězení této otázky uvědomíme několik dalších věcí, které nám pomohou odpovědět na další otázku a tou je „**Je to realizovatelné?**“.





Obrázek 2.1: Diagram metody GTD [4]

Pokud odpovíme ne, můžeme záležitost buď smazat, v případě, že ji nebudeme již potřebovat. Můžeme ji odložit, v případě, že ji možná budeme schopni realizovat později nebo ji založit jako informaci. Odpověď ano použijeme na ty záležitosti, u kterých víme, že je lze posunout dále, či provést. V těchto případech je nutno si uvědomit, zdali je to **projekt** a jaký je **další krok**.

Pojmem projekt jsou myšleny takové záležitosti, které ke svému dokončení potřebují více kroků. Jako příklad můžeme uvést opravu auta. Musíme zavolat do servisu, auto tam odvézt a poté si pro auto přijet. Pokud záležitost považujeme za projekt, měli bychom ji dát do složky projektů. Dále bychom si měli stanovit další nutný krok, viditelnou činnost, která nás přiblíží k jejímu dokončení. Jakmile stanovíme další krok máme následující tři možnosti.

1. **Udělejme to** - Pokud krok zabere méně než 2 minuty udělejme ho hned.
2. **Delegujme to** - Pokud krok bude trvat déle než 2 minuty, ale my nejsme ti praví pro tento úkon, delegujme to příslušné osobě.
3. **Odložme to** - Pokud krok bude trvat déle než 2 minuty a my jsme ta pravá osoba, odložme to na později do jednoho ze seznamů.

### 2.2.3 Zorganizování

Následujícím krokem je zorganizování získaných záležitostí a kroků, jejich zařazení do správných kategorií tak, abychom je našli ve správnou chvíli. Opět zde jde o to, aby náš mozek nemusel stále myslet na to co má udělat, ale byl si jistý, že až ta chvíle přijde, tak budeme mít možnost zjistit, co dělat.

Pojem projekt jsme již zmínili. Jedná se o záležitost, která k dokončení potřebuje více viditelných kroků. Projektem mohou být i malé či drobné úkony, což se může zdát zbytečné, ale často se stává, že si člověk řekne, že musí udělat dva úkony, aby splnil nějaký úkol. Provede první a na druhý zapomene. Proto je vhodné si i dvoukrokové záležitosti zapisovat do složky projektů, abychom stále měli někde informaci, že část je hotová, ale ještě něco chybí. Seznam projektů není třeba nijak zvlášť třídit a seřazovat, jen je důležité, abychom tento seznam v pravidelných intervalech revidovali a aby každý projekt měl zadán alespoň jeden následující viditelný krok.

Po splnění dostatečného počtu kroků se projekt dostane do stavu, kdy se realita velmi podobá naší představě o výsledku projektu a můžeme si tento projekt označit jako dokončený. Nyní máme před sebou jednotlivé kroky, kde ty 2 minutové a dokončené nás již nemusí zatěžovat, jelikož jsou hotové, ale musíme kontrolovat ty, které trvají déle nebo na ně čekáme.

Takové záležitosti lze rozdělit na ty, které jsou vázány na konkrétní den nebo čas a na ty, které je potřeba udělat v nejbližší volné chvíli. Záležitosti vázané na konkrétní čas nebo den patří do diáře či kalendáře nebo někam, kde budeme mít tyto položky viditelné v pravou chvíli. Kdežto záležitosti, které je potřeba udělat co nejdříve, je dobré si zařadit do seznamů dle **kontextu**. Většinou se počet záležitostí, které se musejí udělat co nejdříve, pohybuje okolo 50-150. To se může stát velmi nepřehledné a proto je dobré si k nim přiřazovat kontexty. Pokud budeme mít záležitosti, které je potřeba udělat co nejdříve, rozděleny dle kontextů, budeme mít možnost si zobrazit v danou chvíli ty právě záležitosti, které v daný moment můžeme vykonat. Například k záležitostem typu „Zavolat vedoucímu“, „Zavolat babičce“ apod., nutně potřebujeme telefon. Využijeme tedy volné chvíle kdy máme u sebe telefon a zobrazíme si záležitosti v kontextu „Telefon“.

## 2.2.4 Zhodnocení

Pojmem zhodnocení myslíme princip, který nám umožní efektivně a flexibilně vykonávat jednotlivé záležitosti v ten pravý čas. Není totiž jednoduché si na záležitost, kterou máme zapsanou, vzpomenout tehdy, kdy potřebujeme. Principem je, že po každé vykonané záležitosti je vhodné se podívat co bychom ještě měli udělat, čímž je dostaneme do podvědomí.

Metody zabývající se podobnou tematikou říkají, že bychom si měli vytvořit předem přesný časový plán dne, dle kterého se budeme celý den řídit. Zde opět tyto metody selhávají, protože v pracovním procesu se k člověku dostává velmi mnoho neočekávaných záležitostí, které prostě v danou chvíli mají přednost. Například, když za námi přijde vedoucí a požádá nás o splnění důležitého úkolu. Neváháme a uděláme to, i když nám to kazí plány. Když za den přijde takovýchto nevyzpytatelných záležitostí více, začneme nestíhat svůj připravený rozvrh dne, dostáváme se pod tlak a do stresu. Metoda GTD naopak říká, že bychom se měli po každé vykonané záležitosti, kterou si můžeme označit jako dokončenou, podívat, jaké další záležitosti ještě máme splnit. To nám umožňuje udržet si přehled, aniž bychom byli zavázáni vykonat tyto záležitosti přesně na čas. Což znamená, že když přijde vedoucí s naléhavou žádostí, tak ji v klidu splníme a náš mozek se nestresuje tím, že něco nestíháme, protože má přehled, co se má udělat a prostě tyto záležitosti v rychlosti upraví.

Hodnotit bychom neměli dle metody GTD jenom po každé dokončené záležitosti, ale především každý týden, tzv. **týdenní hodnocení**. Model metody GTD je v tomto bodě tím nejkritičtější místem, jelikož hodnocení není příliš oblíbená činnost. Aby člověk dosáhl úspěchu s metodou GTD, musí kvůli této části modelu upravit svůj styl práce. Týdenní zhodnocení však zabraňuje našemu mozku opět upadnout do původního stavu, kdy je zahlcen. Každý týden bychom si měli projít všechny otevřené záležitosti, projekty i složky Možná/Někdy. Je to chvíle, kdy máme možnost získat přehled o několika předchozích a následujících dnech, revidovat svoje záležitosti, promazat či přidat další kroky. Během týdenního zhodnocení máme čas na:

1. Sebrání a zpracování všech „věcí k vyřízení“,
2. Revizi systému,
3. Aktualizaci seznamů,
4. Ujasnění, aktualizaci a doplnění.

Abychom ze zhodnocení měli dobrý pocit, je třeba mít systém ucelený, mít na **100%** sesbírané veškeré záležitosti tak, aby nic nechybělo. V momentě kdy něco chybí, získáváme pocit nedůvěry v systém, jelikož někde hluboko v sobě víme, že je někde něco špatně, že někde něco chybí, ale nevíme co. Čím více je systém úplnější, tím větší v něj máme důvěru. Čím větší máme důvěru v tento systém, tím větší máme motivaci ho udržovat. A do tohoto stavu se dostaneme při zavedení právě onoho týdenního hodnocení.

David Allen během vývoje metody GTD zjistil, že nejlépe se cítíme týden před dovolenou. Avšak ne kvůli dovolené samotné, ale díky tomu, že si upřesňujeme, uzavíráme, čistíme své závazky a záležitosti. Problém je ten, že to děláme většinou pouze 1 za rok právě před nějakou dovolenou.

## 2.2.5 Vykonání

V této části modelu jde o rozhodování, co ve kterou chvíli udělat. Je velmi složité vybrat z několika záležitostí, které se právě nabízejí. Musíme posoudit, kterou budeme právě nyní vykonávat a kterou ne a současně budeme mít z obou možností dobrý pocit. Neexistuje přesný postup jak bychom se měli v danou chvíli rozhodnout, ale metoda GTD zavádí 3 modely, které nám to přinejmenším velmi usnadní. Základem je mít přehled, který získáme právě poctivým sběrem, zpracováním, zorganizováním a zhodnocením. Dále se musíme řídit i vlastní intuicí, která je podpořena inteligentním přemýšlením o své práci a hodnotách. První model, který nám pomůže při rozhodování, je model čtyř kritérií pro okamžité rozhodování o dalších krocích.

- **Kontext** - Kontext jsme v této práci již zmiňovali. Ale je to právě kontext, který nám jako první říká, co v dané situaci můžeme dělat. Pokud záležitost vyžaduje internet, a my k němu zrovna nemáme přístup, tak záležitost odložíme na později, jelikož ji v daný okamžik nemůžeme vykonat.
- **Dostupný čas** - Dalším kritériem rozhodování je, kolik máme nyní času pro své záležitosti. Pokud máme za 10 minut pracovní oběd, opět to zmenšuje rozsah možností, co můžeme v danou chvíli dělat.
- **Dostupná energie** - Samozřejmě, že energie je také rozhodující. Pokud máme za sebou 12 hodin práce a je už 20:30, nemáme již kolikrát moc sil na náročnější záležitosti např. stěhování nábytku apod.
- **Priorita** - Pokud máme dostatek času a energie, musíme posoudit, která záležitost se nám v daném kontextu nejvíce zúročí? Zde se musíme spolehnout na svou intuici a rozhodnout se, co je pro nás nejdůležitější.

Dalším modelem je Trojitý model pro hodnocení každodenní práce. V momentě, kdy „pracujeme“, tehdy se věnujeme třem typům aktivit.

- **Děláme předem určenou práci** - Předem určenou prací jsou myšleny naplánované záležitosti a jednotlivé po sobě následující kroky.
- **Děláme práci, která se objevila** - Každý den se objevuje náhodná práce, kterou musíme nebo chceme vykonat a tím na chvíli přerušíme svou původní práci.
- **Definujeme svou práci** - Definice vlastní práce spočívá ve sběru záležitostí, jejich zpracování a organizaci, případně i zhodnocení. V tento moment čistíme emailovou schránku, poznámky a rozdělujeme je do jednotlivých projektů, přidáváme jim další kroky a kontexty apod.

Posledním modelem je šestiúrovňový model pro hodnocení naší vlastní práce, který David Allen přirovnává k nadmořské výšce:

- Nad 50 000 stop: Život,
- 40 000 stop: Vize na 3-5 let,
- 30 000 stop: Cíle na příští rok až dva,
- 20 000 stop: Oblasti zodpovědnosti,
- 10 000 stop: Aktuální projekty,
- Vzletová dráha: Aktuální aktivity.

## 2.3 Význam datumu a času v metodě GTD

Čas v metodě GTD je relativní, jelikož metoda GTD nic nenařizuje. Pouze říká, že by člověk měl mít povědomí o svých záležitostech a možnost si je kdykoli najít a během dne pracovat dle toho, jak je to možné. Přesto mnoho uživatelů přidává datum a čas k určitým typům záležitostí a případně i nějaký druh upozornění, jako například budík. Snaží se tímto svému mozku ulevit ještě více. I z našeho průzkumu vyšlo najevo, že většina uživatelů metody GTD přidává ke svým záležitostem datum a čas. Především z toho důvodu je toto aplikováno i v této práci, což umožňuje zobrazovat si všechny následující kroky dle datumu za sebou a mít lepší přehled o nadcházejících záležitostech v kalendáři.

## 2.4 Současný stav

V době psaní této práce jsou na Windows Phone Marketplace přibližně 3-4 aplikace pro metodu GTD. Všechny aplikace se snaží svým způsobem co nejlépe implementovat tuto metodu, snaží se vypadat dle nového uživatelského rozhraní Metro a mají rychlou odezvu. Bohužel ale ve všech je nutné zadávat jednotlivé záležitosti ručně, tedy za pomoci softwarové či hardwarové klávesnice.

Počet aplikací pro metodu GTD a styl zadávání záležitostí nás dovedl k napsání vlastní aplikace pro metodu GTD, která bude schopná konkurovat těm stávajícím, ale bude mít možnost zadat záležitosti i pomocí malování prstem či zadávání hlasem. Z průzkumu vyšlo najevo, že mnoho uživatelů tuto funkčnost nevyužívá. Po konzultacích jsme došli k závěru, že to je způsobeno tím, že k tomu uživatelé nemají patřičné nástroje. Pokud by tuto funkčnost chtěli využívat, museli by si pořídit externí nástroje mimo systém GTD.

Aplikace CleverToDo nabízí správu úkolů. Je založena na metodě GTD a umožňuje vytvářet úkoly, upomínky, zálohy na DropBox, výběr datumu a času, dlaždice a další. Avšak je docela drahý. Moje aplikace by měla být zcela zdarma.



AnyTask! je zdarma, nabízí především synchronizaci s Toodledo serverem. Také implementuje metodu GTD a obsahuje připomínky, dlaždice. Aplikace má docela daleko ke GTD a také proto je možná zadarmo. Rád bych, aby moje aplikace měla ke GTD mnohem blíže než AnyTask!.



Aplikace Chaos Control je opět implementace metody GTD. Jako předchozí aplikace umožňuje přidávat upomínky, synchronizovat se, zadávat datum a čas, obsahuje dlaždice a další. Chaos Control je zdarma a je docela kvalitní aplikací implementující metodu GTD. Avšak schází interaktivní zadávání, které bych rád implementoval.



Lil Todo se jeví jako poměrně povedená aplikace. Obsahuje více méně totéž co předchozí, ale opět zde chybí interaktivní zadávání.



Tabulka 2.1: Tabulka aplikací dostupných na MarketPlace [15]

## Kapitola 3

# Windows Phone 7.5 (Mango)

Windows Phone 7 (WP7) je nový operační systém pro chytré mobilní telefony. Tento operační systém byl vyvinut po nevyhnutelném konci Windows Mobile 6.5. Ve WP7 dochází k velkým změnám. Celá platforma je zaměřena především na uživatelský komfort a pohodlí při ovládání. Vzniká zde zcela nové uživatelské rozhraní, které je popsáno v následující kapitole. Microsoft v této platformě přišel s jednotným hardwarem, kde výrobcům chytrých telefonů nařizuje minimální hardwarové vlastnosti jejich výrobků, aby mohli užívat WP7 na svých zařízeních. Microsoft také zapracoval na životním cyklu aplikace, který je zcela jiný. Další novinkou je Isolated Storage, agenti, notifikace a další [16].

Několik měsíců po vydání operačního systému WP7 vyšla aktualizace pro tento operační systém nazvaná Mango, která aktualizuje operační systém WP7 na WP7.5. Mango přineslo mnoho změn, které se vynutili uživatelé WP7, jelikož některé funkce jim v tomto operačním systému prostě chyběly. V aplikaci Mango například přibylo

- Přepínání úloh - Při podržení tlačítka „zpět“ se zobrazí všechny spuštěné aplikace a můžeme mezi nimi přepínat.
- Diktování SMS zpráv hlasem
- Integrovaní Facebooku a dalších sociálních sítí - Možnost vidět přátele z Facebooku, jejich statusy, jejich zeď a další. Také je zde integrace chatu facebooku přímo do zpráv
- Režim úspory baterie
- Zaostření fotoaparátu - Pouhým dotykem obrazovky na určitém místě se fotoaparát pokusí zaostřit na zvolené místo

a mnoho dalších.



Obrázek 3.1: Přepínání úloh ve WP7 mango [1]

### 3.1 Uživatelské rozhraní Metro

Nové uživatelské rozhraní Windows Phone 7 se jmenuje Metro. Název není jen náhodou stejný jako podzemní vlak jezdící ve velkoměstech. Právě v metru, na vlakových nádražích a na letištích se Microsoft inspiroval a to z jednoho prostého důvodu. Na místech jako metro či letiště se každý člověk ihned zorientuje a to díky mnoha obrázkům, znakům a dalším vizuálním pomůckám, které nám umožňují rychle a efektivně najít cestu tam, kam potřebujeme.



Obrázek 3.2: Znaky z letišť a metra [16]

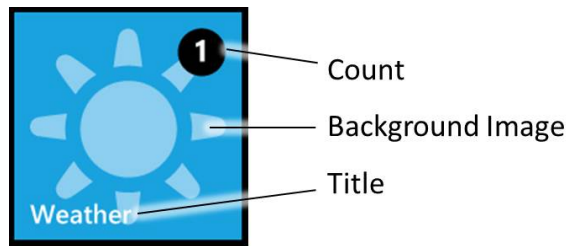
Tento koncept nechal vzniknout novému uživatelskému rozhraní nazvaném Metro. Jeho základem jsou Live Tiles (jako překlad se uchytilo v našich končinách slovo živé dlaždice), které tvoří hlavní obrazovku celé platformy. Dlaždice je vlastně zástupce aplikace, přes kterou se program spouští. Dlaždice se dělí na Aplikační dlaždici (Application Tile) a Sekundární dlaždici (Secondary Tile). Aplikační je zobrazena po připojení na hlavní obrazovku. Každý program má právě jednu tuto dlaždici. Kdežto sekundárních dlaždic může být několik a nebo žádná. Záleží na aplikaci, kolik jich umožňuje vytvořit. Jak již z názvu živá dlaždice vyplývá, dlaždice jsou animované, mají dvě strany a ty se v intervalech otáčejí a posouvají [14].

Z těchto dlaždic je sestavena hlavní obrazovka celého systému. Některé dlaždice jsou systémové a mají stejný úkol jako všechny ostatní, tedy zobrazovat informace a fungovat jako odkaz do aplikací, avšak mají více funkcí. Například dlaždice People se liší od té klasické tím, že je rozdělena na 9 stejných částí, na 9 stejných čtverečků, ve kterých se v pravidelných intervalech mění obrázky našich přátel z Facebooku. Nebo také dlaždice o dvojnásobné šířce. K těmto dlaždicím není povolen přístup a při vývoji aplikace je nemůžete využít.

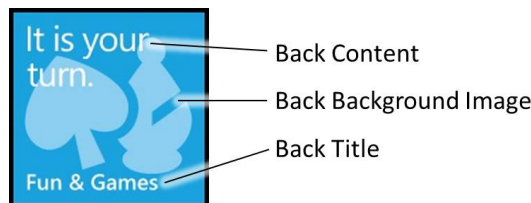
### 3.2 Pokyny pro vývoj

Microsoft vytvořil pro svoji novou platformu pokyny, které by měly vývojářům pomoci při výrobě svých aplikací tak, aby měly co největší pozitivní ohlas.





(a) přední strana dlaždice



(b) zadní strana dlaždice

Obrázek 3.3: Schéma živé dlaždice [14]

Především Microsoft říká, že by naše aplikace měla mít rychlou odezvu. Rychlou odezvou je myšlen i samotný start aplikace. Spuštění aplikace by nemělo trvat déle než 10 vteřin [11]. Je to logické, jelikož uživatelé jsou po delší době skeptičtí k dané aplikaci již při jejím spuštění. Aplikace by po spuštění měla mít schopnost okamžitě reagovat na interakce s uživatelem. Respektive by měla být nějaká viditelná akce, když se uživatel dotkne displeje. Například, když uživatel klikne na tlačítko, které vykonává nějakou delší a náročnější činnost, mělo by se alespoň nějak graficky změnit, aby uživatel na první pohled viděl, že se něco stalo a možná ještě stane. Právě rychlá odezva na dotyky uživateli dodává pocit, že aplikace funguje tak, jak má. Hlavně v dnešní zrychlené době neradi na cokoli čekáme.

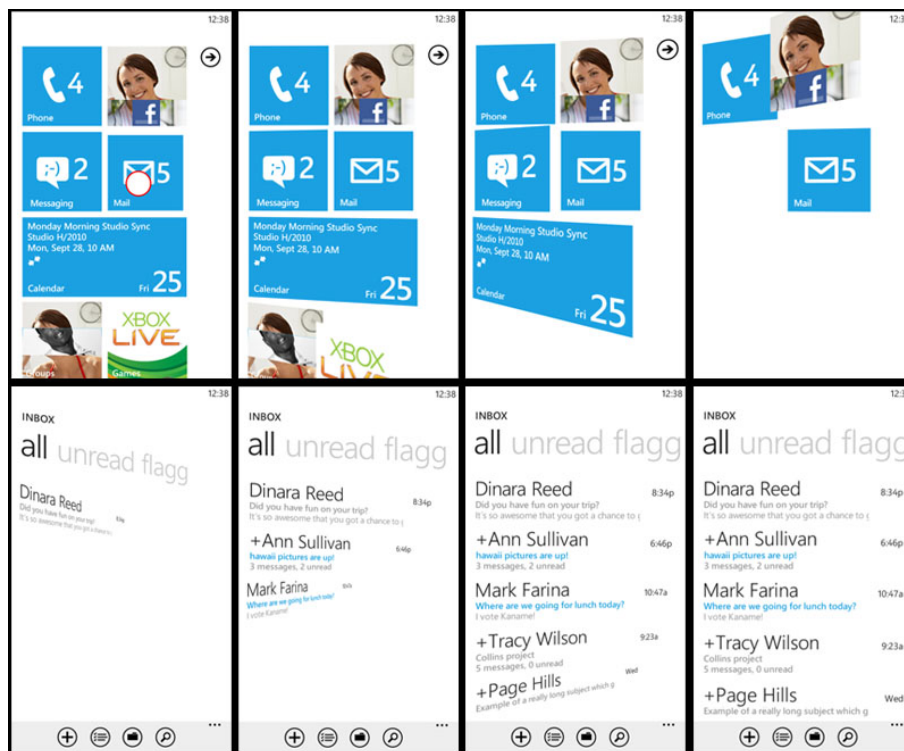
Dále Microsoft doporučuje, aby se při vývoji aplikací pro WP7 využívaly animace. Animace přidávají na zajímavosti a Microsoft proto vybavil WP7 knihovnamí, které animace velmi usnadňují.

### 3.3 Základní prvky UI

V novém operačním systému WP7 je hlavní stránka a stránka aplikací. To se zcela liší od dosud prosazovaných stylů, jako u Iphone nebo Android a dalších, kde je naopak základ plocha s ikonami, podobná právě ploše z klasických Windows. Microsoft šel při vývoji uživatelské prostředí zcela jiným směrem. To se také projevilo ve vývoji aplikací a nabídce prvků pro uživatelské rozhraní.

Před několika lety bylo trendem zobrazit co nejvíce informací na jedné obrazovce tak, aby se nemuselo skrolovat. Dnešní technologie však umožňují rychlé skrolování díky gestům a dotykům. Proto Microsoft vsadil právě na skrolování při vývoji WP7. Názorným příkladem může být hlavní stránka, kde jsou dlaždice ve dvou sloupcích pod sebou a skrolováním hledáme to, co potřebujeme.

Dalším příkladem je stránka s aplikacemi, kde je využit právě jeden ze základních prvků platformy WP7 a to ListBox. Jedná se o klasický seznam položek pod sebou, mezi kterými se skroluje pouhým dotykem prstu. ListBox známe již dlouhá léta, ale ten ve WP7 umožňuje zobrazovat cokoli a jakkoli, jelikož šablonu pro jednotlivé položky seznamu si může každý



Obrázek 3.4: Animace ve WP7 [11]

upravit dle své představy.

Silverlight for Windows Phone Toolkit posouvá skrolování ještě dále. Přináší prvek založený na ListBoxu a zobrazuje jednotlivé položky ve skupinách. Nad každou skupinou je možno zobrazit o jakou skupinu se jedná, opět za pomoci šablony, kterou lze upravit. Také lze zobrazit všechny skupiny a dotykem se přesunout k té požadované, aniž bychom museli dlouho sklorovat v rozsáhlém seznamu. Příkladem může být zobrazení kontaktů, kde se kontakty zobrazují ve skupinách dle abecedy. Příklad je vidět na obrázku 3.5.

V Silverlight for Windows Phone Toolkit je obsažen ještě jeden prvek, který vychází z ListBoxu a to ExpanderView. Jedná se o prvek, který zobrazuje opět seznam položek, ale umožňuje pod jednotlivými položkami zobrazit další seznam. Například, když budeme mít seznam automobilek, tak po rozkliknutí automobilky Škoda se pod ní zobrazí seznam aut, které Škoda nabízí.

### 3.3.1 Panorama Control

Uživatelské rozhraní WP7 obsahuje prvek Panorama Control, který je spojením více stránek do jedné široké stránky, v níž se můžeme posouvat gesty. Celý prvek se skládá z pozadí, které tvoří největší část a posouvuje se zároveň s gesty uživatele. Dále obsahuje titulek stránky a jednotlivé položky, které zaobalují ostatní prvky jako tlačítka, textboxy a jiné.

Panorama Control vykresluje všechny své části najednou, což může mít vliv na výkonost. Tento problém se však dá obejít nastavením prvků na Collapsed, který vykreslování zabrání. Zviditelnění prvků se provede v momentě, kdy se zobrazit mají [3]. Ukázka Panorama Control je na obrázku 3.5.



Obrázek 3.5: Panorama Control ve WP7 [3]

### 3.3.2 Pivot Control

Pivot Control je prvek na první pohled velmi podobný Panorama Control. Je zde zásadní rozdíl v tom, že Pivot Control se nechová jako jedna široká stránka a tudíž nevykresluje všechny své stránky najednou.

Pivot Control je složen z hlavičky a z obsahu, kde v hlavičce jsou vidět jednotlivé nadpisy stránek a obsah zaobaluje všechny ostatní ovládací prvky. Mezi stránkami se přechází opět gesty.

Microsoft nedoporučuje, přesněji zakazuje, zanořovat prvky jako Panorama Control a Pivot Control v jakékoli kombinaci. Pokud bychom zanořili tyto dva ovládací prvky do sebe, dostaneme se do problémů s gesty, jelikož oba dva využívají posun do stran. Také bychom neměli v těchto prvcích využívat další gest, které reagují na posun stranou.

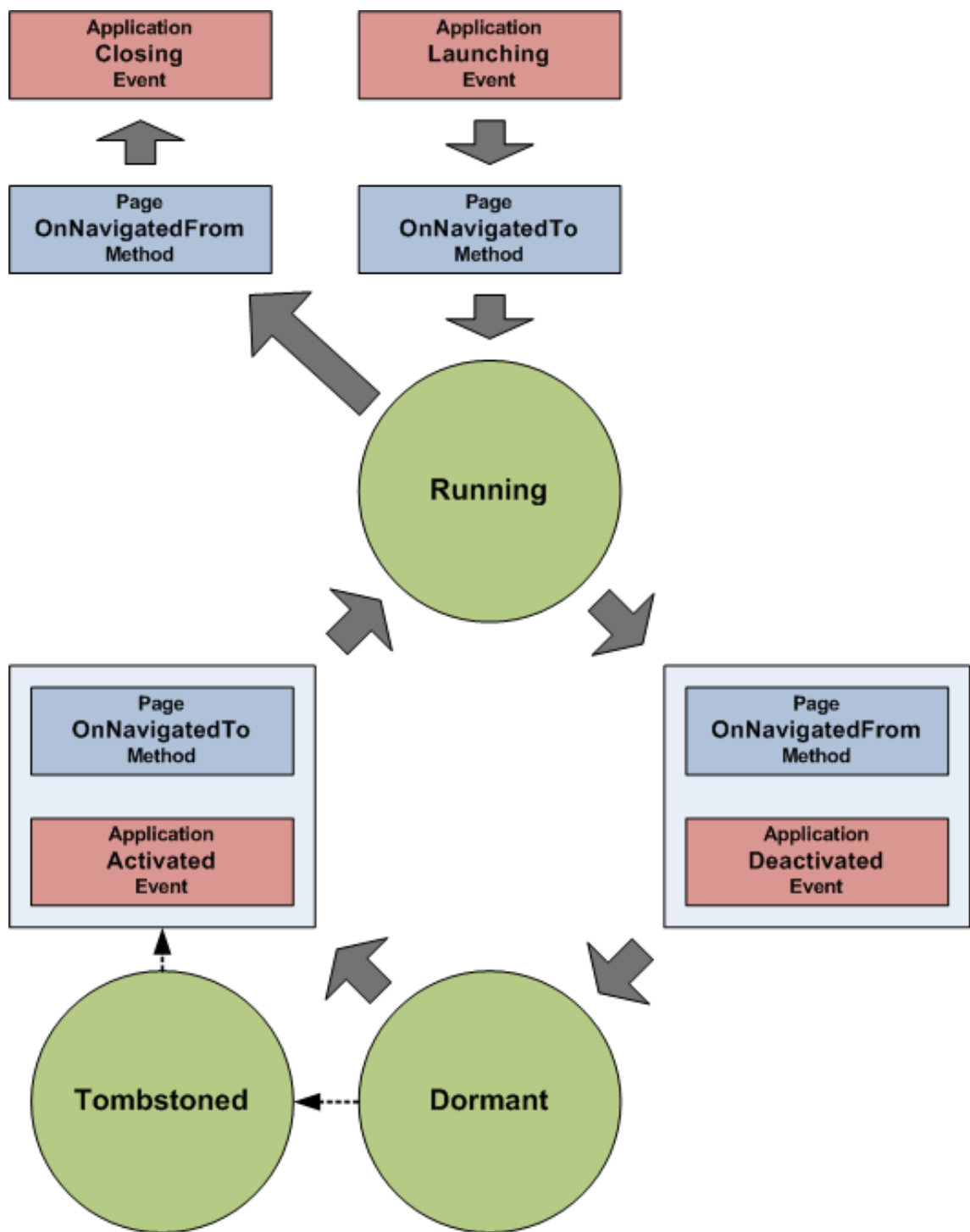
## 3.4 Životní cyklus aplikace

Microsoft vyvinul opět odlišný životní cyklus aplikace, než jsme byli zvyklí dodnes. Celý cyklus se skládá z 3 stavů [10].

1. **Running** - Jedná se o stav, kdy aplikace je v běhu, vykonává svůj kód, reaguje na uživatele či zobrazuje informace na display.
2. **Dormant** - Po navigování se ven z aplikace, například na hlavní obrazovku, se po vyvolání události Deactivated dostane aplikace do stavu Dormant. V tomto stavu jsou pozastaveny všechny procesy aplikace tak, aby nezatěžovaly procesor, ale aplikace zůstala v paměti.

V případě, že se uživatel vrací do aplikace a ta je stále ve stavu Dormant, dochází ke spuštění jejích procesů a aplikace se dostává velmi rychle do stavu, ze kterého uživatel odešel.

V momentě, kdy je aplikace v tomto stavu a je spuštěno několik dalších aplikací tak, že dochází k paměť, potom se aplikace ve stavu Dormant dostává do dalšího stavu.



Obrázek 3.6: Životní cyklus aplikace ve WP7 [10]

3. **Tombstoned** - Do stavu Tombstoned se aplikace dostává pouze ze stavu Dormant. Ve chvíli, kdy dojde k tomuto přechodu, aplikace se odstraní z paměti a ukládá se pouze stav aplikace. WP7 může v jednu chvíli mít uloženo až 5 stavů aplikací. Pokud je požadavek na víc jak 5, aplikace je ukončena úplně.

### 3.5 Hardwarové požadavky

Microsoft při vývoji WP7 došel k závěru, že bude mnohem lepší sjednotit hardware zařízení, na kterých poběží WP7, než kdyby se aplikace musela přizpůsobovat obrovské škále různého hardwaru na různých zařízeních. Proto Microsoft vydal následující požadavky [12].

1. **Tlačítka** - Každé zařízení musí obsahovat tlačítka Start, Hledat a Zpět.
2. **Displej** - Alespoň WVGA (800 x 480), který je schopen vykreslit většinu webových stránek v plné šířce a umožňuje přehrávat video na šířku.
3. **Dotyková vrstva** - Je nutné, aby dotyková vrstva umožňovala minimálně 4 současné dotyky pro rychlé a snadné ovládání.
4. **Datové připojení** - Nutnost podpory celulárního a WiFi připojení.
5. **Paměť** - Minimálně 256MB paměti RAM a alespoň 8GB flash paměti.
6. **GPS** - Zařízení musí obsahovat GPS modul s podporou AGPS.
7. **Akcelerometr** - V zařízení musí být přítomen akcelerometr.

Dále je povolen doplňkový hardware.

1. Kompas
2. Kamera/Fotoaparát
3. Přední kamera/Fotoaparát
4. Gyroskop

### 3.6 MarketPlace

Majitelům zařízení s WP7 není dovoleno nahrávat do zařízení aplikace z PC, natož nelegální. Pro získání nové aplikace musí každý majitel takového zařízení přistupovat na nově vytvořený server jménem **MarketPlace**.

Na MarketPlace jsou umístovány nově vytvořené aplikace a hry, které lze stoprocentně považovat za legální a bezpečné. Je tomu tak díky certifikačnímu procesu, kterým musí projít každá nově nahrávaná aplikace. Po úspěšné certifikaci je aplikace podepsaná certifikátem od Microsoftu.

Aby mohla být aplikace podepsaná certifikátem, musí splňovat stanovené podmínky, o kterých je možno se dozvědět více na stránkách Microsoft Developer Network [8].

## Kapitola 4

# Návrh aplikace

Při projektování aplikace jsem po konzultaci s vedoucím práce vytvořil nejdříve 2 Use-Case digramy, abych si vytvořil lepší představu a používání aplikace. Poté jsem zpracoval dotazník pro získání informací od stávajících uživatelů metody GTD. Nakonec jsem vytvořil návrh databáze pro ukládání dat.

### 4.1 Use-Case diagramy

Diagramy použití neboli Use-Case diagramy mají zobrazovat všechny situace, ve kterých se daná aplikace bude používat. V mém případě jsem vytvořil jeden diagram, který zobrazuje každodenní použití a druhý, který ukazuje roční použití.

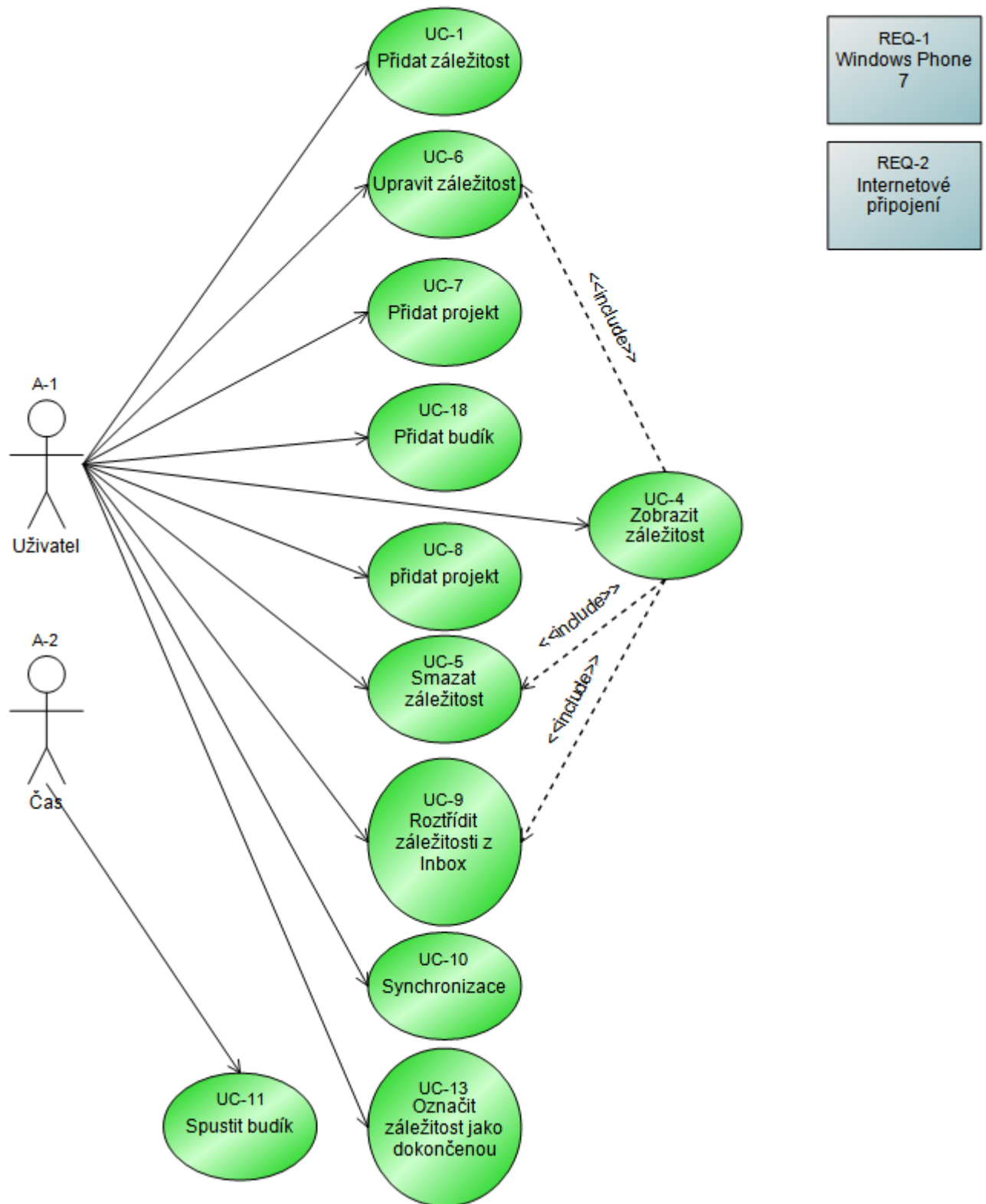
Na obrázku 4.1 je zobrazen diagram denního použití aplikace. Jsou zde vidět možnosti uživatele. Při implementaci aplikace jsem se snažil tímto diagramem řídit, ale samozřejmě došlo k menším úpravám a změnám. Například zobrazení záležitosti není použito zcela tak, jak je uvedeno na diagramu.

Obrázek 4.2 ukazuje roční diagram. Zde jsou zobrazeny akce, které uživatel může provádět během roku a tedy nejsou příliš časté. Většina z nich, jako například synchronizace, byla implementována. Od některých bylo upuštěno. Akce obnovit a zálohovat nebyly implementovány, jelikož synchronizaci jsem považoval za dostačující.

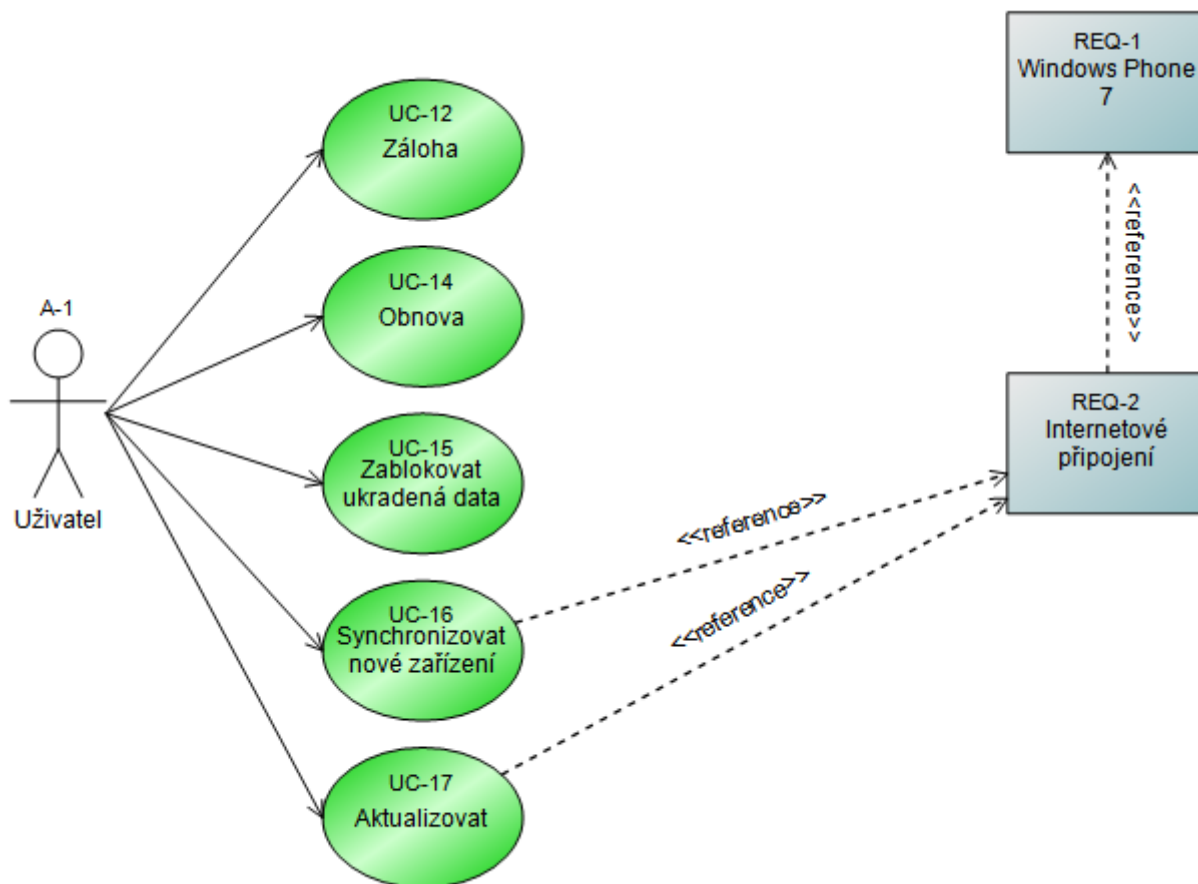
### 4.2 Dotazník

Při navrhování aplikace jsem se potýkal s několika problémy, které jsem po konzultaci s vedoucím práce vyřešil zjištěním preferencí uživatelů pomocí dotazníku. Jedním z problémů bylo, jakým způsobem zobrazit co nejvíce položek tak, aby byly přehledné a čitelné a zároveň tolerovaly vzdušnost WP7.

Dotazník se skládal ze sady 15 otázek. Cílem otázek bylo zjistit, jak uživatelé aplikací, které využívají metodu GTD, tento program používají. Co využívají a co naopak ne. Podle toho jsem se rozhodoval, jakým způsobem zobrazovat jednotlivé položky. Některé otázky byly spíše jen informativní.



Obrázek 4.1: Denní UseCase diagram



Obrázek 4.2: Roční UseCase diagram

Dotazník se skládal z následujících otázek:

1. Používáte kontexty?
2. Jaké je vaše maximum a minimum kontextů?
3. Kolikrát denně procházíte složku Inbox?
4. Zapisujete si položky, které stihnete do 2 minut?
5. Zadáváte k jednotlivým položkám datum a čas?
6. Kolika kliknutími dosáhnete zobrazení obrazovky pro přidání nové položky?
7. S jakými zařízeními či místy v internetu synchronizujete?
8. Pokud provádíte synchronizaci, kolikrát týdně tento úkon provedete?
9. Pokud provádíte synchronizaci, dělá se automaticky nebo ji provádíte manuálně?
10. Synchronizujete častěji přes WiFi/LAN či 3G/HSPDA?
11. Z kolika míst přistupujete ke svým GTD datům? (PC doma, PC v práci, Mobil, Knihovna)



12. Používáte v GTD i papír a tužku? Pokud ano, přepisujete si tyto věci do GTD?
13. Používáte v GTD diktafon či záznamník hlasu? Pokud ano, přepisujete si tyto věci do GTD?
14. Používáte starší záznamy projektů jako šablonu pro vytváření nových? (Třeba jen nahlédnutím jak to bylo?)
15. Kolikrát týdně nahlížíte do dokončených projektů či položek?

Dotazník byl přeložen i do anglického jazyka, jelikož převážná část uživatelů GTD je ze zahraničí. Nakonec jsem tento dotazník, vytvořený v Google Documents, zveřejnil na všech fórech, blozích, facebooku, zkrátka všude, kde by se mohli vyskytovat uživatelé metody GTD. Bohužel přes velkou snahu a zveřejnění na desítkách míst jsem dostal pouze 34 vyplněných dotazníků. Ale i toto množství jsem považoval za dostatečné.

Z dotazníku vyplynulo, že většina uživatelů používá papír a tužku a že většina nepoužívá diktafon. Došel jsem však k závěru, že diktafon lidé nepoužívají s metodou GTD proto, jelikož k tomu nemají potřebné prostředky. V případě, že má člověk aplikace i s metodou GTD v mobilním telefonu, tak pro nahrání hlasu si musí pořídit další aplikaci. To samé platí v případě papíru a tužky. Buď si musí s sebou stále nosit papír a tužku, nebo musí mít další aplikaci pro kreslení poznámek.

V tomto bodě vidím velkou možnost rozvoje metody GTD. Existuje mnoho situací, kdy si člověk nemůže zapsat myšlenku, či úkol jen proto, že jeho mobilní telefon je dotykový, má softwarovou klávesnici. Představme si, že jedeme v autě a něco nás napadne. Nemůžeme vzít telefon a začít si vytukávat myšlenku na softwarové klávesnici. Z toho důvodu bylo mým cílem implementovat možnost kreslení a zaznamenávání zvuku přímo v aplikaci. Tím se velmi odlišuji i od konkurenčních aplikací, které obsahují pouze zadávání pomocí klávesnice.

Dalším jasným indikátorem byla synchronizace. Více jak 75% vyplněných dotazníků odpovědělo na otázku „Synchronizujete data?“, ano. Moje aplikace tedy musí obsahovat synchronizaci. Zde jsem však narazil na řadu problémů, které budou popsány v implementační části.

Informace o tom, kolika kliknutími se uživatel dostane na obrazovku pro přidání nové záležitosti, byla důležitá. Protože rychlost, jakou jsme schopni zadat novou záležitost, je stěžejní. V případě, kdy se pro zadání nové záležitosti musíme proklikat množstvím cest, nás aplikace rychle odradí.

Otázky ohledně počtu kontextů se v dotazníku objevily, abych měl představu kolik položek by se zhruba mělo vejít na jednu obrazovku. Je velmi pohodlné mít všechny kontexty tak, aby je uživatel viděl najednou. Dle výsledků těchto otázek jsem se snažil přizpůsobit velikost a počet informací jednotlivých položek.

Další stěžejní otázkou dotazníku je otázka ohledně datumu a času. Metoda GTD o času jako takovém příliš nemluví. Ale jak vyplynulo z dotazníků, většina lidí datum a čas u svých úkolů prostě potřebuje. Proto jsem se později snažil tuto funkčnost do své aplikace implementovat.

### 4.3 SketchFlow

Než začal vývoj aplikace, pokoušel jsem se vytvořit design uživatelského rozhraní. K tomuto účelu jsem použil Expression Blend 4, který obsahuje nástroje pro návrh uživatelského rozhraní a navigaci mezi stránkami. Pro navrhnutí uživatelského rozhraní pro WP7 je nutno doinstalovat **SketchFlow Template for Windows Phone** [13].

Při navrhování jsem se setkal s problémem. SketchFlow Template for Windows Phone neobsahuje ovládací prvky z **Silverlight for Windows Phone Toolkit** [2]. I přesto, že většina stránek uživatelského rozhraní i navigace mezi nimi byla navržena, nemohl jsem tento koncept použít při vývoji aplikace.

## 4.4 Návrh databáze

Při prvotním navrhování aplikace jsem uvažoval o ukládání veškerých dat do lokálního úložiště. Tento návrh vznikl proto, že WP7 dostalo podporu SQL Compact Edition ve verzi 4.0 až v aktualizaci Mango. Tudíž jsem po vydání aktualizace Mango přešel od XML k SQL. Celé schéma SQL databáze je uvedeno na obrázku 4.3.

### 4.4.1 Lokální SQL databáze

SQL databáze musí mít možnost ukládat data tak, aby je bylo možné snadno zobrazovat a aby umožňovala ukládat všechny důležité položky metody GTD. Každá tabulka databáze obsahuje atribut ID, který je primárním klíčem každé tabulky a je nastaven na autoinkrementaci. Tudíž při správném nastavení relací mezi tabulkami nebude docházet k nekonzistenci dat v databázi a při vkládání nových dat se nemusím starat o duplicitu primárního klíče.

#### Tabulka Contexts

Databáze se tedy skládá ze sedmi tabulek. První z nich je tabulka „Contexts“, která obsahuje především dva atributy, ID a Name. Jak jsem již zmínil, ID je primární klíč tabulky, informace o změně Modified, informace o smazání Deleted a atribut Name obsahuje název kontextu. V této tabulce se ukládají všechny uživatelem vytvořené kontexty.

#### Tabulka Values

Tabulka Values je tabulkou pro uložení hodnot jednotlivých záležitostí. Pokud by hodnota každé záležitosti byla pouze textová, bylo by jednodušší mít pouze atribut hodnoty v tabulce záležitostí. Ale tato aplikace má umožňovat i kreslení prstem a zaznamenávání záležitostí za pomoci hlasu. Proto bylo třeba hodnotu záležitosti oddělit do zvláštní tabulky. Tabulka Values obsahuje opět primární klíč ID, informace o změně Modified, informace o smazání Deleted a pak atributy pro jednotlivé hodnoty. Přičemž je vždy jedna z nich naplněna a ostatní jsou nastaveny na hodnotu NULL.

#### Tabulka Alarms

V další tabulce jsou uloženy informace o vytvořených budících. Budík je vytvářen v systému a lze k němu přistupovat pomocí jména. Tato tabulka byla vytvořena proto, abych měl možnost přistupovat k jednotlivým budíkům. Jejím atributem je primární klíč ID, text budíku AlarmText, datum a čas budíku AlarmDate, datum vytvoření budíku AlarmCreateDate, informace o změně Modified a informace o smazání Deleted.

## Tabulka Inbox

Tabulka Inbox je hlavním prvkem metody GTD. Do ní jsou ukládány všechny záležitosti z části Sběr metody GTD. Skládá se z primárního klíče, datumu vytvoření, informace o změně Modified, informace o smazání Deleted a cizího klíče odkazujícího do tabulky Values.

## Tabulka SomeDay

Tabulka SomeDay v sobě ukládá záležitosti, které nejsou nyní splnitelné, ale uživatel by je někdy v budoucnu, možná vyřídil. Tabulka je složena z 6 atributů. A to z primárního klíče ID, datumu vytvoření CreateDate, poznámky Note, informaci o dokončení záležitosti, cizí klíč odkazující na hodnotu záležitosti, informace o změně Modified, informace o smazání Deleted a cizí klíč ukazující do tabulky Contexts.

## Tabulka Projects

V poslední řadě je zde i tabulka Projects, která ukládá jednotlivé projekty. Pojem projekt je v metodě GTD stěžejní a proto je nutno mít tuto tabulku v databázi. Tabulka Projects obsahuje 7 atributů. Opět primární klíč ID, jméno projektu Name, datum vytvoření projektu CreateDate, cizí klíč odkazující do tabulky Contexts a Alarms, datum a čas dokončení projektu DueDate, informace o změně Modified, informace o smazání Deleted a informaci o dokončení projektu.

## Tabulka Items

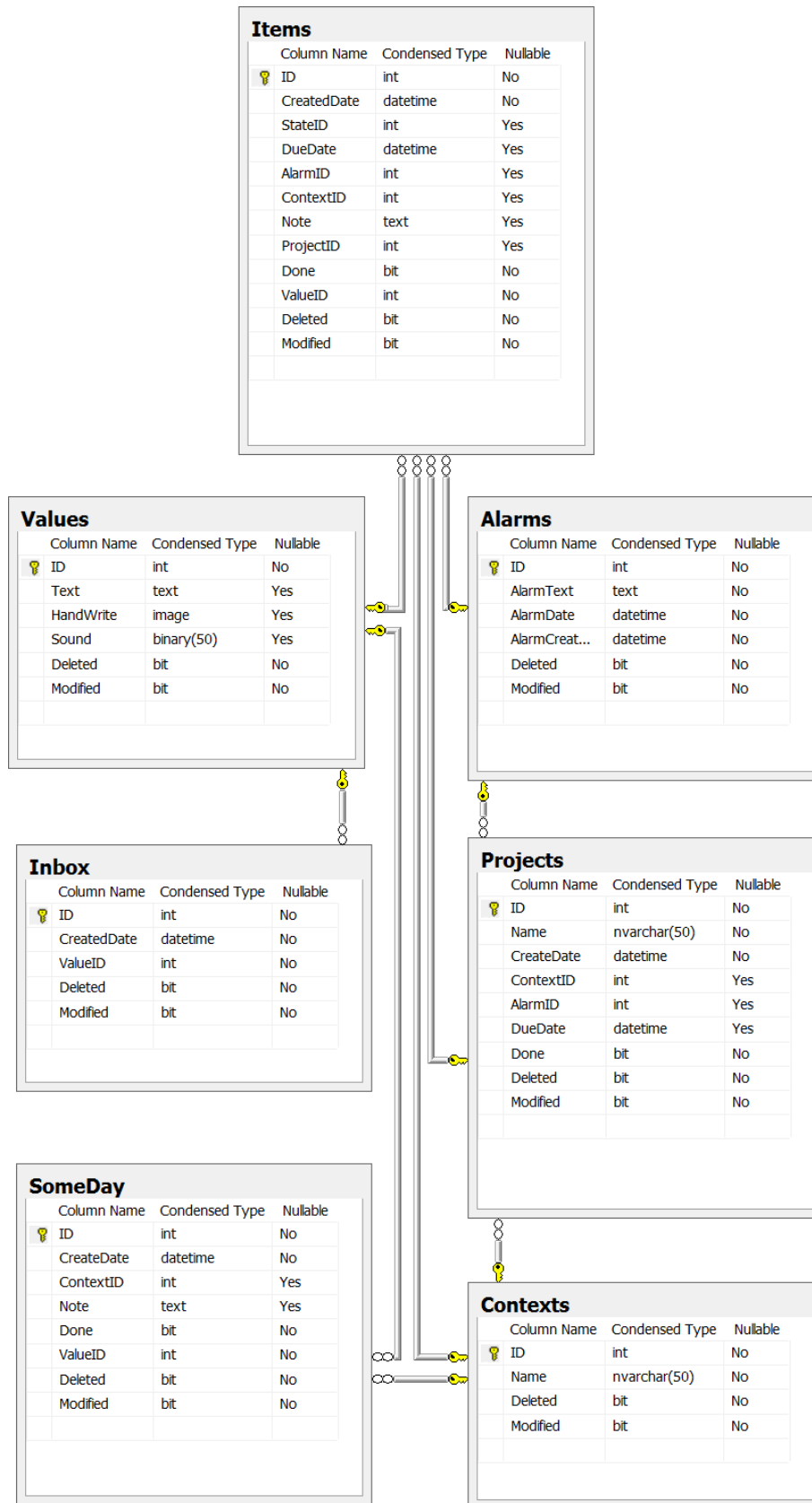
Poslední sedmou tabulkou v databázi je tabulka Items, která ukládá všechny aktuální i dokončené záležitosti. Je složena z 9 atributů, kde prvním je opět primární klíč tabulky. Poté datum vytvoření CreateDate, datum dokončení záležitosti DueDate, poznámka Note, informace o dokončení záležitosti, informace o změně Modified, informace o smazání Deleted a cizí klíče odkazující do tabulek Alarms, Projects, Values a Contexts.

## 4.5 Databáze na serveru

Databáze na serveru, vůči kterému se všichni uživatelé mohou synchronizovat, se od té lokální příliš neliší. Všechny tabulky z lokální databáze jsou samozřejmě i na té serverové. Tabulka Users je na serveru navíc. Tady jsou uloženy přihlašovací údaje všech uživatelů a vůči této tabulce se každá aplikace přihlašuje. Skládá se z primárního klíče ID, uživatelského jména UserName, hesla Password, emailové adresy Email a informaci o povolení uživatele Allowed.

Rozdíl mezi tabulkami v lokální databázi a té serverové je takový, že každá tabulka na serveru obsahuje dva atributy navíc. Jedná se o atribut OriginalID, který ukládá informaci o ID v lokální databázi a o UserID, což je cizí klíč odkazující do tabulky Users a určuje, která data patří kterému uživateli.

Princip synchronizace, registraci nových uživatelů a další operace spojené se synchronizací jsou popsány v následující kapitole.



Obrázek 4.3: Diagram SQL databáze, kde lze vidět vztahy pomocí primárních klíčů tabulek. Obrázek získán z nástroje SQL Management Studio

## 4.6 Použité knihovny

Při vývoji aplikace jsem narazil na některé problémy, které jsem byl schopen vyřešit až po využití externích knihoven. Všechny použité knihovny mají licenci, která je dovoluje využívat bez omezení.

### 4.6.1 Načítání - ProgressBar

Při delším načítání dat z databáze není vhodné, aby uživatelské rozhraní po tuto dobu nereagovalo. Proto veškeré načítání je ve zvláštních vláknech a během načítání je zobrazen progressbar přes celý displej. Bohužel progressbar přes celý displej normálně ve WP7 nelze použít a proto jsem našel a použil již vytvořený. Tento progressbar vytvořil Jeff Wilcox, celý projekt je Open Source a jeho princip a použití je popsáno na jeho webu [7].

### 4.6.2 Přehlednost - Silverlight for Windows Phone Toolkit

V systému WP7 jsou použité ovládací prvky, které bohužel nejsou dostupné pro vývojáře. Příkladem může být seznam kontaktů, neboli People Hub, kde je použit klasický seznam kontaktů, ale navíc jsou zde rozděleny dle abecedy do skupin dle prvního písmene jména. Nad každou skupinou jmen je zobrazeno písmeno dané skupiny z abecedy a po kliknutí na něj se zobrazí všechny skupiny. To nám umožní navigovat se přímo do dané skupiny. Tento ovládací prvek je velmi důležitý pro přehlednost a rychlost navigace [2].

Naštěstí pro vývojáře byla vytvořena knihovna, která obsahuje tento a mnoho dalších uživatelských prvků. Jedná se konkrétně o Silverlight for Windows Phone Toolkit. Tento toolkit podléhá licenci Microsoft Public License (MS-PL), která dovoluje její použití. Z této knihovny jsem použil několik uživatelských prvků, kterými jsem se snažil zpřehlednit celou aplikaci.

### 4.6.3 Zobrazení informací - Coding4Fun Windows Phone Toolkit

Platforma WP7 obsahuje možnost zobrazení informací pomocí Toast. Pokud však chci po uživateli zadat nějaké informace tak, aby se zobrazilo jen Popup okno a ne celá nová stránka, musím využít nalezený toolkit. Jedná se o Coding4Fun Windows Phone Toolkit, který obsahuje mnoho druhů Popup oken, z nichž jsem několik využil. Tento toolkit opět podléhá Microsoft Public License (MS-PL) [9].

### 4.6.4 MessageBox - MessageBoxService

Při návrhu aplikace jsem setkal se zabudovaným messageboxem, který neumožňuje upravovat počet a jména tlačítek. Tedy messagebox s tlačítky „Ano“ a „Ne“ nelze vytvořit. Proto jsem byl nucen opět využít externí možnost a to MessageBoxService vytvořený Derekem Lakinem. Creative Commons Attribution-ShareAlike 3.0 Unported License je licence tohoto projektu a opět mi dovoluje použít projekt v mé aplikaci [6]. MessageBoxService jsem si trochu upravil. Postup úpravy popisují dále v kapitole implementace. Tento projekt mi umožňuje upravovat si messagebox dle své představy, což je přesně to, co jsem potřeboval.

#### 4.6.5 Průhlednost - ImageTools for Silverlight

Tato knihovna byla využita při ukládání kreseb prstem. Jelikož ve WP7 lze přepínat mezi světlým a tmavým pozadím, bylo nutné vždy u každé kresby udělat pozadí průhledné, aby uživatelé mohli mezi těmito režimy přepínat, aniž by měli poté v aplikaci obrázky s opačným pozadím než mají nastavené. V této knihovně existují metody, které umožňují uložit obrázek ve formátu **PNG**, tedy obrázek s průhledným pozadím, což WP7 neobsahuje. Tato knihovna podléhá Microsoft Public License (MS-PL) [\[17\]](#).

## Kapitola 5

# Implementace

### 5.1 Databáze

Jak jsem se již zmínil, ve WP7 se možnost využít SQL datbázi objevila až v aktualizaci Mango. Avšak vytvoření této databáze není možné jednoduše za pomoci nějakého dialogu či příkazu. Je nutné si nejdříve vytvořit třídy, které popisují celou databázi, jednotlivé tabulky a jejich atributy (sloupce), dokonce i relace mezi tabulkami.

#### 5.1.1 Vytvoření databáze

Pro tvoření databáze jsem nejdříve napsal třídy popisující jednotlivé tabulky. Aby klasická třída v jazyku C# mohla popisovat tabulku SQL databáze, musí být tato třída otagována anotací [**Table**]. Po vytvoření anotované třídy jsem mohl tvořit jednotlivé sloupce SQL tabulky. Sloupce se ve třídě nachází v podobě vlastností třídy, které mají opět anotaci [**Column(parameters)**]. V kulaté závorce anotace se nacházejí parametry sloupce, to zda se jedná o primární klíč, zda může být sloupec nulový atd. Rozdíl v anotaci je pouze u sloupce, který je cizím klíčem. U cizího klíče je anotace v podobě [**Association()**], kde se opět v kulatých závorkách nacházejí parametry relace. Parametry mohou být například jméno relace **Name = "FK\_Values\_SomeDay"**, název klíče v cizí tabulce **OtherKey = "Id"** atd. Po vytvoření všech anotovaných vlastností definujících sloupce a cizí klíče SQL tabulky jsem přešel k tvoření relací mezi jednotlivými tabulkami. Relace se tvoří opět jako vlastnost, ale tentokrát se využívá třída **EntitySet<>**, kde se v ostrých tabulkách nachází jméno třídy, která popisuje druhou tabulku relace. Takto se relace musí vytvořit pouze v těch tabulkách, které mají kardinalitu **N** a tudíž neobsahují atribut cizího klíče.

#### 5.1.2 DataContext

Jakmile jsem napsal veškeré třídy, sloupce, anotace a relace, mohl jsem vytvořit poslední třídu, která zastřešuje přístup k SQL databázi. Tato třída dědí ze třídy **DataContext**, což je třída, která umožňuje přístup LINQ to SQL. Jedná se o přístup, kdy za pomoci jazyka LINQ pracujeme s SQL databází. Výhodou je především to, že nemusíme psát složité a dlouhé řetězce představující SQL dotazy. Můžeme využívat jednoduchý jazyk LINQ, který nám nabízí navíc intellisense. V této třídě se nachází vlastnosti, díky kterým je umožněn přístup ke všem tabulkám, změnám v nich, přidávání či mazání a ukládání všech změn do fyzické databáze.

Celý princip je takový, že změny prováděné za běhu aplikace se ukládají právě v Data-Contexu a ve fyzické databázi a žádná změna se neprojeví do zavolání metody **SubmitChanges()**. Po jejím zavolání se všechny provedené změny, ať už se jedná o přidání řádku, smazání řádku, či úpravu řádku, projeví ve fyzické SQL databázi.

### 5.1.3 Model-View databáze

Platforma WP7 přináší možnost rozsáhlého **binding**, což je vázání dat přímo do uživatelského rozhraní s minimem kódu. Této výhody jsem se rozhodl využít jednak proto, že činí aplikaci přehlednější. A také proto, že databinding je maximálně optimalizován a tudíž je jeho výkonost mnohem větší než jakýkoli mnou napsaný kód.

Databinding lze provádět mnoha způsoby. Ale v mém případě bylo třeba navázat data ze SQL databáze do uživatelského rozhraní. Proto bylo nutné vytvořit nejdříve třídy, které zaobalují jednotlivá data ze SQL databáze. Aby tyto třídy správně vázaly a aktualizovaly data v uživatelském rozhraní, je nutné, aby každá třída dědila ze třídy **INotifyPropertyChanged**. Tato dědičnost si vynucuje implementaci metody **NotifyPropertyChanged** a události **PropertyChangedEventHandler**. Díky této třídě jsem byl schopen jednoduše informovat uživatelské rozhraní o změně v datech a zobrazit aktuální.

Každá takto vytvořená třída má jméno končící „Object“ a každá je specifická, jelikož jsem od každé vyžadoval trochu odlišné chování. Všechny však obsahují právě jednu položku (řádek), kterou zaobalují. K této položce přináší další vlastnosti a metody, které určují právě chování a možnosti dané třídy. Například třída **ItemObject** obsahuje vlastnost **Item**, ve které je uložen právě jeden konkrétní řádek z tabulky **Items**. Dále využívá vlastnost **DayCount**, která dopočítává stáří položky. Atributy **IsItemHandWrite**, **IsItemText** a **IsItemSound** určují, jestli položka obsahuje text, kresbu nebo nahrávku a podle toho se zobrazuje ten ovládací prvek uživatelského rozhraní, který odpovídá dané hodnotě. Pro zobrazení textu se zviditelní **TextBox**. Pro nahrávku se zobrazí hned několik uživatelských prvků a to tlačítka a progressbar zobrazující průběh přehrávání. A pro kresbu se zobrazí **Image**. Právě díky databindingu se toto děje automaticky, aniž bych musel někde v kódu stránky toto upravovat manuálně. Některé třídy obsahují vlastnost, která v sobě ukládá seznam položek a ty jsou k položce třídy nějak vázány. Příkladem může být seznam všech záležitostí z tabulky **Items**, které jsou přiřazeny k určitému kontextu. Tudíž třída **ContextObject**, která zaobaluje jeden kontext bude mít ve vlastnosti **ContextItems** všechny záležitosti z tabulky **Items**, které patří k danému kontextu.

Takto vytvořené třídy nad tabulkami SQL databáze postačí při binding do jednoduchých ovládacích prvků jako je **TextBox**, **Image** apod. Ale ve chvíli, kdy potřebuji zobrazit celý seznam těchto položek, mám dvě možnosti. První možností je napsat si kód na každé stránce, který mi bude naplňovat seznam, v mém případě **ListBox** nebo jiný ovládací prvek podobný **ListBoxu** anebo si opět vytvořím třídu s touto funkcí. Já jsem zvolil vytvoření třídy implementující tuto funkčnost a to především proto, že složitější ovládací prvky jako je **LongListSelector** [2] se naplňují docela složitým způsobem a kdybych měl takový kód na každé stránce, bylo by to nepřehledné.

Třída, která by naplňovala klasický **ListBox**, nebo **ListPicker**, musí jednak dědit ze třídy **INotifyPropertyChanged**, ale také ze třídy **ObservableCollection<>**, kde v ostrých závorkách se nachází právě třída zaobalující jeden řádek tabulky, například **ContextObject**. **ObservableCollection** je třída, která umožňuje pracovat s kolekcí objektů a sama automaticky informuje uživatelské rozhraní o přidání, smazání či aktualizaci jednotlivých objektů.

Takto vytvořená třída má v konstruktoru kód, který po tvoření nové instance třídy, ji



automaticky naplní. Například třída **ListOfContexts** se po vytvoření její instance automaticky naplní všemi kontexty v SQL databázi a k tomu na první pozici přidá defaultní položku „—“. Potom tuto třídu můžu jednoduše nabídnout do ListPickeru, aniž bych musel psát další kód ve stránce.

Byl jsem však nucen vytvořit další třídy, které jsou rozšířením předchozích. Pro naplnění ovládacího prvku **LongListSelector** nestačí pouze seznam záležitostí, ale všechny tyto položky musí být rozděleny dle určitých kategorií. Abych tedy mohl naplnit takový ovládací prvek, musel jsem nejdříve vytvořit třídy, které budou uchovávat jednotlivé kategorie prvků. Příkladem je třída **ContextsInGroup**, která obsahuje opět seznam záležitostí a současně obsahuje klíčovou vlastnost **Key**, která určuje danou kategorii. V tomto případě písmeno abecedy a třída uchovává všechny kontexty, které na vybrané písmeno abecedy začíná. Druhou částí pro naplnění LongListSelector je vytvoření třídy, která bude kolekcí právě všech těchto kategorií. Jedná se o kolekci písmen abecedy. Opět se třída automaticky naplní po vytvoření její instance. Takovou třídu mohu naplnit i tak složitým ovládacím prvkem jako je LongListSelector.

#### 5.1.4 Binding konvertory

Binding konkrétních dat do ovládacích prvků nám platforma WP7 nabízí tzv. konvertory. Jsou to třídy, které dědí ze třídy **IValueConverter** a obsahují metody **Convert** a **ConvertBack**. Tyto metody nám umožňují převést hodnotu, která se váže do ovládacího prvku na jinou hodnotu, a to ještě před tím, než dojde k vlastnímu binding. Já jsem tyto konvertory využil a docela hojně. Umožnilo mi to využívat binding i na místech, kde k tomu daná třída nebyla úplně přizpůsobena a já nemusel vytvářet další třídu. Příkladem může být konvertor **BoolToTextConverter**, který mi převádí boolovské hodnoty „True“ a „False“ na jejich textovou variantu, která je automaticky překládána dle nastavené jazykové mutace systému.

## 5.2 Uživatelské rozhraní

Při navrhování uživatelského rozhraní jsem se řídil pravidly pro tvoření **Metro** rozhraní. Cílem byla rychlá odezva a především přehlednost záležitostí na jednotlivých stránkách. V návrhu jsem nevyužil výše zmiňovaný **SketchFlow**, jelikož neobsahoval ovládací prvky z mnou použitých knihoven.

### 5.2.1 Hlavní stránka

Hlavní stránka se však nezměnila. Byla navrhována tak, abych se lišil od konkurence, která ve většině případů využívá podobu dlaždic či panorama control. Hlavní stránka má podobu úlu, tedy šestiúhelníků vedle sebe. Jen některé šestiúhelníky jsou aktivní a tvoří specifický útvar.

Z hlavní obrazovky lze zobrazit Inbox, záležitosti dle kontextů, záležitosti dle projektů, nadcházející záležitosti, záležitosti Někdy/Možná a otevřít nastavení aplikace.

Dále hlavní stránka obsahuje **AppIactionBar**, který je vždy umístěn ve spodní části stránky. V tomto panelu se nacházejí 3 tlačítka. Prvním z nich je tlačítko **Nahrávka**, které naviguje na stránku pro přidání nové záležitosti ve formátu zvukového záznamu. Dalším tlačítkem je **Text**, které naviguje na stránku pro přidání nové záležitosti v textové podobě. A posledním je tlačítko pro přidání nové záležitosti jako kresbu prstem **Kresba**. Tato

tlačítka jsou zde umístěna z prostého důvodu. Jedním z cílů aplikace bylo, aby se uživatel dostal na stránku pro přidávání co nejrychleji. Proto jsem umístil tato tlačítka hned na hlavní stránku. Jsou umístěny na viditelném pozici, ale nepřekáží.

Při stisku šestiúhelníků **Inbox**, **Kontexty**, **Projekty**, **Další akce** a **Někdy/Možná** je nutné před vlastní navigací na další stránku zkontrolovat, zda je vůbec co zobrazit. Pokud aplikace zjistí, že po přechodu na další stránku se nic nezobrazí, vypíše pro uživatele informaci a zůstane na hlavní stránce. Příkladem je **Inbox**. Pokud uživatel ještě nevytvořil žádnou záležitost, která by spadala do složky **Inbox** a chtěl by kliknout na tlačítko **Inbox**, aplikace mu zobrazí: „GTD aplikace nemá žádné záležitosti ve složce **Inbox**“.

Hlavní stránka je místo, na které se navigují všechny vytvořené dlaždice. Při navigaci na stránku se však zkoumá **QueryString**. Pokud obsahuje parametr **Navigate**, dochází k navigaci na stránku, která je uvedena v následujícím parametru **Page**. Tato funkčnost se vyvolá stiskem dlaždice pro přidání nahrávky, kresby a textu. Hlavní stránka v tomto případě ihned naviguje na příslušnou stránku pro přidání nové záležitosti odpovídajícího typu.

Do hlavní stránky bylo také přidáno převrácení obrazovky. Obracení obrazu dle natočení zařízení je implementováno defaultně, avšak bez jakékoli animace. Při otočení zařízení horizontálně se šestiúhelníky nevešly na displej. Proto jsem přidal animaci pro převrácení obrazu a 2 stavy do xaml souboru stránky, kde nastavuji přesné umístění každého šestiúhelníku a kde každý stav představuje jeden způsob otočení zařízení.

### 5.2.2 Přidávání nové nahrávky

Stránka pro přidání nové nahrávky se jmenuje **AddSoundPage**. Hlavní a jedinou funkcí je nahrání hlasu uživatele. Toho je docíleno uložením binárního souboru do úložiště aplikace a uložením cesty k souboru do SQL databáze.

Při nahrávání je uživateli zobrazen **ProgressBar** s časem, který ukazuje kolik zbývá do konce nahrávání. Je to proto, že maximální délka nahrávky je 1 minuta. K omezení maximální délky nahrávky jsem přistoupil z důvodu rychlé manipulace s binárními soubory a plynulým přehráváním.

Stránka obsahuje opět **AppBar**, na kterém jsou 3 tlačítka. Tlačítko **start** spustí nahrávání záznamu, tlačítko **stop** zastaví nahrávání a uloží výsledek a tlačítko **Pin** vytvoří sekundární dlaždici, která umožňuje navigaci přímo do nahrávání záznamu. Pokud je již dlaždice vytvořena, tlačítko **Pin** bude nepřístupné.

Při navigaci na stránku se dle nastavení buď spustí okamžitě nahrávání, nebo se pouze zobrazí stránka. Při spuštění nahrávání se rozpožbuje **ProgressBar** a také kolečko uprostřed stránky. Podle toho, jak hlasitě uživatel mluví do mikrofону, se mění velikost kolečka. Je to pro informaci, aby uživatel viděl na první pohled, že opravdu něco nahrává.

Nahrávání je ukončeno po stisku tlačítka **stop** anebo po vypršení 1 minuty. Jakmile se ukončí nahrávání, je soubor okamžitě uložen a je automaticky vytvořena záležitost ve složce **Inbox**. Tato částečná automatizace je proto, že jsem předpokládal, že hlasový záznam bude uživatel používat ve chvílích kdy nemá možnost, nebo chuť použít klávesnici a proto nemůže vybrat další možnosti jako kontext, projekt nebo datum dokončení. Z tohoto důvodu se nahrávka automaticky ukládá do složky **Inbox**.

### 5.2.3 Přidávání nové kresby

Tato stránka má umožnit uživateli nakreslit si pomocí prstu potřebnou záležitost. Kreslení probíhá zachycením dotyku prstů a vykreslením v **InkPresenter**. Jsou zde implemen-

továny 3 metody. První metoda `inkTest_MouseLeftButtonDown` odchyťává událost dotyku prstu displeje. V této metodě se vytvoří nový **Stroke**, což je nová čára na InkPresenter. Další metoda `inkTest_MouseMove` odchyťává pohyb prstu po displeji, což do Stroke přidává body, přes které má čára procházet a InkPresenter se postará o vykreslení. Poslední metodou je `inkTest_MouseLeftButtonUp`, která odchyťává událost opuštění prstu displeje.

Stránka obsahuje také AppBar, který má 4 tlačítka. Tlačítko pro zrušení a navrácení zpět na hlavní stránku, tlačítko pro vymazání obsahu, tlačítko pro vytvoření sekundární dlaždice a tlačítko pro uložení.

Vyřešit ukládání bylo náročnější, než se na první pohled zdálo. Musel jsem vyřešit problém s pozadím kresby. Při ukládání docházelo k tomu, že uživatel, který uložil kresbu se světlým nastavením systému (bílé pozadí) a později přepnul na tmavé (černé pozadí), měl kresbu s bílým pozadím a ostatní bylo černé. To nevypadalo hezky. Proto při ukládání je vytvořeno nové vlákno, které na pozadí prochází kresbu bod po bodu a pokud narazí na bod, který odpovídá barvě pozadí, nastaví ji na průhlednou.

```
bitmap.SetPixel(x,  
                y,  
                Colors.Transparent.A,  
                Colors.Transparent.R,  
                Colors.Transparent.G,  
                Colors.Transparent.B);
```

Po nastavení pozadí na průhledné se teprve provede uložení kresby do úložiště a uložení cesty do SQL databáze. Stejně jako při ukládání nahrávky i zde se automaticky vytvoří nová záležitost ve složce Inbox. Je to ze stejného důvodu jako u nahrávky.

#### 5.2.4 Přidávání nové záležitosti

Tato stránka představuje klasické přidávání nové záležitosti v textové podobě. Pokud uživatel zadává záležitost na klávesnici, má mnohem víc možností než u použití nahrávky či kresby. Uživatel si může vybrat, zda chce vytvořit nový kontext, projekt, budík, nastavit datum a čas dokončení anebo označit záležitost jako „Někdy/Možná“.

Stejně jako všechny ostatní stránky pro přidávání záležitostí i tato obsahuje AppBar, jehož součástí je tlačítko pro uložení, zrušení a připíchnutí sekundární dlaždice.

Pokud uživatel zadá pouze text záležitosti a uloží ji, záležitost automaticky putuje opět do složky Inbox. V případě, že zadá kontext nebo projekt, záležitost se ukládá jako již zpracovaná a roztříděná do SQL tabulky Items. Po vybrání kontextu nebo projektu se také zobrazí další možnosti vyplnění a to datum a čas dokončení a přidání budíku pro upozornění na danou záležitost. Na stránce je také umístěn přepínač, který umožňuje uložit záležitost do složky Někdy/Možná. Pokud tento přepínač přepne uživatel do sepnuté polohy a byl předtím vybrán nějaký projekt, tak se projekt záležitosti smaže.

Při volbě budíku má uživatel možnost vybrat si z relativních budíků, které se vztahují k datumu a času dokončení. Pokud datum nebo čas dokončení nejsou zadány, je uživatel při vybrání relativního budíku o tomto problému informován. Druhá možnost zadání budíku je nastavení přesného datumu a času. Při ukládání záležitosti s budíkem se kontroluje, zdali uživatel nezadá datum a čas budíku do minulosti. Pokud ano, je o tom opět informován.

Jak jsem se již zmínil, uživatel má možnost nejenom si vybrat kontext nebo projekt, ale i vytvořit nový kontext nebo projekt. Toto se provádí kliknutím na tlačítko „plus“, které

se nachází vedle výběru kontextu a projektu. Jelikož vytváření nového kontextu, projektu a konkrétního budíku je na jiné stránce, bude toto popsáno později.

### 5.2.5 Přidávání projektů, kontextů a budíků

Pro přidání nového kontextu, projektu a budíku jsem vytvořil jednu jedinou stránku, která obsahuje již zmíněný **Pivot Control**. Výhoda tohoto způsobu je taková, že když se uživatel překlikne, stále se může pouhým posunutím v Pivot Control dostat tam kam původně chtěl. Například, když uživatel chce kliknout na přidat projekt, ale klikne na tlačítko „Plus“ pro přidání kontextu, otevře se ta samá stránka, ale na jiné záložce. Uživatel se pouhým jedním gestem dostane tam kam chce a může pokračovat, aniž by se musel nějak vracet.

Tento ovládací prvek má 3 části. První částí je přidání nového kontextu, kde lze zadat pouze název kontextu. Při ukládání dochází ke kontrole, zdali již zadaný kontext neexistuje. Pokud existuje, je o tom uživatel informován. Pokud neexistuje je uložen do SQL databáze. Další část je přidání projektu. Zde uživatel může zadat nejen název projektu, ale i k jakému kontextu daný projekt patří, může přidat datum a čas dokončení projektu, případně přidat budík k tomuto projektu. Pokud uživatel chce k projektu přidat jednu z vypsanych možností, stránka se naviguje sama na sebe s patřičnými parametry v **QueryString**. Otevře se stejná stránka, ale v tomto případě už nemá uživatel možnost vybrat si co chce přidat. V případě že uživatel v tuto chvíli přejde na jinou část Pivot Control, tlačítko pro uložení se zablokuje a povolí se až uživatel gestem najede zpět. Po uložení se stránka vrací zpět na přidání nového projektu a automaticky doplní vše co bylo vyplněno před odchodem a přidá co bylo přidáno. Při ukládání projektu opět dochází ke kontrole duplicity projektu a v případě nalezení stejného projektu je uživatel informován. Poslední 3 částí je přidání budíku. Zde uživatel výběrem z **DatePicker** a z **TimePicker**, což jsou uživatelské prvky z **Silverlight for Windows Phone Toolkit** [2], vybere konkrétní datum a čas nového budíku. I zde se při ukládání provádí kontrola. Avšak ne na duplicitu, ale zdali uživatel nezadal datum a čas do minulosti. Vytvářet budík, který již měl zvonit, je zcela zbytečné a především WP7 toto ani nedovoluje a vyvolává výjimku **InvalidOperationException**.

Stránka obsahuje dvě tlačítka pro uložení a zrušení, která jsou opět umístěna v **ApplicationBar**.

### 5.2.6 Zobrazení kontextů

Do této části aplikace se uživatel dostane z hlavní stránky po kliknutí na šestiúhelník se jménem kontexty. Na této stránce je **LongListSelector**, který je popsán v 3. kapitole. Skupiny jsem vytvořil jako písmena abecedy. Tedy zobrazují se zde všechny kontexty rozdělené do skupin dle počátečních písmen kontextů. Po kliknutí na jednu ze skupin se zobrazí všechny skupiny (písmena abecedy), kde jsou neaktivní a zašedlá, to znamená ty skupiny, které nemají žádný kontext. Po kliknutí na aktivní písmeno se přejde ihned na skupinu kontextů, které začínají vybraným písmenem.

Každý jeden řádek má 2 stavy. První stav je takový, kdy kontext neobsahuje žádné záležitosti a tehdy se zobrazí pouze kontext. Druhý stav je opačný, tedy kontext obsahuje minimálně 1 záležitost a vtom případě se zobrazí název kontextu a pod ním počet záležitostí, které obsahuje a datum poslední přidané záležitosti. Po kliknutí na kontext v tomto stavu dojde k rozbalení kontextu a zobrazí se seznam všech záležitostí s daným kontextem. Tato funkčnost je implementována za pomoci uživatelského prvku jménem **ExpanderView**, který je také součástí **Silverlight for Windows Phone Toolkit** a byl popsán v 3. kapitole.

Na této stránce byl použit ještě jeden prvek z předešlého Toolkitu a to **ContextMenuService**. Tento uživatelský prvek umožňuje zobrazit kontextové menu po přidržení prstu na jednom místě. Tuto funkčnost jsem se rozhodl využít jako možnost upravit vybraný kontext. Pokud uživatel chce upravit kontext, podrží prst na vybraném kontextu a z nabídky vybere editovat. Otevře se stránka pro přidání kontextu, kde se předvyplní jeho název a uživatel ho může upravovat.

### 5.2.7 Zobrazení projektů

Zobrazení projektů je totožné se zobrazením kontextů. Rozdíl je v použitých třídách, které se používají při **Binding**. Je tu i totožná funkčnost. I přesto, že tyto dvě stránky mají stejnou funkci a prvky, potřebujeme obě dvě, aby měl uživatel možnost zobrazit si záležitosti dle kontextů i dle projektů.

Další rozdíl je v tom, že na stránce s projekty je možnost zobrazit si záležitosti, které již byly označeny jako dokončené. Také projekt lze označit jako dokončený, čímž se dokončí všechny jeho záležitosti. To proto, aby si mohl uživatel projít celý projekt, podívat se co již má hotovo a co ho ještě čeká.

### 5.2.8 Zobrazení složky Inbox

Na této stránce se zobrazují všechny záležitosti, které uživatel během dne sesbírá. Je to hlavní sběrné místo, které metoda GTD považuje za klíčové. Jednotlivé položky jsou seřazeny v **MultiselectList** pod sebou. Tento uživatelský prvek má tu funkci, že po kliknutí na levou část jedné ze záležitostí, objeví se u každé **Checkbox** a uživatel může vybrat více záležitostí. Tyto záležitosti se dají smazat všechny naráz.

Při procházení složky Inbox stačí, aby uživatel kliknul na danou záležitost a otevře se stránka pro přidání nové záležitosti a uživatel má možnost přidat další věci jako kontext, projekt atd.

V **MultiselectList** se zobrazují všechny druhy záležitostí. Nahrávka, kresba i text. Toto jsem vyřešil za pomoci **Binding**. Ve třídě, která naplňuje **MultiselectList** existují vlastnosti, které určují viditelnost prvků pro nahrávku, kresbu i text. Pouhým nastavením **Binding** jsem dosáhl automatického zobrazení a naplnění těch správných ovládacích prvků.

Také zde jsem použil **ContextMenuService**, který zobrazuje nabídku s tlačítky pro smazání a převod. Převodem je myšleno přepsání kresby nebo nahrávky na textovou verzi záležitosti. Tuto funkčnost jsem implementoval především proto, že mnoho uživatelů si dle odpovědí v dotazníku své „kresby“ přepisuje. Je také pravda, že když člověk vidí text, hned je mu jasné, co to je. Kdežto nahrávku si musí přehrát.

**AppBar** této stránky obsahuje i menu, kde může uživatel seřadit záležitosti dle data, dle textu záležitosti, pokud nějaký má, a nebo seřadit dle druhů záležitostí.

### 5.2.9 Zobrazení následujících úkolů

Tento seznam primárně slouží k zobrazení těch záležitostí, které mají nastaveno datum dokončení. Ty se zobrazují dle data a u každé z nich je zobrazeno staří záležitosti, datum dokončení a možnost označit záležitost jako dokončenou. U každé záležitosti se podle data dokončení zobrazuje zelený, žlutý a červený animovaný vykřičník. Barva vykřičníku závisí na tom, kolik času zbývá do vypršení data záležitosti. Pokud již datum záležitosti vypršelo nebo odpovídá aktuálnímu datu, je zobrazen červeně, pokud je rozdíl mezi daty do 3 dnů je žlutý, pokud je rozdíl mezi daty do 7 dnů je zobrazen zeleně a pokud je do data dokončení

více než 7 dní není zobrazen žádný. Je to způsob informování uživatele tak, aby na první pohled viděl aktuálnost dané záležitosti a nemusel dlouho zkoumat a porovnávat data v paměti.

Snadno se může stát, že uživatel při pohybu seznamem může omylem zatrhnout záležitost jako dokončenou. Pokud by se tato označila jako dokončená a hned zmizela, mohlo by to uživatele zmást. Proto v **ApplicationBar** této stránky je tlačítko, které se při zatržení záležitosti jako dokončené povolí a až po kliknutí na toto tlačítko se změny uloží. Stránka má také menu, kde může uživatel zobrazit záležitosti dle kontextu, vybrat záležitosti, které nemají datum dokončení a nebo zobrazit normální seznam záležitostí s datem dokončení.

### 5.2.10 Zobrazení složky Někdy/Možná

Ve složce Někdy/Možná si uživatel ukládá záležitosti, které by chtěl v budoucnu splnit, když budou vhodné podmínky. Implementoval jsem tuto složku podobně jako složku Inbox. Za pomoci **MultiselectList** jsou zobrazeny všechny záležitosti, který opět nabízí možnost vybrat si je anebo smazat. Dále lze záležitosti seřadit dle data vytvoření, dle textu záležitosti, pokud existuje, nebo seřadit dle druhů záležitostí.

Po kliknutí na jednu ze záležitostí se zobrazí stránka pro přidání nové a ta se předvyplní vybranou záležitostí. Poté může uživatel roztřídit tuto záležitost dále, přidat do projektu, zadat datum a čas dokončení atd.

### 5.2.11 Náhled a úprava záležitosti

Aby uživatel mohl prohlížet záležitosti jako celek a případně je upravovat, vytvořil jsem tuto stránku. Na ni uživatel vidí všechny informace o dané položce, vše co bylo zadáno při vytváření a může položku označit jako dokončenou i zde.

V **ApplicationBar** stránky jsou dvě tlačítka. První přechází do módu editace záležitosti a druhé záležitost maže, přičemž je uživatel znovu dotázán, zdali chce záležitost opravdu smazat.

V módu editace záležitostí se klasické texty změny na jejich protějšky pro editaci a tlačítko editace se změny na tlačítko pro uložení změn. V tento moment může uživatel změnit kontext, smazat projekt, přidat datum dokončení nebo změnit budík.

### 5.2.12 Přepisování nahrávek a kreseb

Jak jsem již popsal, jedná se o stránku, která nabízí možnost přepsání nahrávky a kresby do textové podoby. Stránka obsahuje pouze ovládací prvky pro přehrání nahrávky a zobrazení kresby a pod těmito ovládacími prvky se vždy nachází **TextBox**. Do tohoto pole uživatel přepíše záležitost a potvrdí změnu.

V **ApplicationBar** stránky jsou pouze dvě tlačítka a to tlačítko pro uložení a zrušení. Při ukládání dochází ke kontrole, zdali uživatel něco zadal. Pokud je **TextBox** prázdný, je o tom uživatel informován.

### 5.2.13 Nastavení

Poslední stránkou celé aplikace je stránka s nastavením. Zde se nachází **PanoramaControl**, který obsahuje 3 části.

První částí je obecné nastavení. Zde si uživatel může vybrat, zdali chce být dotazován o přidávání další záležitosti. Tato otázka je uživateli položena ve chvíli, kdy záležitost uloží.

Objeví se **MessageBoxService**, kde se aplikace uživatele zeptá, zdali chce pokračovat v přidávání další záležitosti. Pokud ano, má uživatel na výběr 3 druhy záležitostí. Nahrávku, kresbu a text. V obecním nastavení má uživatel možnost tuto funkci zapnout a vypnout. Pokud uživatel nechce být dotazován, ale chce automaticky ihned přidat vybraný druh záležitosti, vybere si požadovaný druh z **ListPicker**. Nemusí si také vybrat nic a také se nic nebude dělat automaticky. Další položkou v obecném nastavení je možnost zapnutí automatického startu nahrávání po přechodu na stránku nahrávání.

Další částí nastavení je zobrazení dlaždic, které si uživatel umístí na hlavní obrazovku. Zde o nich má přehled a může je po kliknutí na ně i smazat, aniž by je musel hledat na hlavní obrazovce.

Poslední částí je synchronizace, kde má uživatel možnost se registrovat, synchronizovat svá data a případně uvést systém do stavu, kdy je zcela čistý.

### 5.3 Jazyková lokalizace

Aplikace má dvě jazykové mutace. Češtinu a angličtinu. Lokalizaci aplikace jsem vytvořil za pomoci **Resources**, ve kterých je každý text přeložen a má svůj specifický název pro použití. Každá aplikace musí obsahovat jeden hlavní Resource, který je vybrán v případě, že jazyk systému není podporován. V mé aplikaci jsem jako hlavní Resource použil anglický překlad. Tedy kdekoli na světě kromě České Republiky bude mít uživatel aplikaci v anglickém jazyce. Pro češtinu jsem vytvořil zvláštní Resource, který se jmenuje stejně jako ten hlavní, ale obsahuje tečku a za ní kód jazyku. **Resource.cs** je Resource českého jazyka. WP7 automaticky mezi těmito soubory přepíná dle jazykového nastavení celého systému.

### 5.4 Synchronizace

Synchronizace vznikla proto, aby si uživatelé mohli svoje data v případě ztráty telefonu, či zakoupení nového, tato jednoduše obnovit. Tím se také zvyšuje důvěra v systém GTD, jelikož uživatel ví, že se mu data neztratí.

Při implementaci synchronizace jsem se setkal s velkým problémem. WP7 neumožňuje přímé připojení na vzdálený SQL server. Tím pádem jsem byl nucen hledat alternativu, jak data posílat na vzdálený server. Jako řešení jsem objevil **webovou službu (Web Service)**. Díky ní jsem schopen z WP7 asynchronně komunikovat, i když implementace synchronizace pomocí této služby je mnohem náročnější. Jedním z problémů byla asynchronní komunikace. Je třeba na některé úkony čekat, než se dokončí a k tomu jsem využil třídu **AutoResetEvent**. Tato třída umožňuje zadržet vlákno dokud se **AutoResetEvent** nenastaví na hodnotu **True** metodou **Set**. Synchronizaci jsem vložil do vlastního vlákna a vždy, kdy bylo třeba počkat na dokončení asynchronní operace webové služby, zavolal jsem metodu **WaitOne**. Asynchronní operace webové služby obsahuje vždy 2 metody. První je vlastní spuštění operace a druhá oznamuje její dokončení a obsahuje výsledek operace. Právě v metodě, která informuje o dokončení asynchronní operace, vždy volám metodu **Set**, která uvolní jádro a pokračuje se dále.

Webová služba byla umístěna na webu **ASPone.cz**. U tohoto poskytovatele jsem si zřídil **WebHosting** zdarma a na něm vytvořil webovou službu. Zde jsem také vytvořil serverovou část SQL databáze, která je popsána dříve.

Aby mohlo více uživatelů synchronizovat svá data, obsahuje každý řádek na serveru informaci, kterému uživateli patří. Proto se musí každý uživatel registrovat. Registrace

probíhá zadáním uživatelského jména, hesla a emailu. Po provedení registrace je uživateli zaslán email, ve kterém se nachází odkaz na potvrzení registrace. Tento systém zabráňuje registraci uživatelů, kteří zadají nesprávný email, či chtějí registrovat cizí osobu. Po aktivaci účtu je uživateli povolena synchronizace.

Webová služba pro každou z tabulek obsahuje několik metod, za pomoci kterých aplikace synchronizuje data. Princip synchronizace je takový, že nejprve musíme zaslat ze zařízení čísla **ID** na webovou službu a ta vrátí zpět ty ID, které ještě nebyly synchronizovány. Aplikace poté zašle ty řádky, které na serveru chybí. Po nahrání nových řádků, dochází k aktualizaci změněných řádků. K tomu slouží atribut **Modified** oznamující změnu řádku. Pomocí webové služby se zašlou všechny řádky s atributem **Modified** nastaveným na hodnotu **True** a webová služba tyto řádky aktualizuje. Nakonec dochází k promazání. Na webovou službu se zašlou čísla ID těch řádků, které mají nastaven atribut **Deleted** na hodnotu **True**. Webová služba tyto řádky smaže na serveru a poté se teprve smažou z lokální SQL databáze.

Uživatel má možnost vymazat veškerá svá data a to jak v zařízení, tak na serveru. V nastavení jsem pro tuto situaci umístil tlačítko. Po kliknutí na něj se aplikace zeptá, zdali opravdu uživatel chce vymazat svá data a pokud tento odpoví ano, veškerá data se vymažou.

## 5.5 Testování

Implementace probíhala z větší části na emulátoru. Postupem času jsem dospěl k závěru, že by bylo vhodné otestovat aplikaci na reálném zařízení. Proto jsem si zapůjčil telefon **LG LG-C900** s Windows Phone 7. Telefon byl zapůjčen z Jihomoravského inovačního centra na 14 dní. Na něm jsem testoval vzhled aplikace, její chování apod. Po opravení chyb z tohoto testování jsem si zapůjčil další reálné zařízení. Tentokrát z Fakulty informačních technologií VUT a to **HTC HD7**. U něho jsem narazil na problém při registraci zařízení pro vývoj. Z toho důvodu jsem byl nucen vyměnit zařízení za **HTC Mozart**, kde se registrace povedla napoprvé. Zde jsem testoval především rychlost odezvy, která by po implementaci vláken pro načítání měla být větší. Synchronizace byla z větší části testována také na reálném zařízení a to přes 3G síť.

Na HTC Mozart byla aplikace testována po dobu 14 dní v reálném prostředí a na reálných záležitostech.



## Kapitola 6

# Závěr

Cílem této práce bylo vytvořit aplikaci pro správu času a úkolů, implementující metodu GTD od Davida Allena, avšak s většími možnostmi zadávání záležitostí. Při návrhu programu jsme se snažil všechny zadané cíle splnit. Výsledná aplikace splňuje vše, co jsme si na začátku práce stanovili.

Aplikace se jmenuje „iGTD“. Název je zkratkou „Interactive GTD“, protože nabízí větší možnosti zadávání. Výsledná aplikace umožňuje sbírání záležitostí, jejich třídění a organizování do složek a projektů dle metody GTD. Zadávání jednotlivých záležitostí probíhá pomocí klávesnice, hlasu a kresby prstem, což usnadňuje každodenní používání v různých situacích.

Rozšíření a budoucnost aplikace vidím v možnosti využít živých dlaždic a služeb Microsoftu pro automatickou aktualizaci těchto dlaždic, pro zobrazování následujících záležitostí a projektů. I když v této práci jsou dlaždice zahrnuty, jsou využity jako rychlý přístup pro přidání nové záležitosti. Existuje také možnost propojení s kalendářem WP7. Případný sběr záležitostí z emailové schránky či sms zpráv by mohl být dalším užitečným pomocníkem při sběru dat.

V budoucnu plánuji ve vývoji pokračovat a především reagovat na podněty uživatelů, které tuto aplikaci budou používat. Aplikace iGTD je v době psaní této práce vystavena na Marketplace pro všechny dostupné trhy. iGTD má přes 40 stáhnutí a dokazuje svoji kvalitu reakcemi na Marketplace i emaily. Jeden je uveden v příloze C.

# Literatura

- [1] AddictiveTips: 10 Awesome New Features In Windows Phone 7.5 Mango [online]. 2011 [cit. 26.4.2012]. <http://www.addictivetips.com/mobile/10-awesome-new-features-in-windows-phone-7-5-mango/>.
- [2] Boryana Miloshevska: Windows Phone Toolkit In Depth [online]. 2011 [cit. 4.5.2012]. <http://www.windowsphonegeek.com/upload/ebooks/Windows Phone Toolkit in Depth 2nd edition.pdf>.
- [3] Charles Petzold: *Programming Windows Phone 7*. Microsoft Press, 2010, iSBN 978-0-7356-4335-2.
- [4] David Allen: *Mít vše hotovo: jak zvládnout práci i život a cítit se při tom dobře*. Jan Melvil, 2008, 978-80-903912-8-4.
- [5] David Allen: *Aby vše klapalo: jak hravě zvládat pracovní i životní výzvy*. Jan Melvil, 2009, iSBN 978-80-87270-00-4.
- [6] Derek Lakin: Customizing MessageBox on Windows Phone 7 [online]. 2011 [cit. 23.4.2012]. <http://blog.dereklakin.com/customizing-messagebox-on-windows-phone-7/>.
- [7] Jeff Wilcox: The high performance ProgressBar for Windows Phone [online]. 2011 [cit. 28.4.2012]. <http://www.jeff.wilcox.name/2010/08/performanceprogressbar/>.
- [8] Microsoft: Application Certification Requirements for Windows Phone [online]. 2012 [cit. 19.4.2012]. [http://msdn.microsoft.com/en-us/library/hh184843\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/hh184843(v=vs.92).aspx).
- [9] Microsoft: Coding4Fun Windows Phone Toolkit [online]. 2012 [cit. 21.4.2012]. <http://coding4fun.codeplex.com/>.
- [10] Microsoft: Execution Model Overview for Windows Phone [online]. 2012 [cit. 18.4.2012]. [http://msdn.microsoft.com/en-us/library/ff817008\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff817008(v=vs.92).aspx).
- [11] Microsoft: General Design Principles [online]. 2012 [cit. 19.4.2012]. [http://msdn.microsoft.com/en-us/library/hh202906\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/hh202906(v=vs.92).aspx).
- [12] Microsoft: Hardware Specifications for Windows Phone [online]. 2012 [cit. 19.4.2012]. [http://msdn.microsoft.com/en-us/library/ff637514\(v=vs.92\).aspx](http://msdn.microsoft.com/en-us/library/ff637514(v=vs.92).aspx).
- [13] Microsoft: SketchFlow Template for Windows Phone [online]. 2012 [cit. 21.4.2012]. <http://wp7sketchflow.codeplex.com/>.

- [14] Microsoft: Tiles Overview for Windows Phone [online]. 2012 [cit. 19.4.2012].  
[http://msdn.microsoft.com/en-us/library/hh202948\(v=VS.92\).aspx](http://msdn.microsoft.com/en-us/library/hh202948(v=VS.92).aspx).
- [15] Microsoft: Windows Phone - MarketPlace [online]. 2012 [cit. 26.4.2012].  
<http://windowsphone.com>.
- [16] Microsoft: Windows Phone Design System - Codename Metro [online]. [cit. 2.5.2012].  
<http://go.microsoft.com/fwlink/?LinkID=189338>.
- [17] Sebastian Stehle: ImageTools for Silverlight [online]. 2011 [cit. 16.4.2012].  
<http://imagetools.codeplex.com/>.

# Příloha A

## Obsah CD

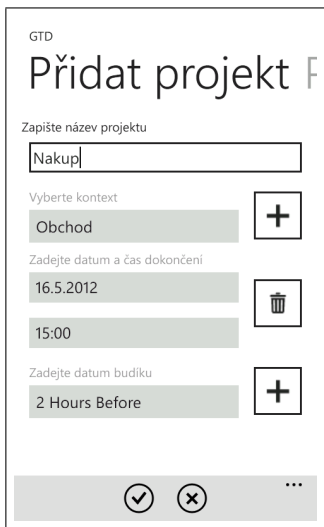
- Zdrojové kódy aplikace iGTD
- Výsledná aplikace iGTD
- Videoprezentace aplikace iGTD
- Technická zpráva
- Zdrojové texty technické zprávy v  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
- Plakát pro prezentaci

## Příloha B

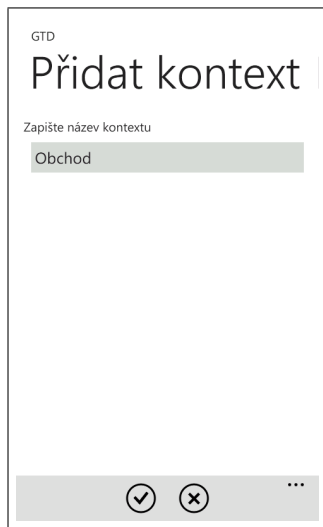
# Dosažené výsledky



Obrázek B.1: Obrázky z aplikace iGTD



Přidání projektu



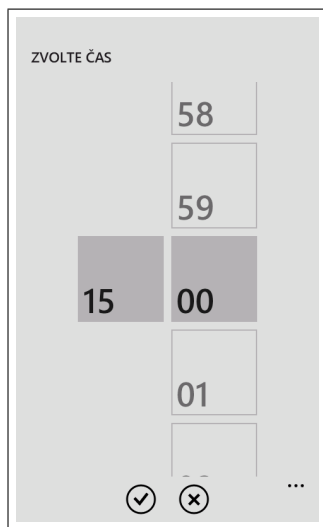
Přidání kontextu



Přidat záležitost



Vybrání data



Vybrání času



Vybrání budíku

Obrázek B.2: Obrázky z aplikace iGTD



Obrázek B.3: Obrázky z aplikace iGTD

# Příloha C

## Email

### Email 5.5.2012

Nice looking app! (iGTD fo windows mobile 7.5)

I am trying it out now.

The registration (?) web link below looks a little basic - and I had to change it to <http://gtdweb.aspone.cz/?ID=28> to follow it on the phone.

(The missing / was most critical)

Am I really user 28?

kind regards

### Email 8.5.2012

Hi,

Thanks for the reply, and good luck with your thesis!

Chrome is obviously even more flexible than I knew.

I really like your main page - the coloured hexagons make the main menu of options stand out very clearly.

So the UI layout seems clean and clear to me (and I hope you get good marks for it!)

I just saw another (very small) error - if I add a task with + from the home screen the "... settings shows the ? (tick/check) mark is labelled as "Cancel" - same as for the ?.

BTW I am also looking at similar app called "Anytask!". (Also free - I am either too mean or too committed to open software to pay for apps!!)



FYI, Anytask! is not so nice to use, it feels more cluttered and has more options, (but it does sync with Toodledo for web access which is cool, but that may be outside the scope of your project.)

I do like your UI so I hope my feedback is useful.

Maybe you could allow user options in the blank ones? (But that might make it less "clean" looking and I don't know what we might want to access as a shortcut.)

In comparison Anytask! provides a rectangular block of 8 squares - which does not even fit on the screen properly in portrait mode. Clearly inferior!

kind regards