

**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ**  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

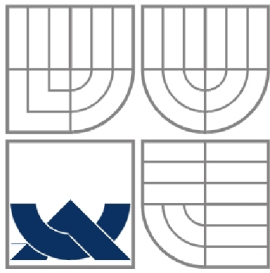
**WINDOWS MOBILE –**  
**IMPLEMENTACE NAHRÁVÁNÍ HOVORŮ**

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

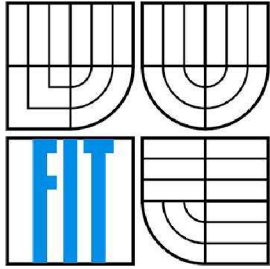
**AUTOR PRÁCE**  
AUTHOR

**RADIM TRÁVNÍČEK**

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER SYSTEMS

WINDOWS MOBILE – IMPLEMENTACE NAHRÁVÁNÍ HOVORŮ  
WINDOWS MOBILE – IMPLEMENTATION OF CALL RECORDING

BAKALÁŘSKÁ PRÁCE  
BACHELOR'S THESIS

AUTOR PRÁCE  
AUTHOR

RADIM TRÁVNÍČEK

VEDOUCÍ PRÁCE  
SUPERVISOR

ING. IGOR SZÖKE

BRNO 2009

## **Abstrakt**

Tato práce si klade za cíl seznámení se s manipulací se zvukovým tokem v mobilních zařízeních pracujících s operačním systémem Windows Mobile a vytvoření aplikace nahrávající tento zvukový proud. Zaměřuje se především na zvukový tok v okamžicích, kdy je mobilní zařízení ve stavu aktivního telefonního hovoru a vlastník mobilního zařízení telefonuje s jinou osobou. Vysvětluje principy práce se zvukem, výhody i nevýhody jak programové implementace, tak hardwarových nedostatků a v dalších částech se detailně zabývá principem zachytávání zvukového toku, analýzou, návrhem a implementací aplikace.

## **Klíčová slova**

PDA, .NET, .NET Compact Framework, C#, Windows Mobile, Windows Forms

## **Abstract**

The objective of this thesis is to take up with work with sound stream in a mobile device using the operation system Windows Mobile. It focuses on the sound stream in the moments when the mobile device is in the active call mode when the user of the mobile device speaks to someone else. The thesis describes the main principles of work with the sound stream, advantages and disadvantages of the software solution as well as of the hardware disadvantages of the mobile devices. Further the thesis focuses in detail on capture of the sound stream, its analysis and the application design and implementation.

## **Keywords**

PDA, .NET, .NET Compact Framework, C#, Windows Mobile, Windows Forms

## **Citace**

Trávníček Radim: Windows Mobile - implementace nahrávání hovorů, bakalářská práce, Brno, FIT VUT v Brně, 2009

# Windows Mobile - implementace nahrávání hovorů

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Szöke. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Radim Trávníček  
13.5.2009

## Poděkování

Děkuji vedoucímu své bakalářské práce Ing. Igoru Szöke za vedení, cenné rady, doporučení a pomoc při vypracování této bakalářské práce.

© Radim Trávníček, 2009

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

# Obsah

Obsah .....	1
1 Úvod.....	3
1.1 Stanovení cíle bakalářské práce .....	3
2 Charakteristika mobilních zařízení .....	4
2.1 Mobilní zařízení PDA .....	4
2.1.1 Stručná historie .....	4
2.1.2 Charakteristické rysy kapesního počítače .....	4
2.1.3 Současné PDA obsahuje .....	5
2.1.4 Služby PDA .....	5
2.2 Windows Mobile .....	5
2.3 Historie Windows Mobile .....	6
2.3.1 Windows Mobile 5.0.....	6
2.3.2 Windows Mobile 6.....	6
2.3.3 Windows Mobile 6.1.....	7
2.4 Telefonní hovor .....	7
2.5 Programování aplikací pro Windows Mobile.....	7
2.5.1 Sledování systémových stavů ve Windows Mobile.....	8
2.5.2 Získání čísla volajícího .....	8
2.5.3 Určení začátku a konce telefonního hovoru.....	8
2.6 Nahrávání zvuku v mobilních zařízeních.....	9
3 Analýza a návrh aplikace .....	10
3.1 Analýza.....	10
3.1.1 Práce se zvukem ve Windows Mobile .....	10
3.1.2 Problém záznamu zvuku .....	10
3.1.3 Problém hardwarového mikrofону .....	11
3.1.4 Určení začátku a konce nahrávání .....	11
3.1.5 Přenos záznamů se vzdáleným serverem .....	11
3.2 Návrh aplikace.....	11
3.2.1 Části aplikace .....	11
3.2.2 Princip funkce programu.....	11
3.2.3 Adresářová struktura.....	12
3.2.4 Vývojové diagramy.....	12
3.2.5 Hlavní Menu a Nastavení.....	12
3.2.6 Nahrávání telefonních hovorů.....	13

3.2.7	Diktafon .....	14
3.2.8	Vyhledávání slov .....	14
3.2.9	Odeslání záznamu na vzdálený server .....	15
3.2.10	Stahování log souborů ze vzdáleného serveru.....	15
3.2.11	Přehrávání záznamů.....	15
4	Použité vývojové prostředky.....	17
4.1	Programovací jazyk C# .....	17
4.2	.NET Framework a .NET Compact Framework .....	17
4.3	Vývoj aplikací v .NET Frameworku .....	17
4.4	Architektura .NET Frameworku.....	18
4.5	Windows Forms .....	20
4.6	Sockety .....	20
4.7	Microsoft Visual Studio 2008 .....	20
5	Implementace aplikace.....	21
5.1	Vytvoření vlastních tříd pro práci se zvukem .....	22
5.2	Menu (Form1) .....	22
5.3	Word Searching (Form2) .....	23
5.4	Dictaphone (Form3) .....	24
5.5	Send a record (Form4).....	25
5.6	Call recording (Form5).....	26
5.7	Call recording – Blacklist (Form6) .....	27
5.8	Call recording – Whitelist (Form7).....	28
5.9	Audio Player (Form8) .....	29
5.10	Download log file (Form9) .....	30
5.11	Options (Form10).....	31
5.12	Server aplikace – Incoming files.....	32
5.13	Server aplikace – Outgoing files .....	32
6	Požadavky na mobilní zařízení .....	34
6.1	Instalace aplikace .....	34
6.2	Testování aplikace.....	34
7	Závěr .....	35
	Literatura .....	36
	Seznam příloh .....	37
	Příloha 1 – Vývojové diagramy .....	38
	Příloha 2 – Obsah CD .....	48

# 1 Úvod

Mobilní telefonování bylo ještě před zhruba patnácti lety věci téměř neznámou a mobilní telefony mohlo vlastnit jen malé množství vyvolených. První přístroje vážily i přes kilogram a svými rozměry se nevešly do kapsy a mnohdy ani do aktovky. Výdrž jejich baterií byla slabá a telefony softwarově zvládly pouze telefonování a někdy i zaslání SMS.

Postupem času se mobilní telefony zmenšovaly, zvětšovala se výdrž baterií a rozšiřovaly se aplikace, které však většinou sloužily pro pracovní využití (kalendář, synchronizace s MS Outlook).

Na začátku tohoto století výrobci mobilních telefonů pochopili, že je nutné oslovit i další skupiny potenciálních zákazníků, hlavně mladých lidí, kteří nebudou mobilní zařízení využívat jen pro práci, ale také pro zábavu. A tak vznikly první multimediální mobilní zařízení, které umožňovaly kromě telefonování a zaslání SMS také přehrávat hudbu, video, fotografovat pomocí vestavěného fotoaparátu a další.

Tento vývoj pokračuje i nyní a dnes je běžné se z mobilního telefonu připojit k internetu (díky Wi-Fi adaptéru nebo datovým službám) a prohlížet si tak webové stránky, číst si elektronickou korespondenci a podobně.

Některé mobilní telefony navíc už obsahují i odlehčený operační systém, kancelářské balíčky a GPS (Global Positioning System), takže téměř nahrazují klasické počítače, přičemž jejich největší výhoda spočívá v mobilitě.

Přestože primární funkcí takovýchto zařízení je telefonování, stále více se mu přibližují i další aplikace a funkce, které známe z použití na klasických počítačů a které se neustále vyvíjejí.

## 1.1 Stanovení cíle bakalářské práce

Tato bakalářská práce si klade za cíl seznámit s problematikou nahrávání zvukového proudu během telefonního hovoru, provést analýzu, vhodný návrh tohoto procesu a následnou implementaci aplikace, která bude určena k nahrávání telefonních hovorů.

Návrh a následná implementace dané aplikace pro nahrávání telefonních hovorů jsou koncipovány pro užití na PDA zařízeních s operačním systémem Windows Mobile 6 a z toho plynoucí podpory technologie .NET Compact Framework. Důraz je kladen také na uživatelskou přívětivost a jednoduchost užití výsledné aplikace.

Program bude prostřednictvím komunikace přes internet na vzdáleném serveru využívat aplikaci pro rozpoznání řeči ve zvukovém záznamu. Samotná aplikace pro rozpoznání řeči není cílem této bakalářské práce.

## 2 Charakteristika mobilních zařízení

Tato kapitola popisuje základní vlastnosti mobilních zařízení a operačního systému Windows Mobile. Dále se věnuje základnímu popisu programování aplikací pro Windows Mobile a nejdůležitějších funkcí, které budu používat při tvorbě aplikace.

### 2.1 Mobilní zařízení PDA

PDA (Personal Digital Assistant - osobní digitální pomocník) či palmtop je malý kapesní počítač podobný mobilnímu telefonu. Bývá ovládán obvykle dotykovou obrazovkou a perem (které se nazývá stylus). Původně měly PDA za cíl především pomoci s organizováním času a kontaktů. Současné PDA jsou velmi výkonné a zvládají i přehrávání videa a velké množství dalších aplikací. Často se používají pro prohlížení internetu, elektronickou poštu nebo čtení elektronických knih.

Srdcem mobilních zařízení je operační systém. Nejčastěji používané operační systémy jsou Windows Mobile, Linux a Symbian [1].

#### 2.1.1 Stručná historie

Historie PDA se začala psát v roce 1993. Firma Apple na trh uvedla přístroj Newton, který byl popisován jako Personal Digital Assistant. Jeho funkce byly výhradně zaměřeny na osobní agendu. Když v roce 1996 firma Palm přišla na trh s přístrojem zvaným Palm Pilot, začal tržní pohyb kapesních počítačů. Koncem devadesátých let se objevují další výrobci, kteří začali licencovat systém. PDA se stávají součástí výrobního portfolia všech velkých počítačových firem. Již od začátku bylo PDA orientováno na efektivní správu času svého majitele a tato funkce mu zůstává dodnes [1].

#### 2.1.2 Charakteristické rysy kapesního počítače

- Hmotnost se pohybuje v průměru mezi 100 a 250 gramy, rozměry obvykle nepřesahují 15 × 10 × 2 cm. Má vlastní zdroj energie, baterii. Díky tomu je vysoce mobilní, menší typy se dají bez problémů nosit v kapse.
- Je vybaven dotykovým displejem, který se ovládá stylusem, což je tužka s umělohmotným hrotem, nahrazující funkci kurzoru myši na PC.
- PDA nemá plnohodnotnou klávesnici, jaká je u klasického PC. Text se zadává pomocí systémů rozpoznávajících písmo. Ani jeden ze systémů není určen pro psaní dlouhého textu, ale pro zaznamenávání adres, poznámek, úkolů atd.
- PDA je schopno komunikovat s okolím a to pomocí bezdrátových technologií nebo kabelů.
- Základní funkcí je synchronizace, umožňující výměnu dat mezi PDA a stolním počítačem.
- PDA má procesor, paměť, operační systém, pro které lze programovat aplikace.



### 2.1.3 Současné PDA obsahuje

- Barevný LCD displej o rozlišení 320x240 (QVGA), 640x480 (VGA), 800x480 (Wide VGA), případně o jiném rozlišení.
- Procesor s taktem kolem 400-600Mhz
- Paměť 128 - 256 MB RAM - tato paměť je závislá na zdroji energie (při přerušení napájení jsou všechna data ztracena) a proto je její velikost omezoována kapacitou baterie
- Paměť flash ROM - na rozdíl od RAM není závislá na zdroji napájení a proto je využívána jako „bezpečné uložení“ dat (včetně OS); u mnoha typů PDA je běžně ovšem uživateli nepřístupná
- U dnešních přístrojů je však čím dál více používána jako "paměť RAM" paměť typu flash což na jednu stranu sice zpomaluje práci s pamětí, ale při úplném vybití nehrozí ztráta obsahu paměti. Odpadá také nutnost udržovat PDA stále částečně zapnuté (nutné udržování paměti typu RAM) a zvyšuje se tím výdrž zařízení. Mimo to je flash stále rychlejší a rozdíly v přístupové době se stále zmenšují.
- Jeden nebo dva sloty (CompactFlash, Secure Digital apod.), které umožňují PDA rozšířit např. o paměť Flash k ukládání dat, GPS modul, síťový (network) adaptér, modem atd.
- Slot na kartu SIM, pokud daný model PDA obsahuje integrovaný modul GSM/GPRS. Pokud jej obsahuje, lze PDA provozovat i jako mobilní telefon nebo k připojení na internet bez dalších periférií

### 2.1.4 Služby PDA

Vzhledem k tomu, že PDA obsahuje operační systém, je možné nainstalovat nepřehledné množství aplikací [2]:

- Textové editory
- Tabulkové procesory
- Aplikace pro vytváření prezentací
- Emailové klienty
- Online kalendáře a organizátory času, využívající synchronizaci se stolním počítačem
- Instant messangery -chaty
- Software pro telekonference - např. Skype
- Webové prohlížeče
- Přehrávání videa a MP3

a mnoho dalších.

## 2.2 Windows Mobile

Windows Mobile je operační systém firmy Microsoft, který je založen na Windows CE. Je určen pro mobilní zařízení (PDA, smartphony a zařízení portable media center). Používá vzhled odvozený od klasických Microsoft Windows a malou podmnožinu jejich Win32 API, avšak obsahuje zcela jiné hybridní jádro.

Poslední aktuální verze systému je Windows Mobile 6 (existuje ve třech verzích a to Standard, Professional a Classic). WM 6 se dočkala inovace v podobě verze 6.1, která je určena pouze pro zařízení s dotykovým displejem. Hlavní předností je upravený interface, který usnadní celkové ovládání telefonu [3].

## 2.2.1 Základní vlastnosti Windows Mobile

Windows Mobile má ve většině verzí tyto rysy:

- Obrazovka Dnes ukazuje datum, informace o vlastníkovi, nadcházející události, e-mailové zprávy a úkoly. Uživatelé mohou upravit, které informace se mají na obrazovce zobrazovat. Na obrazovce se nachází rovněž stavové ikony Bluetooth, Wi-Fi atd. Programy nainstalované v zařízení mohou přidat na obrazovku další položky. Obrázek na pozadí lze změnit přímo v zařízení nebo lze změnit celé téma.
- Horní pruh ukazuje čas, hlasitost a stav připojení. Hlavní součástí horního pruhu je tlačítko Start, které je podobné tomu z desktopové verze Windows. Start menu zahrnuje seznam nedávno spuštěných programů, devět upravitelných položek menu a odkazy na seznam programů, nastavení, vyhledávání a nápovědu.
- Office Mobile je sada mobilních verzí aplikací Microsoft Office zahrnutých ve Windows Mobile. Zahrnuje Word Mobile, Excel Mobile a PowerPoint Mobile (od verze Windows Mobile 5.0). Mobilní verze mají mnoho rysů desktopových verzí, ale některé funkce byly značně redukovány.
- Outlook Mobile zahrnuje úkoly, kalendář, kontakty a poštu, které lze synchronizovat s aplikací Outlook nebo s přímo s Exchange Server
- Windows Media Player pro Windows Mobile podporuje formáty .WMA, .WMV, .MP3 a .AVI
- Klient pro PPTP VPN

## 2.3 Historie Windows Mobile

Podobně jako operační systém Windows určený pro stolní PC se i jeho mobilní odnož s přídomkem Mobile neustále vyvíjí a vylepšuje.

### 2.3.1 Windows Mobile 5.0

Windows Mobile 5.0, dříve známý pod kódovým označením Magneto, byl představen 9. května 2005 na Mobile and Embedded Developers Conference 2005 (MEDC) v Las Vegas. Systém pohání Windows CE 5.0 a používá .NET Compact Framework 1.0 SP2.

### 2.3.2 Windows Mobile 6

Windows Mobile 6, známý dříve pod kódovým označením Crossbow, je poslední vývojovou řadou tohoto mobilního operačního systému, která byla představena 12. února 2007 na 3GSM World Congress 2007. Existují celkem tři verze systému: Windows Mobile 6 Standard pro smartphony Windows Mobile 6 Professional pro PDA s integrovaným telefonem (Pocket PC Phone Edition) a Windows Mobile 6 Classic pro PDA bez telefonního modulu. Windows Mobile 6 je úzce propojen s Windows Live and Exchange 2007. Windows Mobile 6 Standard poprvé nabídl Orange SPV E650, zatímco Windows Mobile 6 Professional se nejdříve stal součástí O2 Xda Terra [6].

Přehled vlastností Windows Mobile 6:

- Podpora rozlišení 800x480 (WVGA).
- Schopnost upravovat dokumenty Office na smartphonech
- Schopnost automatických aktualizací systému
- Zlepšení přístupu přes vzdálenou plochu
- Podpora VoIP
- Integrace Windows Live
- Šifrování dat na výměnné paměťové kartě.
- Smartfilter pro rychlejší vyhledávání
- Podpora prohlížení HTML e-mailů v Outlook Mobile
- Schopnost hledat v Exchange Server Address Book.
- Podpora AJAX, JavaScript a XMLHttpRequest v Windows Internet Explorer Mobile

### 2.3.3 Windows Mobile 6.1

Windows Mobile 6.1 byl představen 1. dubna 2008. Tento operační systém je určen pro zařízení, která jsou vybavena dotykovým displejem (smartphone telefony). Její hlavní předností by měl být upravený interface, který usnadní celkové ovládání telefonu.

V průběhu roku 2009 by měl vyjít nástupce WM 6.1 a to systém Windows Mobile 6.5. Na internetu se již objevily informace o práci na dalším nástupci, Windows Mobile 7.

## 2.4 Telefonní hovor

Telefonní hovor je způsob komunikace probíhající zpravidla mezi dvěma účastníky po telefonním vedení nebo bezdrátově, za pomoci telefonního přístroje.

Popis sestavení hovoru a jeho průběhu z pohledu nejnižší úrovně není náplní této bakalářské práce.

Podrobně o této problematice pojednává literatura [4].

## 2.5 Programování aplikací pro Windows Mobile

V minulosti bylo pro vývojáře k dispozici několik vývojových nástrojů. Microsoft vytvořil několik vývojových prostředí pokrývajících v podstatě všechny Pocket PC a Smartphone zařízení založené na architektuře Windows Mobile.

Jednalo se o tyto vývojové nástroje:

- eMbedded Visual Tools
- eMbedded Visual C++
- Visual Studio .NET
- ASP .NET Mobile Controls (rozšíření .NET Compact Framework a Visual Studio .NET)

Společnost Microsoft postupem času sjednotila všechny doposud nabízené nástroje a poskytla tím vývojáři nástroj univerzální (Visual Studio) založený na jednotné platformě. Touto platformou je .NET Framework, která se stává hlavním hnacím motorem pro všechny aktivity společnosti Microsoft [5].

## 2.5.1 Sledování systémových stavů ve Windows Mobile

Monitorování různých stavů ve Windows Mobile nebylo vždy jednoduché. Často jen nejzkušenější vývojáři si s touto výzvou dokázali poradit. Na základě toho přišel Microsoft s novým způsobem řízení stavů a upozornění pro Windows Mobile 5.0 (State and Notification Broker API). Toto nové API poskytuje vývojářům v C# VB.NET (Visual Basic.NET) řadu výhod v omezené velikosti aplikačního kódu.

### Popis State and Notification Broker API

API poskytuje operačnímu systému Windows Mobile 5.0 prostřední s novou, zabudovanou strukturou upozornění. Programátoři používající .NET Compact Framework verze 2.0 mohou využívat tento mechanismus, ale API není možno využít pro vývoj aplikací pro dřívější operační systémy. Instalace API je zahrnuta přímo v instalaci operačního systému Windows Mobile 5 a vyšší.

API se skládá z několika hlavních komponent. Třída SystemState umožňuje přístup k specifickým systémovým stavům. Číselník SystemProperty obsahuje seznam dostupných systémových stavů, které můžeme sledovat. K monitorování systémových stavů se používá také speciální ovladač událostí (Event handler), který je obsažený v Microsoft.WindowsMobile.Status.ChangeEventHandler .

Implementace kódu k monitorování a reakci na změnu stavu systému je relativně přímý proces.

### Kroky k implementaci State and Notification Broker API

- Přidat nezbytné reference do svého .NET Compact Framework projektu
- Identifikace konkrétního systémového stavu k monitorování
- Přidat instanci systémového stavu do projektu.
- Přidat ovladač událostí (event handler) pro systémový stav do projektu
- Napsat kód reagující na změnu v systémovém stavu

Změnu těchto stavů je možno sledovat právě v okamžiku změny hodnoty (většinou pravda/nepravda) [6].

## 2.5.2 Získání čísla volajícího

Díky sledování systémových stavů Windows Mobile pomocí vývojových prostředků je možné v okamžiku, kdy s někým hovoříme, získat údaje o této osobě, jako například telefonní číslo, jméno (pokud máme osobu uloženou v kontaktech) a další, jako je například počet aktivních hovorů.

## 2.5.3 Určení začátku a konce telefonního hovoru

Pomocí nastavení sledování těchto změn, jak píše výše, je možno zjistit, kdy začal telefonní hovor, kdy je ukončen a tak je možno ve správný okamžik zavolat funkce pro spuštění nahrávání a ukončení nahrávání.

## **2.6 Nahrávání zvuku v mobilních zařízeních**

Pro zachycení zvukového toku slouží v mobilním zařízení mikrofon, který je implementován jako jeho hardwarová součást. Ve výsledku nahrávání zvuku funguje tak, že mikrofon zachytává veškerý zvuk ze zdrojů zvuku ve svém okolí.

Zde dochází k dvěma hlavním problémům, rozdílnosti kvality mikrofonu dle výrobce a omezení na získání zvuku pouze z mikrofonu. Těmto problémům se věnuji více v analýze.

## 3 Analýza a návrh aplikace

Tato kapitola se zabývá analýzou a návrhem řešení výsledné aplikace pro nahrávání telefonních hovorů. Řeší problémy záznamu zvukového toku, jeho nedostatky a také nedostatky hardwarového řešení mobilních zařízení.

V části návrhu aplikace popisují, jak budoucí aplikace bude fungovat.

### 3.1 Analýza

Hlavním cílem analýzy bylo poznat způsoby, jak je možno v mobilních zařízeních s Windows Mobile zaznamenávat zvuk, možné problémy a vybrat nejvhodnější řešení.

Jako nejlepší zdroj informací pro práci se zvukem ve Windows Mobile (a celkově pro vývoj aplikací pro Windows Mobile) se jeví Microsoft Developers Network [7], což je komunitní server sdružující programátory, kteří vytvářejí aplikace pro všechny typy operačního systému Windows. Zde programátoři Microsoftu přímo radí, jaké vývojové prostředí používat a jak si počínat při různých problémech při vývoji aplikací.

Dalším zdrojem informací byla diskusní fóra programátorů pracujících s C# a .NET popř. .NET Compact Framework [8,9], o kterých se zmiňuji dále. Díky jejich radám jsem zjistil, jak přesně zaznamenávat zvuk, jak řídit spouštění nahrávání, jeho ukončení a jak řešit problémy vzniklé při tvorbě aplikace a o kterých se zmiňuji dále.

#### 3.1.1 Práce se zvukem ve Windows Mobile

Po zkoumání, jakým způsobem pracovat se zvukem v zařízeních s Windows Mobile a jak ho ukládat, jsem došel k závěru, že .NET Compact Framework [9] neumožňuje programátorovi nahrávat zvuk přímo, protože k tomu nemá implementované funkce. Proto si programátor musí vytvořit třídy, které definují funkce pro práci se zvukem jako je nahrávání, ukládání a přehrávání zvuku.

#### 3.1.2 Problém záznamu zvuku

Během analyzování nahrávání zvuku v zařízeních se systémem Windows Mobile jsem narazil na problém nastavení bytového bufferu, do kterého se ukládá zaznamenaný zvuk. Tomuto bufferu se musí před začátkem nahrávání nastavit jeho pevná velikost, takže nelze nahrávat zvuk neomezeně dlouho, ale jsme omezeni velikostí nahrávacího bufferu. Možným řešením je nastavit buffer dostatečně velký (např. 1 hodinu – 3600000 ms). Dalším možným řešením je nastavit buffer na menší velikost a v průběhu nahrávání ho testovat, zda již je naplněn a pokud ano, uložit jeho obsah do souboru, vyprázdnit ho a začít znova nahrávat, dokud nebude hovor ukončen druhou stranou nebo uživatelem.

Další problém, na který jsem narazil je ten, že Windows Mobile neumožňuje zachytávat zvuk z mikrofonu a reproduktoru zařízení najednou. To znamená, že softwarově můžeme zachytit zvuk pouze z mikrofonu zařízení a jsme tedy závislí na hardwarovém řešení mobilního zařízení. Vysvětlení je uvedeno v další kapitole.

### **3.1.3 Problém hardwarového mikrofonu**

Jak je uvedeno výše, nelze ukládat jeden tok zvuku z mikrofonu a druhý tok z reproduktoru zařízení do jednoho bufferu. Při zkoumání a analyzování problémů jsem nenalezl řešení, jak zachytit zvukovou stopu, která by měla být přehrána v reproduktoru, ještě před jejím přehráním.

Z toho vyplývá, že jedinou možností jak zaznamenat kompletní telefonní hovor, je jej zachytit mikrofonem a to tak, že mikrofon bude nahrávat i výstup reproduktoru. Zde ovšem může dojít k problému, že výrobce zařízení umístí mikrofon příliš daleko od reproduktoru, takže záznam zvuku z mikrofonu bude v pořádku, ale záznam zvuku z reproduktoru může být nekvalitní. Záleží na výrobcu, jak mobilní zařízení po hardwarové stránce sestavil.

### **3.1.4 Určení začátku a konce nahrávání**

Největším problémem bylo zjistit, jak v systému detekovat informaci o začátku a konci telefonního hovoru, ale následně jsem našel řešení v podobě řízení událostí na základě změny systémového stavu ve Windows Mobile, o kterém se zmiňuji v kapitole 2.

Jako nejlepší řešení se tedy jeví sledovat stav mobilního zařízení a v okamžiku, kdy začne telefonní hovor (změní se stav indikující telefonní hovor), vyvolám funkci pro spuštění nahrávání. Následně, když je hovor ukončen (změní se opět stav indikující telefonní hovor na původní hodnotu), vyvolám funkci pro ukončení nahrávání a záznam uložím do souboru.

### **3.1.5 Přenos záznamů se vzdáleným serverem**

Vzhledem k tomu, že aplikace bude umožňovat odeslat na vzdálený server záznam ke zpracování a ze serveru stáhnout výsledný log soubor, bylo zapotřebí analyzovat možnosti síťové komunikace a datového přenosu.

Po zkušenostech získaných během studia na FIT se jako nejlepší možnost jeví využití Socketů [10], které vytvoří spojení mezi aplikací a vzdáleným serverem na základě IP adresy a portu. Princip funkce Socketů je více popsán v kapitole 4.

Po navázání spojení bude zapotřebí převést zvukový záznam na datový typ FileStream a následně ho odeslat na vzdálený server, po té bude spojení ukončeno. Při stahování log souboru se nejprve uloží obsah stažený se serveru do proměnné datového typu FileStream a po ukončení přenosu se jeho obsah uloží do výsledného textového souboru.

## **3.2 Návrh aplikace**

### **3.2.1 Části aplikace**

Aplikace bude poskytovat možnost nahrávat telefonní hovory s rozšířenými možnostmi, možnost záznamu zvuku (diktafon), odeslání záznamu vzdálenému serveru, stažení log souboru ze vzdáleného serveru, vyhledávání slov ve stažených log souborech.

Program bude navíc rozšířen o přehrávač záznamů.

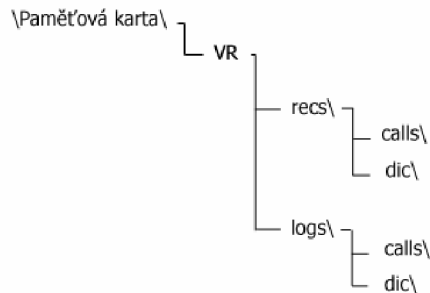
### **3.2.2 Princip funkce programu**

Program bude pracovat na bázi Windows Forms, o kterých se zmiňuji v kapitole 4. Každá jednotlivá obrazovka (Form) bude vykonávat určitou část programu. Bude se jednat o tyto části:

hlavní Menu, Nahrávání telefonních hovorů, Seznam povolených osob, Seznam zakázaných osob, Diktafon, Vyhledávání slov, Odeslání záznamu na vzdálený server, Stažení log souboru ze vzdáleného serveru, Přehrávač záznamů a Nastavení. Nad těmito formuláři budou stále běžet třídy pro práci s pamětí a zvukem [11].

### 3.2.3 Adresářová struktura

Program bude využívat adresářovou strukturu na paměťové kartě dle obrázku:



Obrázek 3.1 adresářová struktura

Na paměťové kartě po startu programu bude vytvořen adresář *VR* a v něm podadresáře *recs* a *logs*. Podadresář *recs* slouží k ukládání zvukových záznamů. V tomto adresáři se nachází další dva adresáře, *calls* a *dic*. Do adresáře *calls* budou ukládány záznamy telefonních hovorů a do adresáře *dic* nahrávky z diktafonu.

Do adresáře *logs* budou ukládány log soubory, které budou staženy ze vzdáleného serveru. Do podadresáře *calls* budou uloženy log soubory k záznamům telefonních hovorů a do podadresáře *dic* log soubory k záznamům pořízených diktafonem.

### 3.2.4 Vývojové diagramy

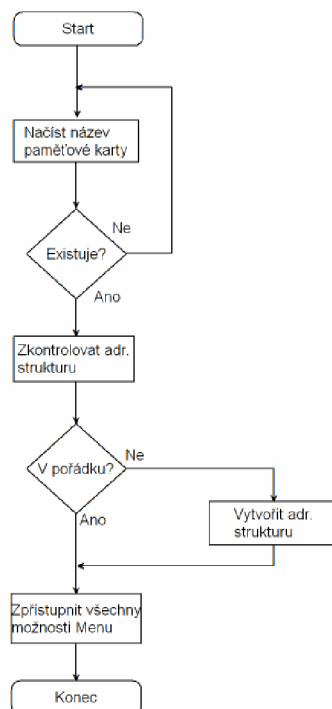
Vývojové diagramy jednotlivých částí byly z důvodu svého rozsahu vloženy do samostatné části a jejich převážná část je uvedena v příloze 1.

### 3.2.5 Hlavní Menu a Nastavení

Program po spuštění nabídne uživateli nabídku možností programu, ze kterých si uživatel vybere právě takovou volbu, k čemu chce program v nejbližších chvílích použít. Ovšem z nabídky bude ihned po spuštění aktivní pouze nastavení (Options), ve kterém bude nutno zadat název paměťové karty popř. adresáře, kam si přeje záznamy ukládat. To z toho důvodu, že výrobci paměťových karet dávají svým výrobkům rozdílné názvy a tudíž není možné zadat do zdrojového kódu programu jedno neměnné jméno paměťové karty, protože by program v případě neexistence takové paměťové karty nemohl ukládat záznamy.

Program po zadání jména otestuje, zda je paměťová karta v mobilním zařízení přítomna a pokud ano, vytvoří (pokud se tak již nestalo) na ni adresářovou strukturu a odemkne všechny možnosti programu, ze kterých si již bude možné vybírat v hlavním Menu





Obrázek 3.1 Vývojový diagram Nastavení

### 3.2.6 Nahrávání telefonních hovorů

Jako první možností v hlavním Menu bude nahrávání hovorů, které je primárním cílem této práce. Po té, co si uživatel vybere tuto možnost, bude zobrazena nabídka s třemi možnostmi, a to: Nahrávání všech hovorů, Use Blacklist a Use Whitelist.

Jak již název napovídá, při výběru první možnosti bude program nahrávat všechny příchozí a odchozí telefonní hovory a ukládat je do příslušného adresáře na paměťové kartě ve formátu *datum\_čas-začátku-hovoru\_čas-konec-hovoru-telefonní-číslo-volajícího.wav*.

U možnosti Blacklist si uživatel vytvoří seznam telefonních čísel a jmen, od kterých si nepřeje telefonní hovory nahrávat. Po zvolení této možnosti a vytvoření seznamu se v případě příchozího nebo odchozího tel. hovoru otestuje, zda se telefonní číslo popř. jméno osoby, se kterou uživatel bude komunikovat, nachází v seznamu (Blacklistu) a pokud ano, program hovor nenahrává. Uživatel si může svůj seznam (Blacklist) uložit do souboru a při příštím spuštění programu si ho z tohoto souboru nahrát, aby ho nemusel znova pracně vytvářet. Formát souboru nahrávky je shodný s formátem výše zmíněným.

Třetí možností je Whitelist, tedy seznam osob, pouze od kterých se budou hovory nahrávat. Pokud při příchozím nebo odchozím telefonním hovoru program zjistí, že telefonní číslo, popř. jméno osoby, se kterou uživatel programu bude hovořit, se nachází v tomto seznamu (Whitelistu), program hovor nahrává. Uživatel si stejně jako v případě Blacklistu může svůj seznam „povolených čísel“ uložit do souboru a při příštím spuštění programu ho využít. Formát souboru nahrávky je shodný s formátem výše zmíněným.

V této části budu využívat třídu WaveIn.cs, protože implementuje funkce pro nahrávání zvuku [12].

Vývojový diagram se nachází v příloze 1.

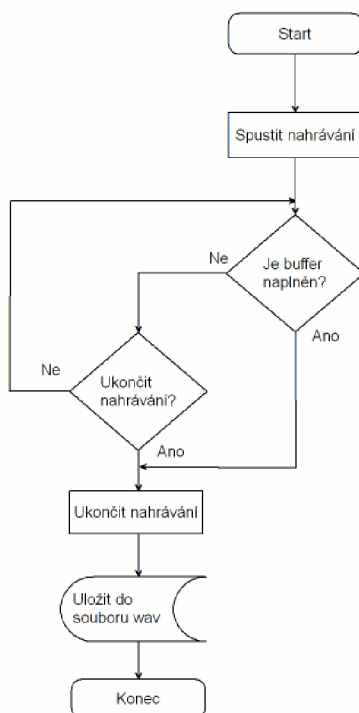
### 3.2.7 Diktafon

Jelikož pracujeme se zvukem, aplikace by měla obsahovat funkci pro nahrávání zvuk kdykoliv podle přání uživatele. K tomu slouží právě diktafon.

Po výběru diktafonu v hlavním Menu se uživateli objeví obrazovka se dvěma tlačítky (*Spustit nahrávání* a *Zastavit nahrávání*) a ukazatelem času. Po stisknutí tlačítka *Spustit nahrávání* se spustí čítač, který bude „hýbat“ s časem na obrazovce a zároveň aplikace začne nahrávat zvuk. Navíc se vypne tlačítko *Spustit nahrávání* a zaktivní se tlačítko *Zastavit nahrávání* a to z toho důvodu, aby nedocházelo ke konfliktu v aplikaci.

Formát výsledného souboru bude ve tvaru : *datum-cas-zacatku-nahravani\_cas-konce-nahravani.wav* a záznam bude uložen v příslušném adresáři.

V této části budu také využívat třídu *WaveIn.cs*.



Obrázek 3.3 Vývojový diagram Diktafon

### 3.2.8 Vyhledávání slov

Aplikace uživateli umožní ve spolupráci se vzdáleným serverem vyhledávat slova v záznamech, které byly pořízeny touto aplikací.

Uživatel si do příslušného textového okénka zadá slovo, které chce vyhledávat a vybere, zda si přeje hledat v záznamech telefonních hovorů nebo v záznamech vytvořených diktafonem.

Po stisknutí tlačítka *Hledat* se v seznamu, který bude umístěn pod tlačítkem, zobrazí názvy nahrávek, ve kterých se dané slovo vyskytuje.

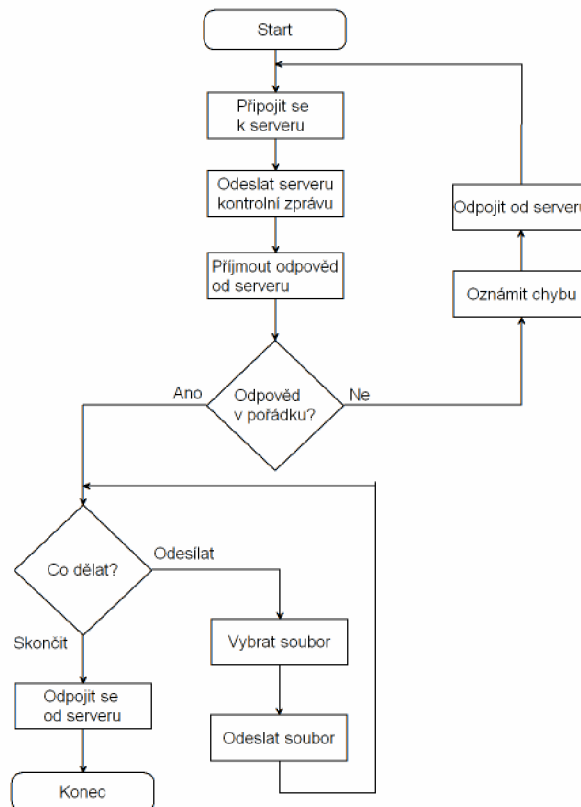
Dalším tlačítkem bude možno záznamy seřadit vzestupně nebo sestupně podle toho, s jakou pravděpodobností se v nich hledané slovo nachází.

Vývojový diagram se nachází v příloze 1.

### 3.2.9 Odeslání záznamu na vzdálený server

Aby mohl uživatel aplikace vyhledávat slova ve svých záznamech, musí nejprve odeslat záznam na vzdálený server, kde bude zpracován. Připojení je zprostředkováno pomocí Wi-Fi adaptérů nebo datových služeb GPRS.

Pro tuto komunikaci na úrovni programu bude aplikace využívat Socketů, které slouží k programování síťové komunikace [10].



Obrázek 3.4 Vývojový diagram Odeslání záznamu na vzdálený server

### 3.2.10 Stahování log souborů ze vzdáleného serveru

Poté, co uživatel odešle záznam na vzdálený server, bude mít možnost si stáhnout výsledek překladu na text v podobě log souboru ve formátu .txt . Tento log soubor bude po stažení uložen do příslušného adresáře a při vyhledávání slov (viz. výše) bude dané slovo vyhledáváno právě v těchto textových souborech.

Jméno log souboru bude shodné se jménem záznamu. Rozdílná bude pouze koncovka souboru. Pro tuto komunikaci bude program také využívat Socketů.

Vývojový diagram se nachází v příloze 1.

### 3.2.11 Přehrávání záznamů

Pořízené zvukové záznamy bude možno přehrávat v přehrávači záznamů. K tomu program využije třídu pro přehrávání zvuku.

Po výběru této možnosti se uživateli zobrazí obrazovka, ve které si vybere zda si přeje přehrávat telefonní hovory nebo záznamy z diktafonu. Po výběru jedné z těchto možností se mu v seznamu objeví názvy nahrávek závisle na výběru.

Pod seznamem budou tři tlačítka (Start, Pause a Stop) pro přehrávání. Přehrávání záznamu bude možno spustit, zastavit si ho a následně pokračovat v přehrávání nebo ho stopnout úplně.

V této části budu využívat třídu WaveOut.cs, která implementuje funkce pro přehrávání zvuku [12].

Vývojový diagram se nachází v příloze 1.

## 4 Použité vývojové prostředky

Při tvorbě a implementaci mnou navržené aplikace bylo použito několik technologií, z nichž většina je součástí Microsoft .NET architektury. V této kapitole jsou popsány ty, které měly na výslednou aplikaci největší vliv.

Jedná se především o technologii Microsoft .NET Framework, konkrétněji její odlehčenou (light) verzi Microsoft .NET Compact Framework. Na této technologii, resp. platformě je postavena celá aplikace. A to z důvodu přímého vývoje této technologie pro PDA zařízení obsahující systém Windows Mobile 6.0 a také z tohoto předpokladu plynoucí dostatečně robustní podporu včetně plně funkční kompatibility s ostatními prvky a nástroji společnosti Microsoft.

Cílem této kapitoly není detailně popsat principy jednotlivých částí .NET technologií, ale pouze přiblížit jejich funkci, možnosti využití a jejich umístění v .NET architektuře. Podrobnější informace k těmto technologiím je možno nalézt v níže uvedených odkazech na literaturu.

### 4.1 Programovací jazyk C#

Aplikace je psána v programovacím jazyce C#. Tento jazyk byl zvolen díky jeho plné kompatibilitě s níže popsanými technologiemi a také díky výhodám a usnadněním, která tento moderní objektově-orientovaný jazyk nabízí.

C# je jednoduchý, moderní, objektově orientovaný programovací jazyk z dílny Microsoftu odvozený z jazyka C a C++. Spojuje vysokou produktivitu Visual Basicu a hrubou sílu jazyka C++.

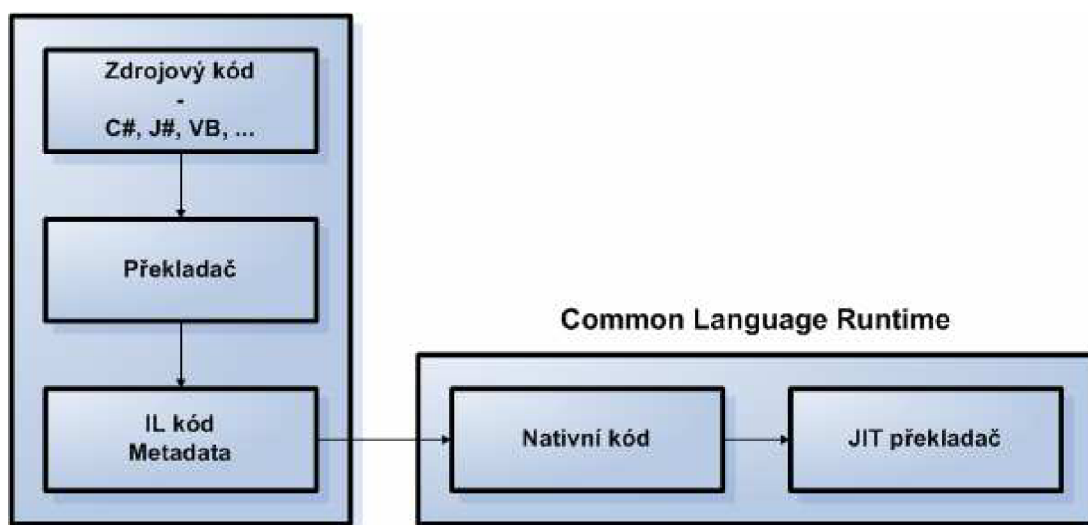
### 4.2 .NET Framework a .NET Compact Framework

Microsoft .NET Framework je softwarová infrastruktura umožňující vývoj a spouštění moderních aplikací vytvořených pro různá zařízení na platformě Microsoft. Jedná se o softwarový nástroj, který lze nainstalovat do operačních systémů firmy Microsoft a získat tak zázemí pro moderní vývoj klasických aplikací, kdy nejsme omezeni pouze na aplikace v zařízeních, ale můžeme psát také webové aplikace, dynamické webové stránky (psány v jazyce ASP.NET) a další. Nespornou výhodou je také prostředí zajišťující bezpečný běh, spouštění a používání těchto aplikací.

Pro mobilní zařízení je k dispozici Microsoft .NET Compact Framework, který je speciálně upraven pro požadavky a možnosti mobilních zařízení se systémem Windows CE nebo Windows Mobile. Je to v podstatě lite verze .NET Frameworku [8,9]

### 4.3 Vývoj aplikací v .NET Frameworku

.NET Framework poskytuje pevný objektově-orientovaný programový model pro vývoj všech typů aplikací. To znamená, že přístup k vývoji ať už pro Windows, webových aplikací či pro mobilní zařízení je vždy stejný.

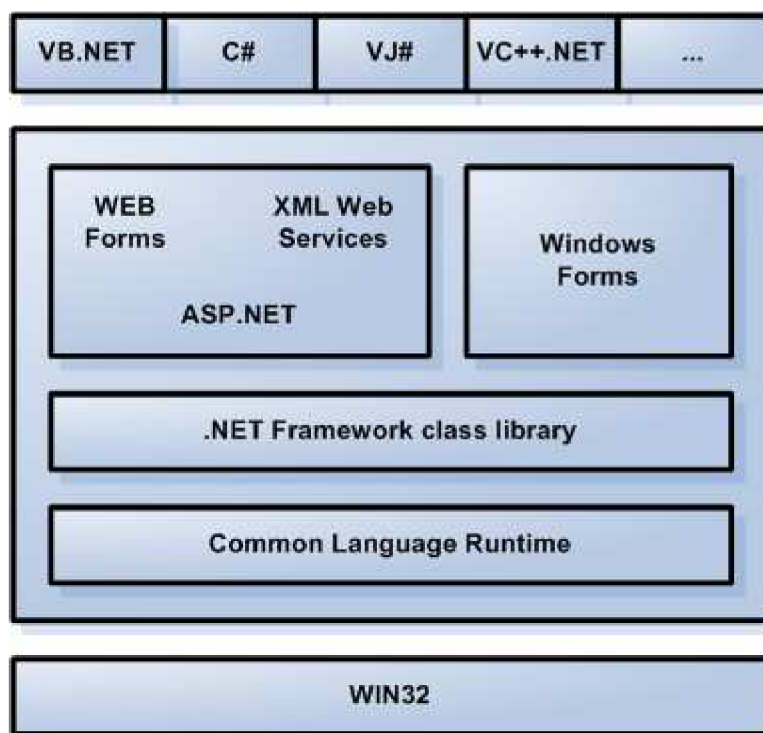


Obrázek 4.1 Princip překladau .NET aplikace [15]

K vývoji .NET aplikace je třeba vytvořit třídu a definovat její funkčnost pomocí vlastností, událostí a metod. I v případě webových aplikací je kód, který kontroluje chování webové stránky, zapouzdřen ve třídě. Třídy jsou tedy základní prvky pro vývoj aplikací v prostředí .NET. Navíc díky tzv. jazykové interoperabilitě (možnosti psát aplikaci v různých jazycích podporovaných v architektuře .NET) lze třídy psané v jednom jazyce použít nebo dědit i ve třídě psané jiným jazykem. Tato jazyková interoperabilita je zajištěna díky pravidlům definovaným v ECMA (European Computer Manufacturers Association) Common Language Specification standardu (CLS). Každý kód psaný v jazyce odpovídajícím standardu CLS je pak kompilován do tzv. IL (Intermediate Language) kódu, který je kompatibilní s ostatními IL kódy kompilovanými z jiných CLS jazyků. Tento IL kód je platformově nezávislý, jeho tzv. Just-in-time (JIT) překlad do nativního kódu konkrétního počítače pak provádí Common Language Runtime (CLR), jež je součástí .NET Frameworku.

#### 4.4 Architektura .NET Frameworku

.NET Framework se skládá ze dvou hlavních částí: Class Library a již dříve zmíněného Common Language Runtime (CLR).



Obrázek 4.2 Architektura .NET Frameworku [15]

Class Library poskytuje sadu datových typů a operací, které jsou společné pro všechny jazyky. Tyto typy mohou být použity pro vývoj různých druhů aplikací jako konzolové aplikace, Windows aplikace nebo webové služby. CLR obsahuje komponenty, které se starají o načítání IL kódu a jeho spouštění, zajišťuje bezpečnost a poskytuje podporu vláken a další užitečné funkce. Nad základní Class Library se pak nachází část poskytující rozhraní pro vývoj klasických formulářových Windows aplikací a část pro vývoj webových aplikací.

V případě .NET Compact Frameworku je jeho architektura podobná, zahrnuje však komponenty přizpůsobené pro účely mobilních aplikací. Je navržen s důrazem na co nejmenší velikost, a proto také neposkytuje tolik možností jako klasický .NET Framework. Tabulka níže názorně zobrazuje rozdíl ve velikosti, počtu tříd a počtu metod implementovaných v klasickém a Compact Frameworku [8].

	Velikost souboru			Počet tříd			Počet metod			Metoda minimalizace
	.NETCF	Desktop	%	.NETCF	Desktop	%	.NETCF	Desktop	%	
"MSCorEE"	400kB	2.2MB	18%	N/A	N/A	N/A	N/A	N/A	N/A	Kompletní přepis
MSCorLib	200kB	2MB	10%	364	1286	28%	3989	13817	29%	Úprava a ořezání
System	100kB	1.2MB	8%	140	765	18%	1090	6953	16%	Úprava a ořezání
System.Drawing	20kB	458kB	5%	41	254	16%	385	3509	11%	Kompletní přepis
System.Web.Services	67kB	503kB	13%	54	274	20%	302	2083	14%	Kompletní přepis
System.Windows.Forms	56+55kB	2MB	5%	43	823	5%	393	11337	3%	Kompletní přepis
System.XML	138kB	1.2MB	12%	100	724	14%	927	7227	13%	Úprava a ořezání

Obrázek 4.3 Rozdíl ve velikosti .NET Framework a .NET Compact Framework [15]

## 4.5 Windows Forms

Windows Forms je jméno pro rozhraní určené pro programování aplikačního grafického uživatelského rozhraní (GUI, neboli vzhledu aplikace) zahrnuté jako část Microsoft .NET Frameworku. Toto rozhraní poskytuje přístup k nativním Microsoft Windows prvkům rozhraní (jako například tlačítka, textová pole, nadpisy atd.) díky obalování existujícího Windows API v řízeném kódu. I když jsou chápány jako náhražka pro dřívější a komplexnější Microsoft Foundation Class Library (Microsoft knihovny), neposkytují vzory srovnatelné s MVC (Model-View-Controller) [11].

## 4.6 Sockety

Socket je spojení dvou procesů. Důležité je zde slovo spojení. Z něj vyplývá to, že sockety jsou způsobem meziprocesová komunikace. V praxi to znamená, že můžeme napsat dva programy, které spolu budou moci vzájemně komunikovat. První program - server - bude naslouchat na určeném portu (port je číslo reprezentující vstupní cestu do počítače) a bude čekat na připojení druhého programu - klienta. V okamžiku, kdy se klient připojí, vznikne spojení a od této chvíle mohou obě strany komunikovat. Oba programy mohou běžet na různých počítačích, které jsou síťově propojeny.

Existují dva způsoby komunikace. Se spojením nebo beze spojením - tedy TCP nebo UDP. Typickým příkladem komunikace bez spojením je email. Mezi odesílatelem a příjemcem se nevytváří žádné spojení - odeslání emailu není závislé na příjemci a přijetí na odesílateli. Není zaručeno pořadí jednotlivých zpráv ani to, že někdo zprávu skutečně přijme. Naopak mezi komunikací se spojením patří telefonní hovor. Oba účastníci jsou v jednom okamžiku spojeni komunikačním kanálem, který zaručuje správné pořadí zpráv [10].

Spojení pomocí socketů budu využívat při komunikaci se vzdáleným serverem.

## 4.7 Microsoft Visual Studio 2008

Microsoft Visual Studio je vývojové prostředí (IDE) od Microsoftu. Může být použito pro vývoj konzolových aplikací a aplikací s grafickým rozhraním spolu s Windows Forms aplikacemi, webovými stránkami, webovými aplikacemi a webovými službami jak ve strojovém kódu, tak ve spravovaném kódu na platformách Microsoft Windows, Windows Mobile, Windows CE, .NET, .NET Compact Framework a Microsoft Silverlight.

Visual Studio podporuje jazyky prostřednictvím jazykových služeb, což umožňuje, aby editor kódu a debugger podporoval jakýkoliv programovací jazyk. Mezi vestavěné jazyky patří C/C++ (použitím Visual C++), VB.NET (použitím Visual Basic .NET) a C# (použitím Visual C#). Podpora dalších jazyků jako Chrome, F#, Python a Ruby spolu s ostatními může být přidána jazykovými službami, které musí být nainstalovány zvlášť. Také je podporováno XML/XSLT, HTML/XHTML, JavaScript a CSS. Existují i verze Visual Studia pro určitý jazyk, které uživatelům poskytují omezenější jazykové služby. Tyto individuální balíčky jsou Microsoft Visual Basic, Visual J#, Visual C# a Visual C++ [13].



# 5 Implementace aplikace

Na základě analýzy a návrhu jsem si následně vybral vývojové prostředí Visual Studio 2008, ve kterém jsem aplikaci vytvořil.

Pro širší využití aplikace jsem se rozhodl napsat grafické rozhraní aplikace v anglickém jazyce.

## Seznam částí programu s anglickým překladem:

- Hlavní Menu – Menu
- Vyhledávání slov – Word Searching
- Diktafon - Dictaphone
- Odeslat záznam na vzdálený server – Send a record
- Nahrávání hovorů – Call Recording
- Přehrávač záznamů – Audio Player
- Stažení log souboru ze vzdáleného server – Download log file
- Nastavení – Options
- Seznam povolených osob – Whitelist
- Seznam zakázaných osob - Blacklist

## Použité komponenty vývojového prostředí Visual Studio 2008:

- *Label* – text
- *Button* – tlačítko
- *MenuItem* – tlačítko ve spodní liště formuláře
- *RadioButton* – zaškrtačací kolečko pro výběr
- *Listbox* – seznam
- *OpenDialog* – dialogové okno pro načtení souboru
- *SaveDialog* – dialogové okno pro uložení souboru
- *ProgressBar* – ukazatel stavu určité činnosti
- *Timer* – časové stopky
- *TextBox* – okno pro vložení textu (i více řádků)

Dále se budu podrobně zabývat implementací jednotlivých částí programu, a to nahráváním zvuku, nahráváním telefonních hovorů s rozšířenými možnostmi, přehráváním zvukových záznamu, vyhledáváním slov v log souborech, odesíláním záznamů na vzdálený server a stahováním log souborů ze vzdáleného serveru.

Tento program má implementovanou oboustrannou komunikaci se vzdáleným serverem, na kterém může běžet aplikace pro převod řeči na text. Tvorba aplikace pro zpracování řeči nebyla úkolem této bakalářské práce.

## 5.1 Vytvoření vlastních tříd pro práci se zvukem

Při analýze jsem zjistil, že pro ukládání zvuku v zařízeních s Windows Mobile budu potřebovat vytvořit nové třídy, které by implementovaly funkce pro práci se zvukem.

Na Microsoft Developer Network [7] jsem našel tyto třídy již implementované. V literatuře [12] jsou následně popsány, takže si člověk může danou třídu sestavit sám, dle návodu anebo si může ze stejného odkazu stáhnout již vytvořený balíček tříd.

Tyto třídy jsem sestavil podle návodu a následně jsem je přeprovedl a upravil podle tříd, které jsou uvedeny na Microsoft Developer Network.

## 5.2 Menu (Form1)



Obrázek 5.1 Hlavní menu

Po spuštění programu se zobrazí obrazovka úvodního Menu, kde si uživatel vybere jednu ze zobrazených možností. Výběr možností je řešen pomocí komponenty RadioButton. Těch je na úvodním formuláři 7 a mají tu vlastnost, že může být v jednom okamžiku vybrána pouze jedna, tzn. že uživatel nemůže vybrat více možností najednou.

Při prvním startu aplikace jsou všechny možnosti kromě Nastavení (Options) nepřístupné a zpřístupní se až po nastavení přístupu k paměťové kartě.

Kontrola nastavení přístupu je prováděna při každém vyvolání hlavního menu a kontroluje se, zda již byl přístup nastaven (pomocná proměnná změnila hodnotu z false na true). Pokud má při vyvolání hlavní obrazovky testovaná pomocná proměnná hodnotu true (pravda), zpřístupní se všechny možnosti hlavního menu.

## 5.3 Word Searching (Form2)



Obrázek 5.2 Vyhledávání slov

Uživatel si může vyhledat slovo ve svých zvukových záznamech. Dosáhne toho tak, že zadá hledané slovo do textového okénka a následně provede stiskem tlačítka Search hledání.

Po stisku tlačítka Search je vyvolána funkce Searchword s dvěma parametry, prvním je hledané slovo z textové okénka a druhým je cesta k adresáři, ve kterém se bude vyhledávat slovo. Cesta k adresáři je nastavena podle toho, zda si uživatel vybere Phone calls (telefonní hovory) nebo Dictaphone (diktafon).

Funkce následně prochází všechny textové soubory (log soubory vytvořené vzdáleným serverem, který převedl zvukový záznam na text) a vyhledává v nich hledané slovo.

Pokud je slovo nalezeno, testuje se, zda se zvukový záznam, jehož jméno odpovídá jménu textovému log souboru nachází na paměťové kartě a pokud ano, je jméno tohoto zvukového záznamu přidáno do výsledků hledání.

Uživatel si dále může výsledky hledání seřadit podle pravděpodobnosti výskytu slova v záznamu a to vzestupně nebo sestupně. K řazení byla použita metoda BubbleSort.

Dále si uživatel může vybraný záznam ze seznamu přehrát v přehrávači a to tak, že klikne na záznam, který chce přehrát a následně na tlačítko v liště Open player. To způsobí uložení cesty k záznamu do paměti, zavře se obrazovka s vyhledáváním a je vyvolána obrazovka s přehráváním záznamů. Při vyvolání této obrazovky je testováno, zda byla nastavena do pomocné proměnné cesta k záznamu a pokud ano, je záznam připraven k přehrávání v přehrávači.

## 5.4 Dictaphone (Form3)



Obrázek 5.3 Diktafon

Pokud si uživatel v Menu zvolí možnost Dictaphone, zobrazí se mu tato obrazovka. Jedná se o jednoduché rozhraní pro nahrávání zvuku, které umožní uživateli spustit a ukončit nahrávání kdykoliv si to bude přát.

Při vyvolání této obrazovky je alokovan buffer pro nahrávání zvuku v délce dvaceti minut a vše je připraveno k nahrávání.

Uživatel stiskem tlačítka Start recording vyvolá funkci pro spuštění nahrávání zvuku a zároveň jsou spuštěny stopky s tikem o délce jedné vteřiny. Následně je zpřístupněno tlačítko pro ukončení nahrávání Stop recording a je znepřístupněno tlačítko pro spuštění nahrávání.

V každém tiku je testováno naplnění bufferu. V případě, že by byl buffer naplněn, bylo by nahrávání ukončeno a obsah bufferu by byl uložen do souboru.

Dále v každém tiku stopek probíhá funkce, která pracuje s časem nahrávání a zobrazuje je ho obrazovku.

Nahrávání je ukončeno po stisknutí tlačítka Stop recording. Obsah bufferu je následně uložen do souboru, vynuluje se čas na obrazovce a zpřístupní se opět tlačítko pro spuštění nahrávání Start recording.

## 5.5 Send a record (Form4)



Obrázek 5.4 Odesílání záznamu

Pro odeslání zvukového záznamu na vzdálený server je nutné se nejprve připojit ke vzdálenému serveru. K tomu slouží tlačítko Connect, po jehož stisknutí se vyvolá funkce Connect(), která provede připojení.

Pokud se program ke vzdálenému serveru připojí v pořádku, v okénku stavů se zobrazí zpráva, že se program úspěšně připojil "Program is connected to server". Program zašle vzdálenému serveru kontrolní zprávu a server odpoví kladně v případě, že je možno odeslat záznam. V takovém případě se v okénku zobrazí zpráva „Server is waiting for file“.

Po úspěšném připojení se vypne tlačítko Connect a zaktivní se tlačítko Disconnect pro odpojení od vzdáleného serveru a tlačítko Send file.

Pro výběr souboru k odeslání klikne uživatel na tlačítko Send File, které vyvolá dialogové okno pro výběr souboru k odeslání.

Po vybrání souboru a potvrzení se dialogové okno pro výběr souboru zavře a program začne soubor odesílat na vzdálený server.

Po úspěšném odeslání se v okénku zobrazí zpráva, že soubor byl úspěšně odeslán „File succesfully sent“. Pokud by nedošlo k úspěšnému odeslání, program by v okénku oznámil, že došlo k chybě.

Při programování síťové komunikace se vzdáleným serverem jsem využil Socketů.

## 5.6 Call recording (Form5)



Obrázek 5.5 Nahrávání hovorů

Pokud si uživatel v hlavním Menu vybere možnost nahrávání hovorů Call Recording, vyvolá se tato obrazovka. V ní si může vybrat jednu ze tří možností.

Pokud si bude přát nahrávat všechny telefonní hovory, klikne na kolečko u nápisu Record all phone calls. Pokud si bude přát nahrávat všechny telefonní hovory kromě hovorů od lidí, které má v Blacklistu, klikne na kolečko u nápisu Use Blacklist A pokud si bude přát nahrávat telefonní hovory pouze od lidí, které má v seznamu-Whitelistu, klikne na kolečko u nápisu Use Whitelist.

Dále si může vytvořit seznam zakázaných osob (Blacklist) kliknutím na tlačítko Edit Blacklist anebo vytvořit seznam povolených osob (Whitelist) kliknutím na tlačítko Edit Whitelist.

Po nastavení režimu nahrávání je nutno spustit sledování telefonních hovorů stiskem tlačítka Apply and start recording. Po jeho stisknutí proběhne na pozadí alokace bufferu pro nahrávání hovoru v délce dvaceti minut a program začne sledovat systémové stavy související s telefonováním.

Pokud program zjistí aktivní telefonní hovor, program díky sledování stavů zjistí, se kterým telefonním číslem je hovor navázán, popř. s kterým jménem, pokud je číslo volajícího uloženo v kontaktech pod jménem a porovná s režimem nahrávání. Podle zvoleného režimu nahrávání program porovná číslo popř. jméno volajícího s údaji v seznamu zakázaných nebo povolených osob.

Na základě porovnání program rozhodne, zda hovor může nahrát a pokud ano, vyvolá program funkci pro spuštění nahrávání. Zároveň jsou spuštěny stopky, které kontrolují v každém tiku (1 sekunda), zda je buffer naplněn.

Pokud by hovor trval déle než dvacet minut, program sám vyvolá funkci pro ukončení nahrávání a uloží obsah bufferu do souboru. Zároveň ihned začíná nahrávat hovor dále, dokud program díky sledování systémových stavů nezjistí, že hovor byl ukončen. Následně program vyvolá funkci pro ukončení nahrávání a uloží obsah bufferu do nového souboru.

V této části jsem řešil závažný problém a to ten, že program nereagoval na zjištění aktivního hovoru zcela správně. Přímé vyvolání funkce pro nahrávání hovorů v reakci na detekci aktivního hovoru zde nefungovalo (program pouze vytvořil prázdný soubor). Po dlouhém zkoumání dostupných zdrojů zabývajících se nahráváním telefonních hovorů jsem sice řešení nenalezl, ale testováním různých postupů jsem problém vyřešil. Řešení spočívalo v tom, že po zjištění aktivního hovoru program vyvolá stisk tlačítka, které je uživateli skryto a po stisku tohoto tlačítka je vyvolána funkce pro nahrávání hovorů již správně. Stejným způsobem je vyřešeno ukončení hovoru.

## 5.7 Call recording – Blacklist (Form6)



Obrázek 5.6 Seznam zakázaných kontaktů

Blacklist slouží k vytvoření seznamu osob, od kterých si uživatel nepřeje telefonní hovory nahrávat.

Nový záznam do seznamu je přidán stisknutím tlačítka Add to list, které vyvolá funkci přidání položky do seznamu. Ze seznamu se dá jednoduše odebrat záznam jeho výběrem a stisknutím tlačítka Delete number, které vyvolá funkci pro vymazání vybraného záznamu ze seznamu.

Seznam je možno uložit do souboru stiskem tlačítka Save to file, které vyvolá dialogové okno, ve kterém uživatel zadá název souboru a vybere adresář, kde chce seznam uložit. Opačným způsobem funguje program po stisku tlačítka Load from file, kdy program vyvolá dialogové okno pro vybrání souboru, ze kterého se má program načíst.

Hlavním problémem zde bylo udržet seznam zakázaných osob v paměti po dobu běhu programu. To bylo vyřešeno datovým polem, které je přístupné po celou dobu běhu programu. Při vyvolání této obrazovky jsou do seznamu načteny údaje z pole v paměti a při uzavření této obrazovky jsou údaje

v paměti vymazány a jsou do ni překopírovány údaje ze seznamu. Při uzavření obrazovky je seznam vyprázdněn, ovšem údaje zůstanou uloženy v paměti.

## 5.8 Call recording – Whitelist (Form7)



Obrázek 5.7 Seznam povolených kontaktů

Whitelist slouží k vytvoření seznamu osob, od kterých si uživatel přeje telefonní hovory nahrávat.

Nový záznam do seznamu je přidán stisknutím tlačítka Add to list, které vyvolá funkci přidání položky do seznamu. Ze seznamu se dá jednoduše odebrat záznam jeho výběrem stisknutím tlačítka Delete number, které vyvolá funkci pro vymazání vybraného záznamu ze seznamu.

Seznam je možno uložit do souboru stiskem tlačítka Save to file, které vyvolá dialogové okno, ve kterém uživatel zadá název souboru a vybere adresář, kde chce seznam uložit. Opačným způsobem funguje program po stisku tlačítka Load from file, kdy program vyvolá dialogové okno pro vybrání souboru, ze kterého se má program načíst.

Hlavním problémem zde bylo udržet seznam povolených osob v paměti po dobu běhu programu. To bylo vyřešeno datovým polem, které je přístupné po celou dobu běhu programu. Při vyvolání této obrazovky jsou do seznamu načteny údaje z pole v paměti a při uzavření této obrazovky jsou údaje v paměti vymazány a jsou do ni překopírovány údaje ze seznamu. Při uzavření obrazovky se seznam vyprázdňuje, ovšem údaje zůstanou uloženy v paměti.



## 5.9 Audio Player (Form8)



Obrázek 5.8 Přehrávač záznamů

V této části programu si může uživatel přehrávat své záznamy. Pro vypsání záznamů je potřeba kliknout na Phone call nebo na Dictaphone, což vyvolá funkci, která vyhledá příslušné zvukové záznamy a jejich názvy vypíše do seznamu.

Následně je nutno vybrat některý ze záznamů v seznamu a stiskem tlačítka Play se vyvolá funkce, která začne zvukový záznam přehrávat.

S přehráváním zvuku se dá dále pracovat díky tlačítek Pause a Stop, které přehrávání pozastaví nebo úplně zastaví.

## 5.10 Download log file (Form9)



Obrázek 5.9 Stažení log souboru

V této části programu si může uživatel stáhnout výsledek zpracování zvukového záznamu vzdáleným serverem.

Program se ke vzdálenému serveru připojí po stisknutí tlačítka Connect, kdy se vyvolá funkce Connect(). Pokud se vyskytne chyba, program uživateli oznámí, že nemohlo dojít k vytvoření komunikace mezi programem a vzdáleným serverem.

Po úspěšném připojení si uživatel z nabídky vybere, zda si přeje vypsát seznam log souborů vytvořených zpracováním záznamů telefonních hovorů kliknutím na Phone Calls nebo záznamů diktafonu kliknutím na Dictaphone.

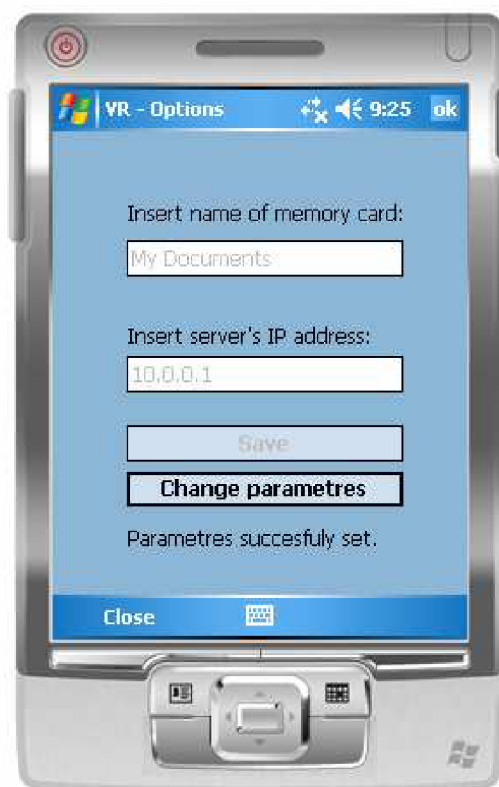
Odeslání informace na server se provede po stisknutí tlačítka Get list, kdy program odešle serveru zprávu o typu záznamů. Server zprávu přijme, provede její zpracování a odešle zpět programu seznam dostupných log souborů.

Uživatel si vybere jeden záznam se v seznamu a po stisku tlačítka Download provede program funkci, která odešle požadavek na soubor vzdálenému serveru. Ten ji zpracuje a začne odesílat soubor. Program ho po částech ukládá do paměti a v okamžiku, kdy je přenos dokončen (server již neodesílá žádná data), je obsah paměti uložen do příslušného souboru.

Po stisku tlačítka Disconnect se provede funkce, která odpojí program od vzdáleného serveru.

Při programování této komunikace jsem využil tutoriálu [14].

## 5.11 Options (Form10)



Obrázek 5.10 Nastavení

V nastavení je uživatel vyzván k zadání jména paměťové karty. To je potřeba zadat z toho důvodů, že existuje mnoho výrobců paměťových karet a každý ji pojmenuje podle svého systému. Pokud by bylo ve zdrojovém kódu napevno zadáno jméno paměťové karty, mohlo by dojít ke kolizi, protože program by nenašel paměťovou kartu s uživatelem zadaným jménem a nedošlo by k odemknutí položek v hlavním menu a vytvoření adresářové struktury.

Po zadání jména program testuje, zda je paměťová karta s tímto jménem přítomná v mobilním zařízení. Pokud ne, program oznámí, že nemohl najít paměťovou kartu a vyžádá si zadání nového jména.

V případě, že jméno paměťové karty souhlasí, otestuje, zda již na ni existuje adresářová struktura (viz. kapitola 3.2.2). Pokud adresářová struktura na paměťové kartě neexistuje, vytvoří ji a následně program odemkne všechny možnosti v hlavním Menu.

Dále se na obrazovce nachází okénko pro vložení IP adresy vzdáleného serveru, které sice není nutno zadat, ovšem pokud si bude uživatel přát připojit ke vzdálenému serveru pro odeslání zvukového záznamu nebo stažení log souboru, bude mu oznámeno, že před připojením je nutno IP adresu nastavit.

O úspěšném nastavení cesty k paměťové kartě a IP adresy vzdáleného serveru program informuje zobrazením zprávy „Parametres succesfully set“.

## 5.12 Server aplikace – Incoming files



Obrázek 5.12 Server aplikace pro přijímání záznamů

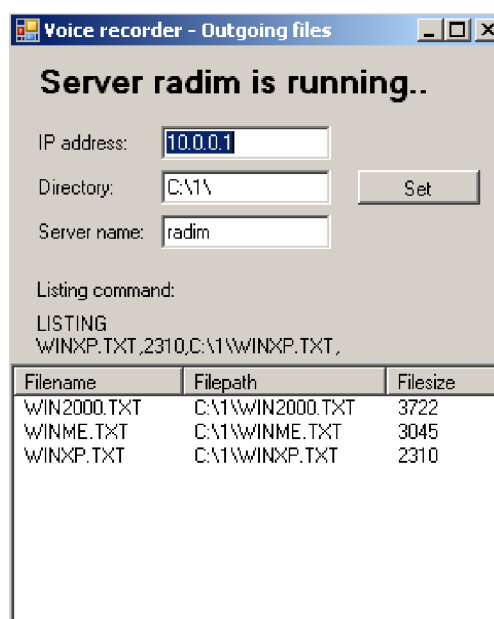
Tato část popisuje aplikaci, která běží na straně serveru a zajišťuje přijímání a ukládání zvukového záznamu na vzdáleném serveru.

Po startu aplikace začne naslouchat na portu 8091 na příchozí spojení. V okamžiku, kdy se klient prostřednictvím své aplikace připojí k serveru, proběhne mezi serverem a klientem výměna informací o odesílaném zvukovém záznamu a následně je proveden samotný přenos. Po přijetí zprávy je možno nastavit ve zdrojovém kódu programu zavolání aplikace pro rozpoznání řeči ve zvukovém záznamu a provést tak samotné rozpoznání v právě uloženém zvukovém záznamu.

Po dokončení přenosu je soubor uložen v adresáři, který si uživatel může změnit prostřednictvím zadání nové cesty do políčka Directory a potvrzením tlačítkem Set.

Při ukončení aplikace je naslouchání na portu přerušeno a port je uvolněn.

## 5.13 Server aplikace – Outgoing files



Obrázek 5.13 Server aplikace pro odeslání log souborů

Tato část popisuje aplikaci, která běží na straně serveru a zajišťuje odeslání log souboru, který vznikl zpracováním zvukového záznamu na serveru, uživateli mobilního zařízení.

Po startu aplikace začne naslouchat na portu 8090 na příchozí spojení. V okamžiku, kdy se klient prostřednictvím své aplikace připojí k serveru, proběhne mezi serverem a klientem výměna informací, kdy uživatel žádá aplikaci běžící na serveru o seznam dostupných log souborů ke stažení.

Server provede vyhledávání a odešle mobilnímu zařízení informaci o dostupných log souborech ke stažení. Následně si uživatel na svém mobilním zařízení vybere jeden z nich a zašle serveru žádost o stažení. Server žádost zpracuje a začne soubor odesílat.

Po úspěšném odeslání log souboru zašle server mobilnímu zařízení kontrolní zprávu, čím ověří, že mobilní zařízení soubor stáhlo a uložilo správně.

Uživatel mobilního zařízení si může stáhnout další záznam anebo se od vzdáleného serveru (této aplikace) odpojit.

Při ukončení aplikace je naslouchání na portu přerušeno a port je uvolněn.

## 6 Požadavky na mobilní zařízení

Program byl vytvořen pro mobilní zařízení, které používají operační systém Windows Mobile verze 6.0 a vyšší. Součástí systému je také Microsoft .NET Compact Framework v. 3.5 zajišťující podporu jazyka C#.

Mobilní zařízení musí obsahovat podporu GSM telefonování, jinak by program zůstal prakticky bez využití. Dále by měl mobilní zařízení pohánět procesor s taktem minimálně 200Mhz a zařízení by mělo disponovat operační pamětí RAM minimálně 64MB.

### 6.1 Instalace aplikace

Ke spuštění samotné aplikace zapotřebí uložit výsledný .exe soubor do adresářové struktury PDA. Tento .exe soubor aplikace je přiložen na CD ve složce se zdrojovými kódy Source\_code. Konkrétněji ve složce zdrojove\_kody\WM6.0\_aplikace\bakalari\bin\Debug\bakalari.exe. Poté již stačí kliknout na soubor aplikace, která se následně vyvolá.

Zkopírování .exe souboru do PDA zařízení se provádí pomocí aplikace ActiveSync, která je určena pro synchronizaci mobilního zařízení se stolním počítačem.

### 6.2 Testování aplikace

Zde jsou vypsány parametry PDA zařízení, na kterém byla aplikace testována.

**Typ:** XDA Orbit (změněný vzhled HTC P3300 Artemis se stejnými parametry)

**Procesor** – OMAP850, frekvence 201 MHz

**Paměť RAM** – 64 MB

**Paměť ROM** – 128 MB

**Rozlišení displeje** – 240x320 pixelů, 65 536 barev

**Operační systém** – Windows Mobile 6.0

Jednotlivé části aplikace byly testovány a odladovány v průběhu vývoje. Největší problémy byly zaznamenány při vývoji nahrávání telefonních hovorů, kdy program neplnil požadovanou funkci správně. Po vyřešení problému a dalším otestování již tato část pracovala bezchybně. Po dokončení vývoje všechny funkce aplikace běžely na PDA stabilně, bez potíží a aplikace plnila účel, pro který byla vytvořena.

## 7 Závěr

Cílem této bakalářské práce bylo vytvořit aplikaci pro záznam hovorů a aplikaci diktafon, která bude ukládat nahrávky na telefon. Tyto nahrávky budou v případě potřeby dále zpracovány základními algoritmy pro zpracování řeči a odeslány na server, kde by měly být rozpoznány. Výsledek bude odeslán zpět na telefon, tam uložen a použit pro vyhledávání v archivu telefonních hovorů.

Pro seznámení s teoretickými základy této problematiky byla nastudována řada literatury a informací z internetových stránek vývojářů společnosti Microsoft.

Na základě těchto informací bylo pro zpracování aplikace zvoleno vývojové prostředí Visual Studio 2008, které je pro programování aplikací pro Windows Mobile optimálním nástrojem.

Poté byla provedena dekompozice problému, analýza jednotlivých částí a návrh jejich řešení. Na základě těchto návrhů byla aplikace naprogramována, odladěna a testována na zařízení s Windows Mobile 6.0.

V průběhu vývoje aplikace jsem narazil na několik problémů, k jejichž vyřešení jsem využil informací z Microsoft Developer Network. Nejvýznamnější problém při tvorbě aplikace se vyskytl při implementování nahrávání hovorů, kdy program nereagoval na změnu systémového stavu tak, jak bylo požadováno. Tento problém byl vyřešen změnou principu řízení funkcí po změně systémových stavů.

Bylo odzkoušeno, že aplikace funguje, je stabilní a splňuje všechny parametry zadání. Navíc aplikace poskytuje rozšířené nastavení pro nahrávání hovorů a přehrávání pořízených záznamů.

Program komunikuje se vzdáleným serverem, který může využívat aplikaci pro rozpoznání řeči ve zvukových záznamech. Pro oboustrannou komunikaci byly navíc vytvořeny dvě aplikace, které běží na straně serveru. První zajišťuje přijímání zvukových záznamů ze strany mobilního zařízení a druhá se stará o odesílání výsledku rozpoznání záznamu v podobě log souboru mobilnímu zařízení. Samotná aplikace pro rozpoznání řeči ve zvukovém záznamu a její převod na text nebyla součástí zadání této bakalářské práce.

Cíl bakalářské práce byl splněn a aplikace je použitelná s kterýmkoli zařízením používající operační systém Windows Mobile 6.0 a vyšší.

Pro další vývoj této aplikace doporučuji zabývat se metodou ukládání záznamu telefonního hovoru tak, aby se záznam průběžně ukládal do souboru v průběhu telefonního hovoru a ne až po jeho ukončení, jak je tomu nyní, což může ale být s ohledem na omezené hardwarové možnosti mobilních zařízení složité. Dále doporučuji zabývat se rozšířením síťové komunikace v podobě zavedení přihlašování uživatele mobilního zařízení ke vzdálenému serveru pod jménem a heslem, které by zpřístupnilo jen ty log soubory zvukových záznamů, které daný uživatel vzdálenému serveru odeslal.

Při tvorbě bakalářské práce jsem nenavazoval na žádný předchozí projekt.

# Literatura

- [1] Wikipedia.: Personal Digital Assistant [online]. 2007, aktualizováno 2009-03-19 [cit. 2009-03-19]. Dostupné na URL: <[http://cs.wikipedia.org/wiki/Personal\\_Digital\\_Assistant](http://cs.wikipedia.org/wiki/Personal_Digital_Assistant)>
- [2] Skála, D.: PDA a mobilní zařízení pro práci a komunikaci online [online]. 10.8.2008, [cit. 25.3.2009]. Dostupné na URL: <<http://www.workline.cz/Pruvodce/PDA-a-mobilni-zarizeni.aspx/>>
- [3] Mikudík, R.: Z pravěku do přítomnosti: historie systému Windows Mobile [online]. 2007-12-14, aktualizováno 2007-12-14 [cit. 2009-03-25]. Dostupné na URL: <[http://palmare.idnes.cz/z-praveku-do-pritomnosti-historie-systemu-windows-mobile-pn9/tech-a-trendy.asp?c=A071126\\_121747\\_tech-a-trendy\\_ram](http://palmare.idnes.cz/z-praveku-do-pritomnosti-historie-systemu-windows-mobile-pn9/tech-a-trendy.asp?c=A071126_121747_tech-a-trendy_ram)>
- [4] Vokáč, L.: Jak putuje hovor aneb Co všechno se musí stát, abyste si zavolali [online]. 2008-07-18, aktualizováno 2008-07-18 [cit. 2009-04-02]. Dostupné na URL: <[http://mobil.idnes.cz/mob\\_tech.asp?c=A080715\\_125524\\_svet-mobilu\\_onphtml](http://mobil.idnes.cz/mob_tech.asp?c=A080715_125524_svet-mobilu_onphtml)>
- [5] eXEden: Úvod do programování pro Windows Mobile 2002 a 2003 - Vývojové nástroje [online]. 2003-09-18, aktualizováno 2003-09-18 [cit. 2009-04-10]. Dostupné na URL: <<http://www.ce4you.cz/articles/detail.asp?a=54>>
- [6] Sorcinelli, D.: Your Windows Mobile 5.0 Applications Can Monitor Clients and Respond to Change [online]. 2008-01-17, aktualizováno 2008-01-17 [cit. 2009-04-13]. Dostupné na URL: <<http://www.developer.com/ws/pc/article.php/3547381>>
- [7] Microsoft.com.: Microsoft Developer Network [online]. 2009, [cit. 2009-04-13]. Dostupné na URL: <<http://msdn.microsoft.com/en-us/default.aspx>>
- [8] Wikipedia: Microsoft .NET Framework [online]. 2008-02-02, aktualizováno 2009-05-04 [cit. 2009-04-14]. Dostupné na URL: <[http://en.wikipedia.org/wiki/Microsoft\\_.NET\\_Framework](http://en.wikipedia.org/wiki/Microsoft_.NET_Framework)>
- [9] Wigley, A., Wheelwright, S.: *Microsoft .NET Compact Framework*. Microsoft Press, Redmond, Washington, 2003. ISBN: 0-7356-1725-2
- [10] Václavík, J.: Perl (74) - Sockety [online]. 2008-11-18, [cit. 2009-04-15]. Dostupné na URL: <[http://www.linuxsoft.cz/article.php?id\\_article=1571](http://www.linuxsoft.cz/article.php?id_article=1571)>
- [11] Wikipedia: Windows Forms [online]. 2008-12-15, aktualizováno 2009-03-17 [cit. 2009-04-15]. Dostupné na URL: <[http://en.wikipedia.org/wiki/Windows\\_Forms](http://en.wikipedia.org/wiki/Windows_Forms)>
- [12] Microsoft, Windows Embedded Developer Center: Recording and Playing Sound with the Waveform Audio Interface [online]. 2004-01-01, [cit. 2009-04-16]. Dostupné na URL: <<http://msdn.microsoft.com/en-us/library/aa446573.aspx>>
- [13] Wikipedia: Microsoft Visual Studio [online]. 2008-03-14, aktualizováno 2009-04-24 [cit. 2009-05-01]. Dostupné na URL: <[http://cs.wikipedia.org/wiki/Microsoft\\_Visual\\_Studio](http://cs.wikipedia.org/wiki/Microsoft_Visual_Studio)>
- [14] CSHARPWORLD.COM: CLIENT / SERVER : Simple File Sharing and File Transfer [online]. 2006-06-02, [cit. 2009-05-02]. Dostupné na URL: <<http://www.csharpworld.com/archives2/archive335.html>>
- [15] Walder, L.: *Sledování stavu síťových serverů*. Bakalářská práce VŠB-TU Ostrava, 2007



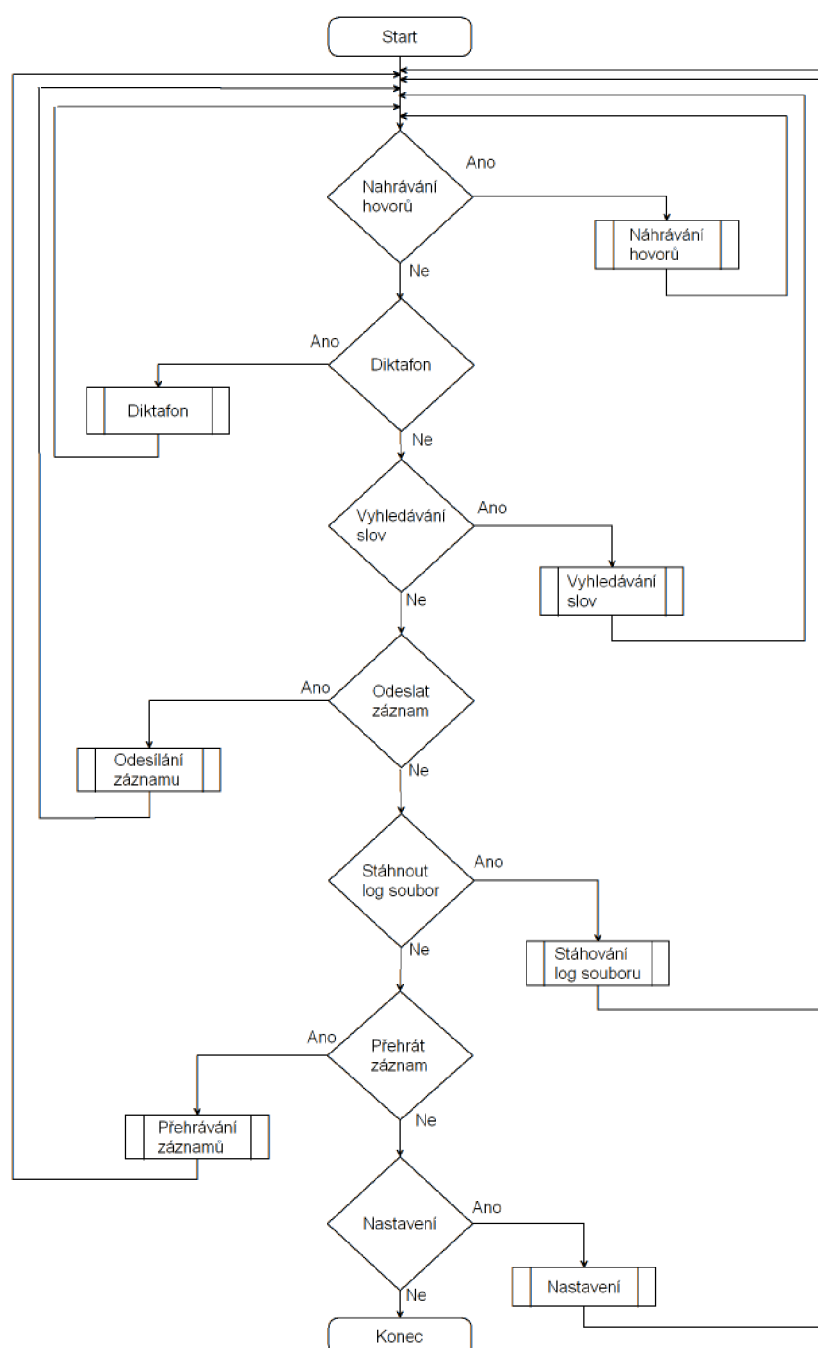
# Seznam příloh

Příloha 1. Vývojové diagramy

Příloha 2. CD obsahující zdrojové kódy aplikace, technickou zprávu, aplikaci v binární podobě

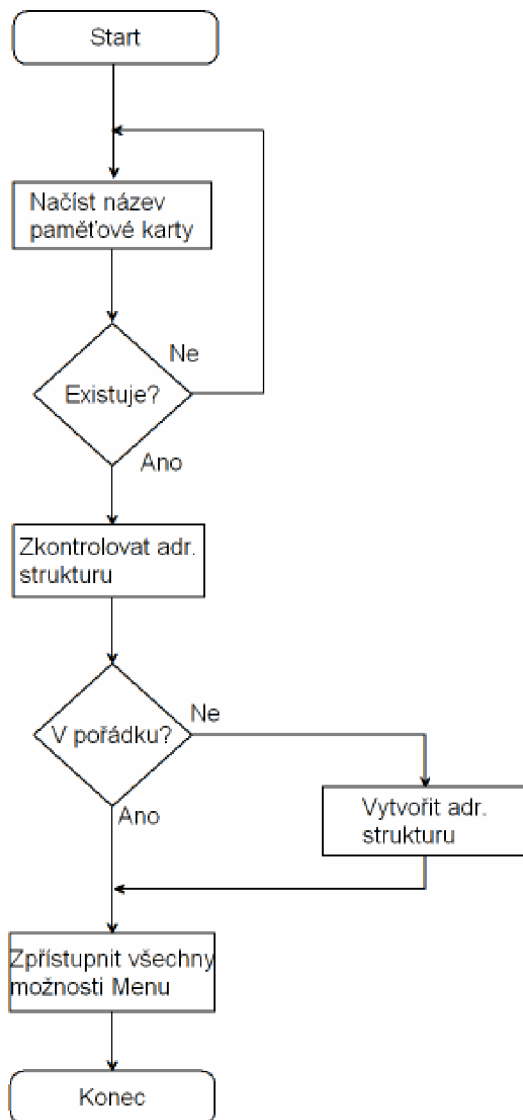
# Příloha 1 - Vývojové diagramy

## Hlavní menu



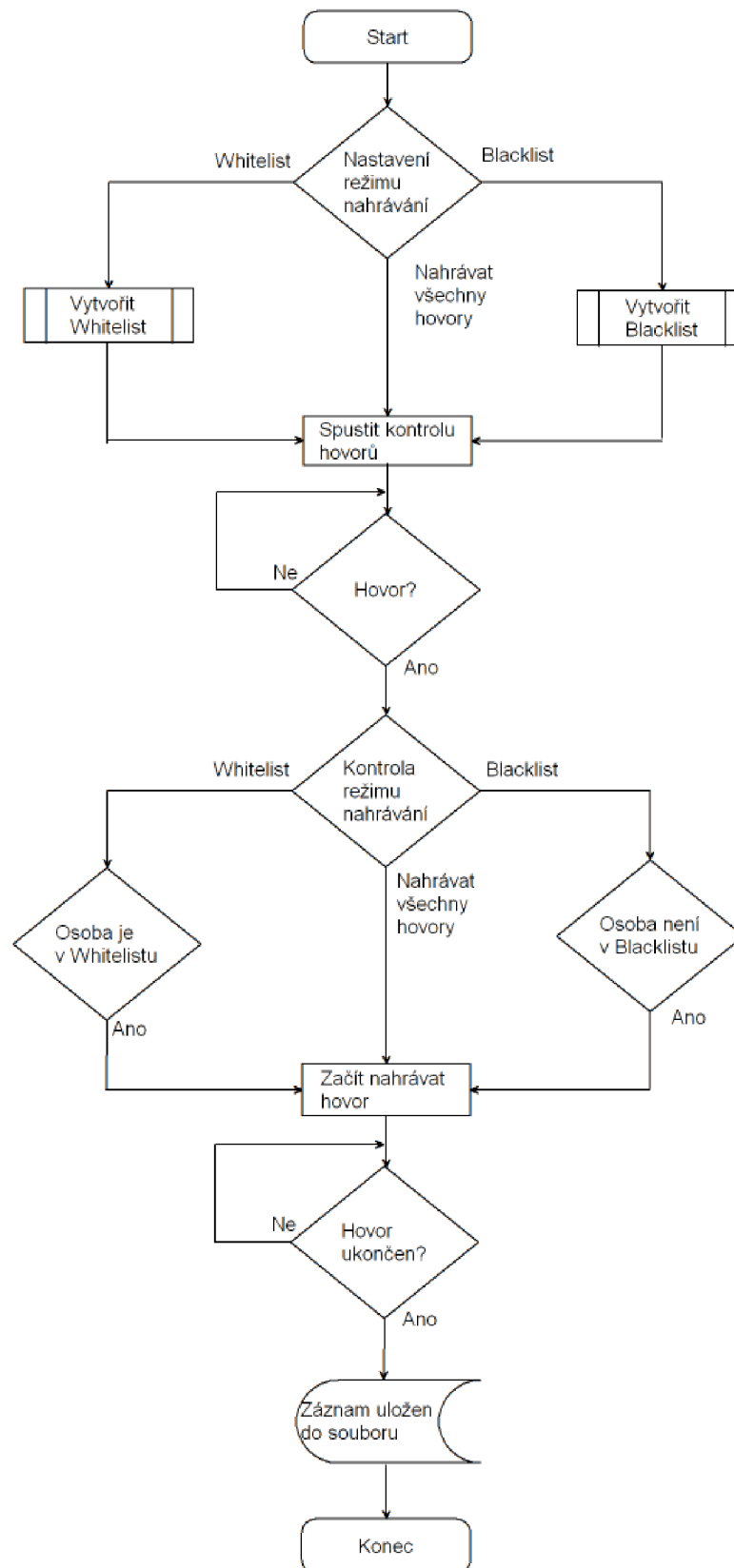
Tento diagram popisuje hlavní nabídku (hlavní menu), kde si uživatel vybírá, kterou část programu bude chtít dále použít.

## Nastavení



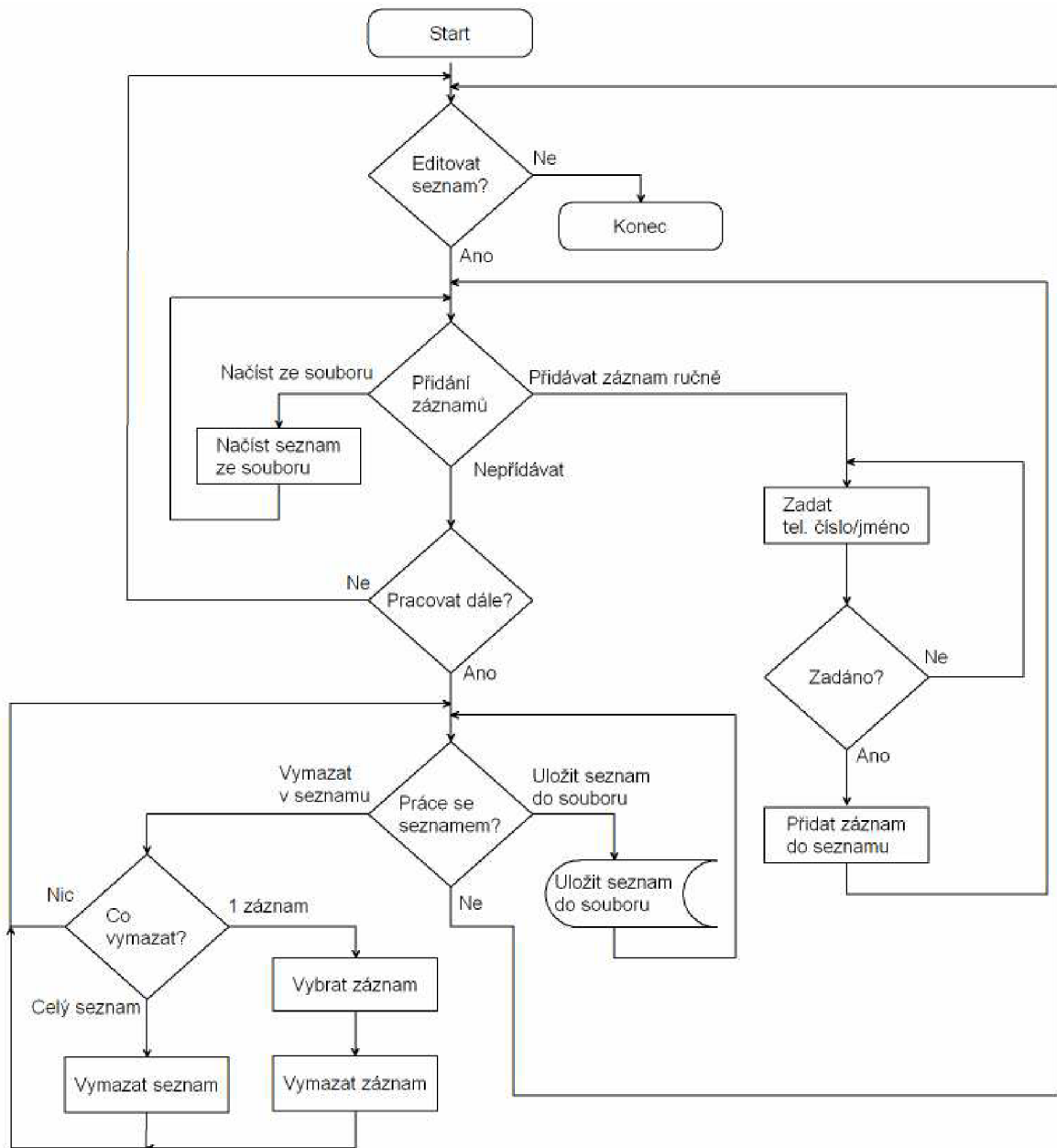
Tento diagram popisuje funkci nastavení, kdy program načte uživatelem zadaný název paměťové karty, otestuje, zda na ni existuje adresářová struktura a pokud ne, tak ji vytvoří. Následně odemkne všechny volby hlavní nabídky.

## Nahrávání telefonních hovorů



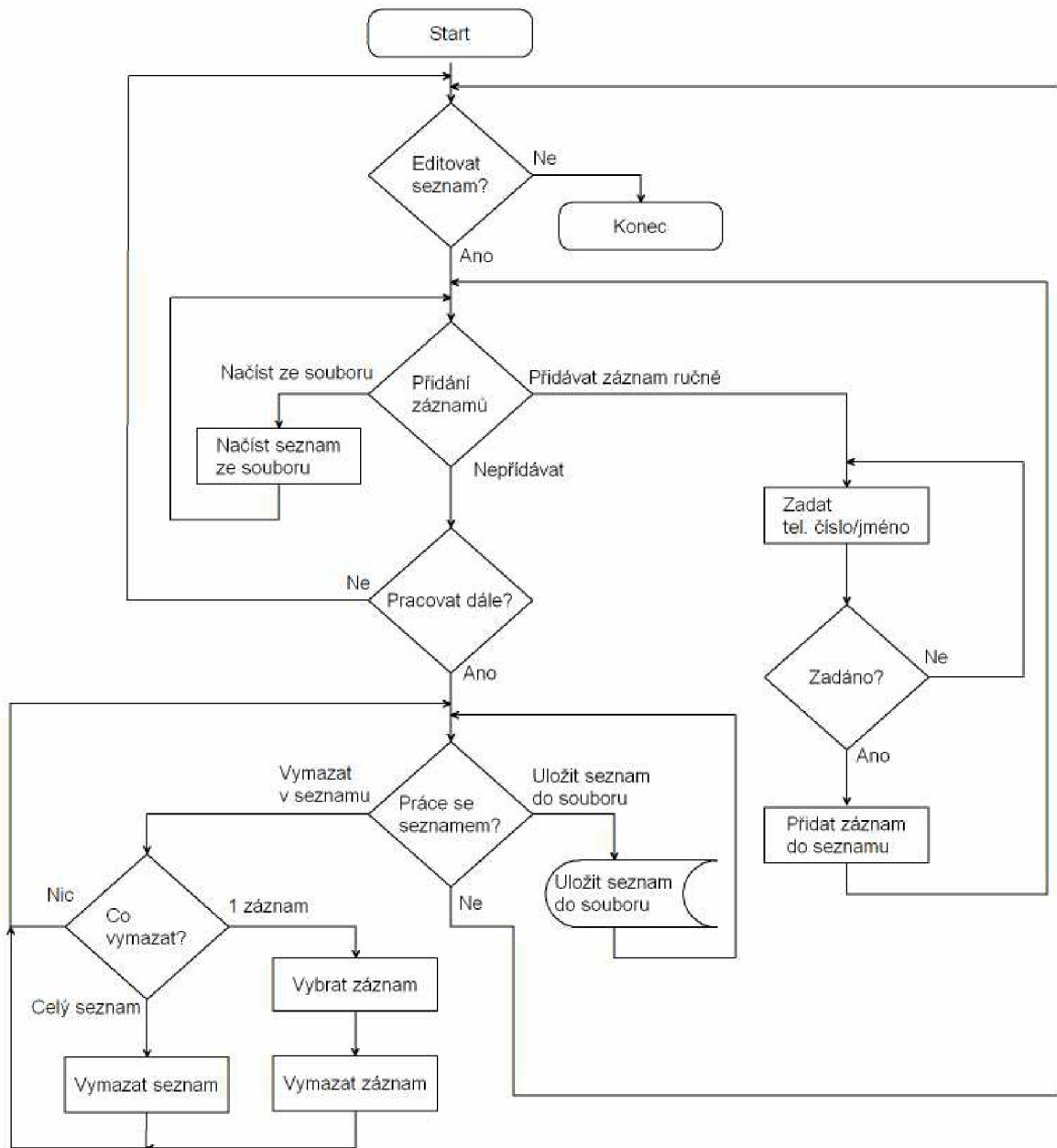
Tento diagram popisuje nahrávání telefonního hovoru s rozšířenou možností nastavení.

## Seznam zakázaných osob - Blacklist



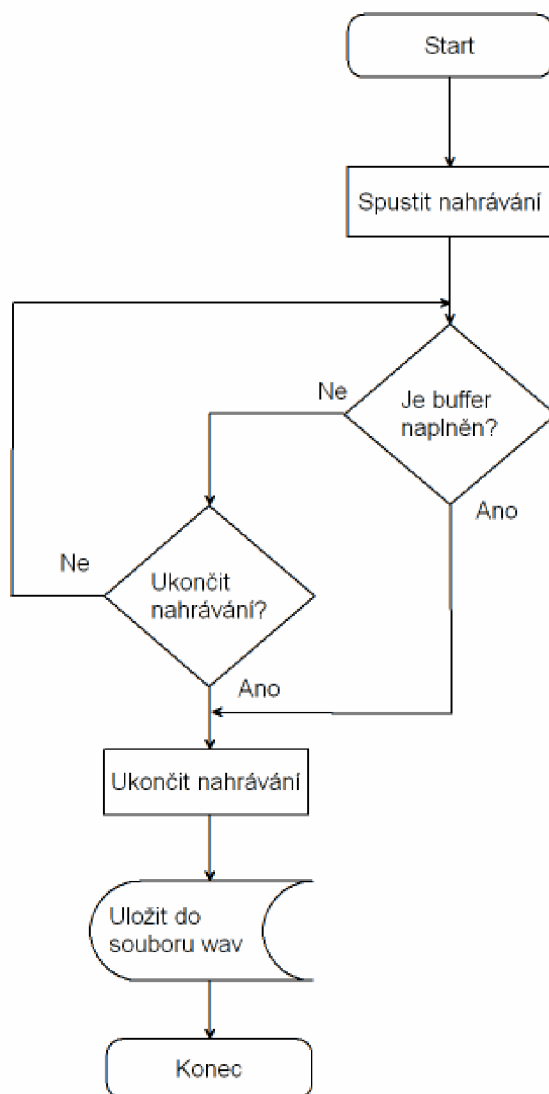
Tento vývojový diagram popisuje rozšířenou možnost nahrávání a to Blacklist, čili seznam zakázaných osob, od kterých se telefonní hovory nahrávat nebudou. Uživatel si může seznam editovat a to tak, že přidávat záznamy, mazat záznamy, ukládat seznam do souboru nebo seznam ze souboru načíst.

## Seznam povolených osob - Whitelist



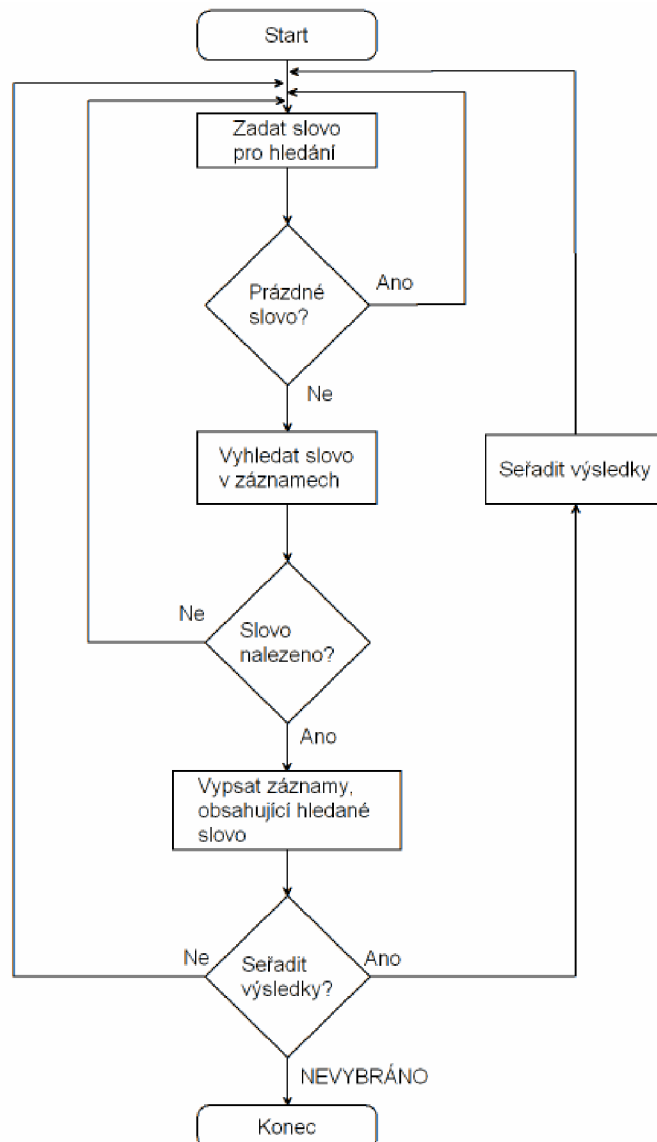
Tento vývojový diagram popisuje rozšířenou možnost nahrávání a to Whitelist, čili seznam osob, pouze od kterých se telefonní hovory budou nahrávat. Uživatel si může seznam editovat a to tak, že přidávat záznamy, mazat záznamy, ukládat seznam do souboru nebo seznam ze souboru načít.

## Diktafon



Tento diagram popisuje základní funkci diktafonu, kdy uživatel spustí nahrávání a může ho ukončit ručně nebo je nahrávání ukončeno po naplnění bufferu.

## Vyhledávání slov

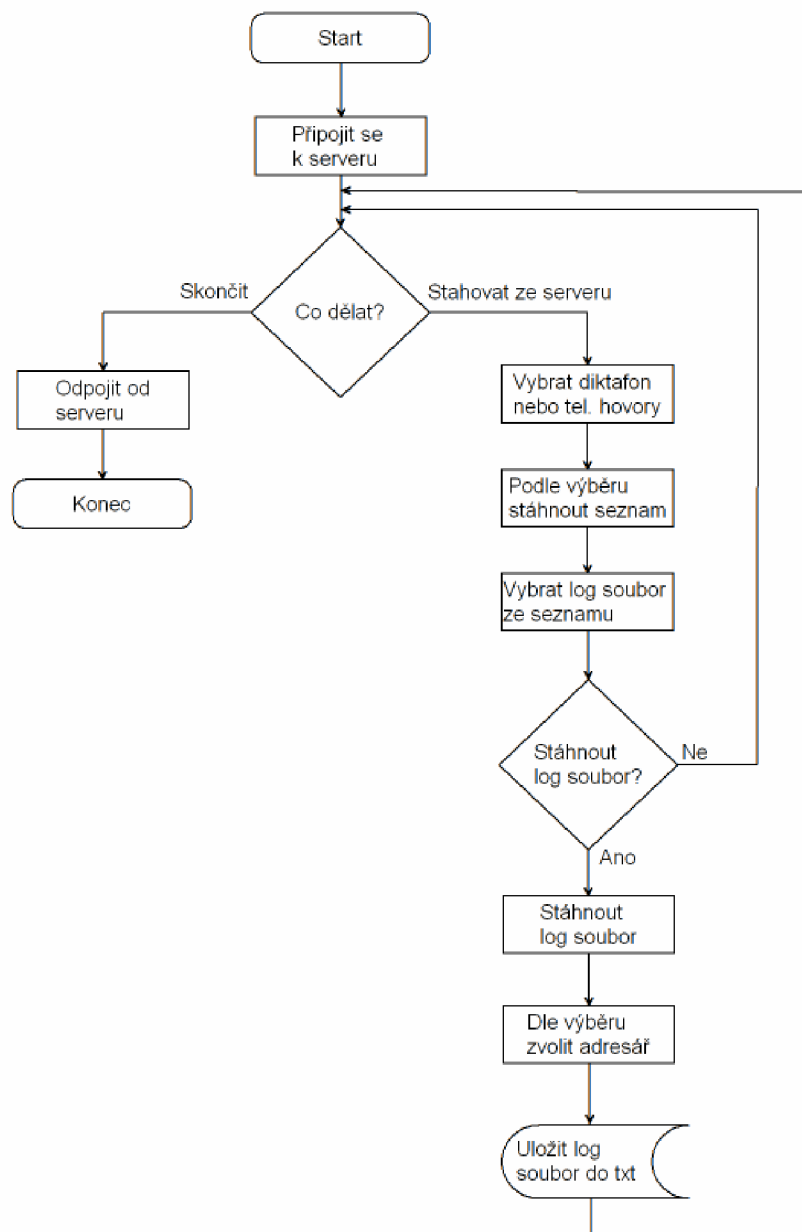


Tento diagram popisuje funkčnost části programu Vyhledávání slov, kdy uživatel hledá určité slovo v textových záznamech nahrávek a pokud je nalezeno, vypíše výsledky a má i možnost si výsledky seřadit.



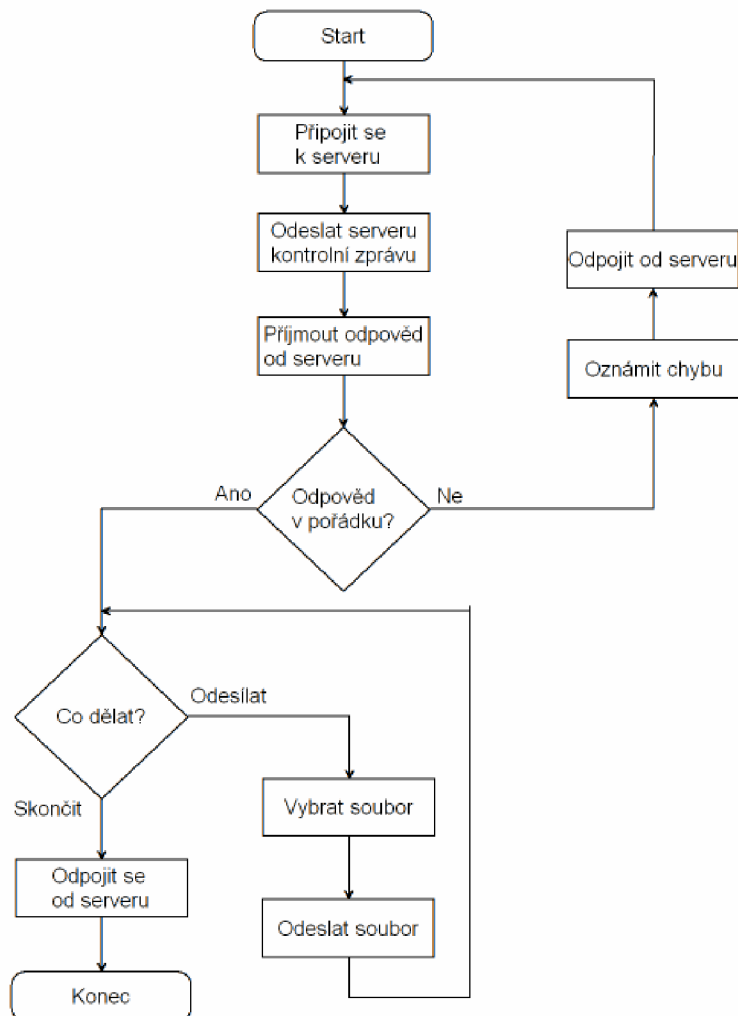


## Stáhnout log soubor ze vzdáleného serveru



Tento vývojový diagram popisuje funkci stažení log souboru ze vzdáleného serveru a jeho následné uložení do příslušného adresáře.

## Odeslat záznam na vzdálený server



Tento vývojový diagram popisuje funkci odesílání zvukového záznamu vzdálenému serveru.

## Příloha 2 – Obsah CD

Na přiloženém CD se nachází všechny zdrojové kódy projektu, technická zpráva v formátu PDF a uživatelská příručka ve formátu TXT.

V adresáři *zdrojove\_kody* se nachází zdrojové kódy k aplikaci pro nahrávání hovorů v mobilních zařízeních s operačním systémem Windows Mobile 6.0. Dále se v tomto adresáři nachází zdrojové kódy k aplikacím, které budou běžet na straně serveru.

V adresáři *exe* se nacházejí spustitelné soubory aplikace pro mobilní zařízení a spustitelné soubory aplikací, které poběží na straně serveru.

V adresáři *dokumenty* se nachází technická zpráva ve formátu PDF a uživatelská příručka ve formátu TXT.