

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

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

INFORMAČNÍ SYSTÉM PRO PODPORU
VINOHRADNICTVÍ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR LABOUNEK

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM PRO PODPORU VINOHRADNICTVÍ

INFORMATION SYSTEM FOR SUPPORT OF VINICULTURE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

PETR LABOUNEK

VEDOUCÍ PRÁCE
SUPERVISOR

ING. ZBYNĚK KŘIVKA, PH.D.

BRNO 2008

Abstrakt

Bakalářská práce se zabývá návrhem a implementací informačního systému s využitím mobilních technologií. Aplikace umožňuje synchronizaci s databázovým serverem a distribuci dat prostřednictvím webových služeb. Realizace byla zaměřena na mobilní zařízení s důrazem na požadavky vinaře, na uživatelsky přívětivé a snadno ovladatelné rozhraní a na synchronizaci dat. Pro implementaci byl zvolen programovací jazyk C#, SQL, PHP a HTML.

Klíčová slova

Informační systém, mobilní aplikace, PDA, .NET Framework, C#, SQL Server 2005, Visual Studio 2008

Abstract

The bachelor thesis describes design and implementation of information system using mobile technologies. The application allows synchronization with database server and data distribution via web services. Realization was focused on mobile device with emphasis on winemaker requests, "user-friendly" and "easy-to-use" user interface and data synchronization. Selected programming languages were C#, SQL, PHP and HTML.

Keywords

Information system, mobile application, PDA, .NET Framework, C# SQL Server 200, Visual Studio 2008

Citace

Labounek Petr: Informační systém pro podporu vinohradnictví. Brno, 2008, bakalářská práce, FIT VUT v Brně.

Informační systém pro podporu vinohradnictví

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením

Ing. Zbyňka Křivky, PH.D.

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

.....
Petr Labounek
12.5.2008

© Petr Labounek, 2008.

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
Úvod	3
1 Požadavky vinaře	4
1.1 Proces výroby vína	4
1.2 Požadavky na aplikaci	5
1.3 Objem zpracovávaných dat	5
1.4 Požadavky na uživatelské rozhraní	5
1.5 Zvyšování cukernatosti	6
1.5.1 Bílá vína	6
1.5.2 Červená vína	6
2 Použité mobilní technologie	7
2.1 Platformy a typy mobilních zařízení	7
2.1.1 PDA	7
2.2 Výběr technologie	8
2.3 Visual Studio 2008 a .NET Framework	9
2.4 ADO.NET	10
3 Možnosti datové synchronizace	12
3.1 Synchronizace a ActiveSync	12
3.2 Aplikace typu off-line	12
3.3 Replikace dat	13
3.3.1 Replikace souborů XML prostřednictvím ActiveSync	13
3.3.2 SQL CE replikace	13
3.3.3 Vzdálený přístup k databázi	14
4 Návrh aplikace	15
4.1 Aplikace v mobilním zařízení	15
4.1.1 Výběr programovacího jazyka	16
4.1.2 Jmenné prostory	16
4.1.3 Modální a nemedální okno	17
4.1.4 Vlákna	17
4.1.5 Sezóny a sklizně	17
4.1.6 Číselníky	18
4.1.7 Databáze	18
4.1.8 Připojení k databázi a čtení dat	19
4.1.9 Komponenty	20

4.1.10	Popis tříd.....	20
4.1.11	Struktura tříd.....	21
4.1.12	Popis činnosti výsledné aplikace	22
4.2	Aplikace v počítači.....	22
4.2.1	Databáze.....	22
4.2.2	PHP	23
4.2.3	Připojení k databázovému serveru	23
5	Problémy při řešení	24
5.1	Uvolňování paměti v PDA	24
5.2	Konfigurace SQL Serveru	24
6	Testování.....	26
6.1	Použitelnost v reálných podmínkách.....	26
7	Návrh na rozšíření.....	27
7.1	Mobilní zařízení	27
7.2	Stolní aplikace	27
7.3	Internetový obchod.....	27
8	Závěr	28
	Literatura	29
	Seznam příloh	30
	Příloha 1 – Instalace a spuštění aplikace	31
	Příloha 2 – Manuál k aplikaci na PDA	32

Úvod

Využití mobilních zařízení pro práci s daty v terénu se stále více rozšiřuje. Ať už se jedná o mobilní telefony nebo o PDA (*personal digital assistant*), usnadňují práci a zaznamenávání dat ve chvílích, kdy je použití stolního počítače nebo notebooku obtížné nebo nemožné. Pominu-li tradiční využití těchto přístrojů, setkáváme se s využitím v restauracích v podobě systému, který číšník využívá k zadávání objednávek, ve skladech pomáhá při inventarizaci zboží nebo třeba ve vlacích při nákupu jízdenek.

Cílem této bakalářské práce je seznámit se s požadavky amatérského vinaře nebo malovýrobce vína na informační systém a navrhnout použití vhodných mobilních řešení. Právě vinař je dalším z adeptů na využití mobilních zařízení pro udržování přehledu o nádobách, ve kterých uchovává víno. Výsledný systém by měl umožnit práci v terénu bez připojení k počítačové síti a následnou synchronizaci s databázovým serverem běžícím na stolním počítači. Tento server se pak stává zdrojem dat pro webové stránky, které uživatelům umožňují prohlížení údajů a tisk sestav.

V první kapitole jsou uvedeny jednotlivé kroky, kterými víno projde v průběhu výroby. Dále jsou shrnuty požadavky vinaře na informační systém, které jsem nastudoval z různých publikací nebo je konzultoval přímo s vinařem.

Druhá kapitola pojednává o použitých mobilních technologiích a důvody jejich volby. Popsán je .NET Framework, Visual Studio a prostředky pro práci s databází. Následující kapitola je věnována možnostem datové synchronizace mobilního zařízení se stolním počítačem a popisuje princip replikace databází.

Čtvrtá kapitola se zabývá návrhem aplikace nejprve z pohledu mobilního zařízení a poté z pohledu stolního počítače. Na začátku jsou uvedeny důvody výběru programovacího jazyka C# a jeho klíčové vlastnosti. Následuje návrh uživatelského rozhraní s důrazem na potřeby vinaře, návrh databáze, uvedení komponent využívaných pro zobrazování informací a popis činnosti výsledné aplikace. Další část pojednává o aplikaci v počítači a o použitých technologiích.

Problémy, které bylo v průběhu vývoje nutné řešit jsou uvedeny v kapitole pět. Jedná se především o práci s pamětí a o konfiguraci databázového serveru. Následující kapitola pojednává o testování aplikace v průběhu vývoje a hodnotí její použitelnost v reálných podmínkách. V předposlední kapitole navrhuji možnosti dalšího vývoje.

1 Požadavky vinaře

Prvním krokem v návrhu této aplikace je seznámení s potřebami amatérského vinaře na informační systém. Důležité je získat informace o procesu zpracování vína a rozsah údajů, které je potřeba zaznamenávat. K tomuto účelu jsem si udělal malý výlet po jižní Moravě, během kterého jsem navštívil malovýrobce i velkovýrobce vína. Získané znalosti mi umožnily lépe porozumět potřebám vinaře a vhodně navrhnout požadovanou aplikaci.

1.1 Proces výroby vína

Aplikace je zaměřená na tu část procesu výroby vína, která probíhá po sklizni ve vinném sklepě. Následuje přehled hlavních kroků při výrobě bílého vína (více viz [13]).

- Surovina - základní předpoklad pro výrobu kvalitních vín tvoří zdravé hrozny. Hrozny jsou zpracovány odděleně dle jednotlivých odrůd a měly by být zpracovány co nejdříve po sklizni. Nejen odrůda, stanovištní podmínky a samotná vyzrálost suroviny dávají předpoklady k výrobě kvalitních vín. Na konečný produkt má samozřejmě vliv použitá výrobní technologie.
- Odzrnění – oddělení pevných částí hroznu (třapiny a dřevité části) od bobulí. Hrozny se nejprve zpracují na hroznovou drť a poté dojde k odzrnění.
- Lisování – technologický proces, při kterém se v lisu získává tlakem mošt ze rmutu nebo z celých hroznů. Lisováním se odděluje mošt od pevných částí. Pevné části po vylisování nazýváme matoliny.
- Úprava moštu před kvašením – nejčastěji odkalování (odstranění nečistot), síření (zabránění působení škodlivých mikroorganismů) a přiměřené provzdušnění.
- Alkoholové kvašení (fermentace) – biochemický proces, při kterém jsou cukry hroznů (glukosa, fruktosa) přeměňovány na alkohol za přítomnosti kvasinek.
- Stáčení vína – po usazení sedimentů se víno stáčí do jiné nádoby. Dlouhodobé ležení na kalech může způsobit zhoršení jakosti vína.
- Školení vína – všechny operace provedené sklepmistrem od stočení z prvních kalů do lahvování - číření, filtrace (mechanické nebo chemické odstranění zákalů ve víně) a mnoho dalších.
- Stabilizace vína – fáze úpravy vína před lahvováním. Provádíme likvidaci nežádoucích mikroorganismů, které by mohly negativně ovlivnit víno po lahvování. Stabilitu vína dnes ovlivňujeme chladem, teplem a používáním oxidu siřičitého.
- Stáčení vína.
- Zrání vína v dřevěných sudech, lahvích nebo skleněných demižonech.

1.2 Požadavky na aplikaci

Sklizeň a následné zpracování vína probíhá každou sezónu, respektive každé vegetační období, které obvykle odpovídá jednomu roku. Vinař může pěstovat více odrůd vína, proto je nutné to v rámci každé sezóny rozlišit. Aplikace musí být schopna zaznamenávat všechny výše uvedené kroky, kterými víno projde. Musí být zaznamenáváno, kdy bylo víno přelito, do jakých lahví, jaký byl jeho objem a vlastnosti. V průběhu zrání je také nutno víno kontrolovat a uchovávat naměřené hodnoty.

Užitečnou pomůckou jsou také různé kalkulátory – odhad optimálního využití nádob podle předběžného odhadu výnosu, výpočet cukru potřebného k dosazení vína na požadovanou cukernatost apod.

Sledované vlastnosti vína:

- Druh vína, odrůda
- Přívlastek
- Množství cukru, množství alkoholu
- Vizuelní, chuťový a čichový dojem vína

1.3 Objem zpracovávaných dat

Po prostudování na internetu dostupného katalogu menších výrobců vína jsem došel k následujícím poznatkům, které budu brát v úvahu především při návrhu a vytváření databáze.

- Malovýrobce vyrobí každoročně 5000-10000 lahví vína
- Víno je při výrobě přibližně 7x přeléváno

1.4 Požadavky na uživatelské rozhraní

Aplikace, která vinařovi zjednoduší ukládání i získávání informací o svém vinném sklepě přímo v terénu, musí být snadno ovládatelná. Vzhledem k tomu, že dostupná mobilní zařízení většinou nemají klávesnici a mají relativně malý displej, je nutné navrhnout aplikaci tak, aby uživatel mohl co nejvíce využívat přednastavených číselníků. Příkladem může být zadávání odrůdy vína – uživatel si ji jednoduše vybere ze seznamu, odpadá tedy pracné vyplňování, realizované většinou pomocí malé klávesnice dostupné přes dotykový displej přístroje.

Vhodným rozšířením je také synchronizace se stolním počítačem nebo notebookem, na kterém si uživatel může pohodně prohlížet nebo tisknout rozsáhlé tabulky, popřípadě data zpřístupnit prostřednictvím internetu.

1.5 Zvyšování cukernatosti

Cukernatost je podle našeho vinařského zákona nejdůležitějším parametrem pro klasifikaci vín. Zároveň jde o nejjednodušeji měřitelný kvalitativní parametr, který můžeme zjišťovat již na vinici pomocí refraktometru a poté po vylisování hroznů ve vinném sklepě pomocí moštoměru. Hodnoty na stupnici neboli „stupně normalizovaného moštoměru“, ve zkratce °NM, uvádějí, jaké množství cukru je obsaženo ve 100 litrech moštu při teplotě 15 °C. [1]

Příjemným rozšířením funkcí programu je kalkulátor, který po zadání všech hodnot vypočítá množství cukru potřebné k doslazení.

1.5.1 Bílá vína

U bílých vín používáme k výpočtu potřeby přidání řepného cukru do moštu poměrně jednoduchý postup.

Potřeba řepného cukru (v kg) = počet °NM, o který je třeba cukernatost zvýšit × objem moštu (v hl) × 1,1 kg cukru

1.5.2 Červená vína

Při doslazování rmutu modrých odrůd je třeba zohlednit podíl moštu ve rmutu modrých odrůd révy vinné. U odrůd s velkými bobulemi používáme ve vzorci koeficient 0,85–0,90, u odrůd s malými bobulemi potom používáme přepočítávací koeficient 0,80.

Potřeba řepného cukru (v kg) = počet °NM, o který je třeba cukernatost zvýšit × objem moštu (v hl) × 1,1 kg cukru × koeficient

2 Použití mobilní technologie

Součástí práce je i výběr vhodných mobilních technologií a vývojových prostředí. V současné době lze vybírat z několika technologií.

2.1 Platformy a typy mobilních zařízení

Windows Mobile je operační systém spojený se základními aplikacemi pro mobilní zařízení založený na Win32 API z Microsoft Windows. Jeho produkce začala s nástupem mobilních zařízení přibližně v polovině 90. let. Nyní se nachází ve verzi 6. Konkurenčními produkty jsou PalmOs a Symbian, které jsou určeny pro tzv. smartphony – chytré telefony s operačním systémem.

Windows Mobile má velmi silné postavení na trhu, používá se především v PDA a chytrých telefonech (viz obrázek 2-1). Podpora ze strany Microsoftu ho staví z mého pohledu jako nejlepšího kandidáta pro zadanou mobilní aplikaci.



Obrázek 2-1: Mobilní zařízení – PDA, smartphone

2.1.1 PDA

PDA (*personal digital assistant* – osobní digitální pomocník) je malý kapesní počítač určený původně k organizování času a kontaktů. V posledních letech je díky vyšší technologické vyspělosti využíván také pro přehrávání hudby a filmů, obsahuje nástroje pro zpracování a prohlížení textu, umožňuje synchronizaci a připojení k internetu pomocí běžně dostupných bezdrátových technologií (Bluetooth, Wi-fi, IrDA, GSM) nebo prostřednictvím USB kabelu. Stává se tak osobním společníkem, který může sloužit manažerovi místo klasického diáře, prostřednictvím kterého se může připojit k internetu a stáhnout si e-maily nebo využít vestavěného mobilního prohlížeče webových stránek pro brouzdání

po internetu. Může také sloužit číšníkovi pro zadávání objednávek nebo například skladníkovi pro inventarizaci zboží. Díky dostupnosti GPS modulů lze PDA využít i jako navigaci.

Některé vlastnosti PDA, které jsou významné pro vytvářenou aplikaci:

- Malé rozměry, displej o rozlišení nejčastěji 320x240 (QVGA) nebo 640x480 (VGA)
- Možnost připojení k počítači – bezdrátově nebo pomocí USB kabelu
- Vestavěná paměť, slot pro paměťovou kartu
- Dotykový displej
- Nemá plnohodnotnou klávesnici, text lze zadávat pomocí systému pro rozpoznávání písma nebo pomocí virtuální klávesnice, která se zobrazí na části displeje.
- Operační systém, pro který lze programovat aplikace
- Rychlost procesoru a paměť je vhodná pro menší aplikace

2.2 Výběr technologie

Pro vývoj výsledné mobilní aplikace jsem se rozhodoval mezi dvěma technologiemi – Java 2 Micro Edition (*J2ME*) a .NET Compact Framework. První z uvedených je hojně využívána v mobilních telefonech a dalších komunikačních zařízeních. Vzhledem k tomu, že výsledná aplikace by měla běžet na mobilním zařízení s operačním systémem Microsoft Windows Mobile a je nutno uvažovat synchronizaci s databází na straně počítače, představuje .NET technologie lepší volbu. K tomu přispívá i fakt, že tuto platformu podporuje Microsoft Visual Studio, které se v poslední verzi 2008 pyšní perfektní podporou mobilních zařízení. Některé vlastnosti obou technologií, mezi kterými jsem se rozhodoval, jsem zapsal do tabulky (viz tabulka 2-1).

	.NET Compact Framework	J2ME
Virtuální stroj	Common Language Runtime (CLR)	Java Virtual Machina (JVM)
Přenositelný kód	Intermediate Language (IL)	Mezikód nezávislý na architektuře
Just-In-Time (JIT) Compiling	ANO	ANO
Garbage Collection	ANO	ANO
Programovací jazyky	C#, J#, Visual Basic .NET	Java
Platformy	Pocket PC, Windows Mobile	Všechny mobilní platformy
Vývojové nástroje	Visual Studio 2005, 2008	JDeveloper, Netbeans, Eclipse

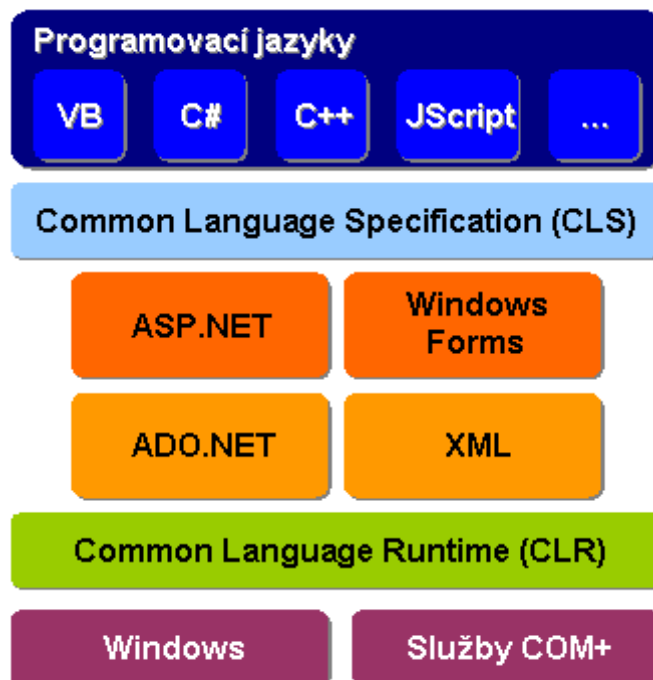
API databáze	ADO.NET	JDBC
Mobilní databáze	SQL Server CE	IBM DB2 Everyplace, iAnywhere Solutions, Oracle9i Lite
Vzdálená databáze	Kompatibilní s ADO.NET	Kompatibilní s JDBC

Tabulka 2-1: Porovnání .NET CF a J2ME

2.3 Visual Studio 2008 a .NET Framework

.NET je softwarová platforma vyvinutá společností Microsoft, která umožňuje vývoj a spuštění aplikací a webových služeb. První verzi vydal Microsoft v únoru 2002, následovaly verze 1.1, 2.0, 3.0 a v současné době aktuální verze 3.5.

Platforma .NET Framework (viz obrázek 2-2) se skládá ze dvou částí – společné jazykové prostředí pro běh aplikací (*CLR – Common Language Runtime*) a knihovna Framework Class Library (*FCL*). Součástí běhového prostředí je pro oznamování chyb vytvořen mechanismus výjimek, všechny jazyky tedy také oznamují chyby prostřednictvím tohoto mechanismu, stejně tak běhové prostředí umožňuje vytvářet vlákna, takže i všechny jazyky podporující CLR to také umožňují.



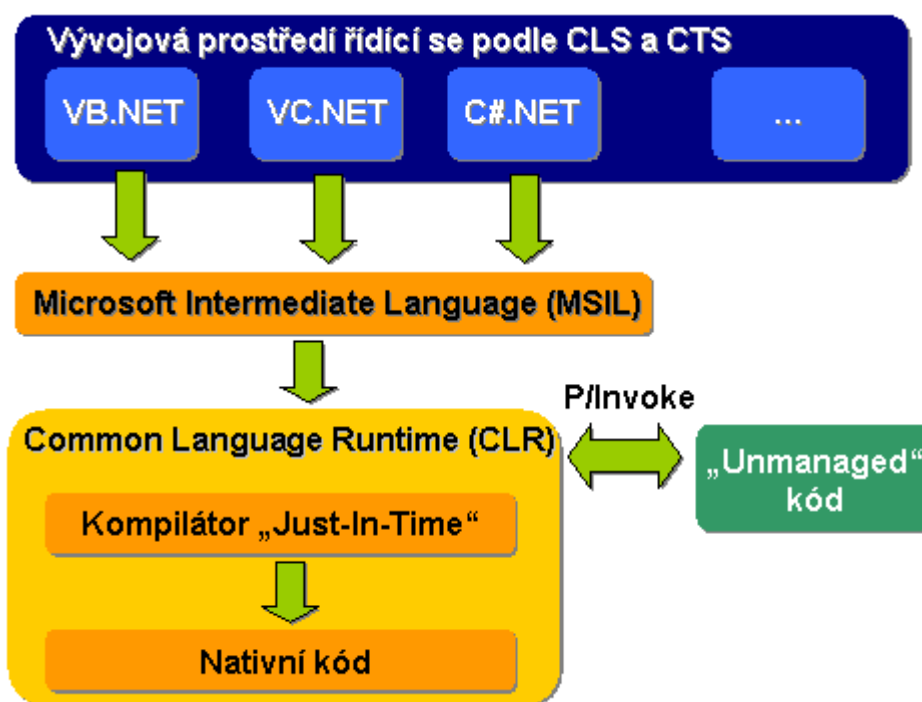
Obrázek 2-2: .NET Framework [10]

O platformě Microsoft .NET by se dalo říci, že je mezivrstvou mezi aplikací a operačním systémem. Program napsaný pro tuto platformu se nejprve zkompile do tzv. mezikódu – MSIL

(*Microsoft Intermediate Language*). Bez ohledu na použitý překladač (pro jazyk C#, Visual Basic, J# apod.) je výsledkem vždy tzv. řízený modul (viz obrázek 2-3). Řízený modul představuje standardní přenositelný a spustitelný soubor Windows, jenž ke svému spuštění vyžaduje CLR. V okamžiku spuštění souboru na klientském počítači .NET Framework kód zkompiluje do nativního kódu a spustí ho. Nezáleží tedy v jakém jazyce je program napsán, příslušný kompilátor přeloží zdrojový kód vždy do stejného jazyka – MSIL.

Pro běh aplikací je potřeba mít nainstalovaný .NET Framework příslušné verze. Existuje jednak verze pro osobní počítač (instalátor má asi 20 MB), jednak verze pro mobilní zařízení s Windows Mobile (asi 2 MB), která obsahuje asi čtvrtinu funkcí. Nejnovější verze operačního systému od Microsoftu (Windows Vista) již .NET Framework standardně obsahuje.

Pro vývoj .NET aplikací vydal Microsoft Visual Studio, které je nyní ve verzi 2008. Je to balík softwarových produktů, nástrojů a technologií určených pro vývoj serverových, stolních, mobilních i webových aplikací na platformách Windows.



Obrázek 2-3: Integrace vývojových prostředí [10]

2.4 ADO.NET

Databáze je organizovaná skupina informací rozdělená do tabulek. Každá tabulka pak obsahuje řádky a sloupce; tyto sloupce uchovávají informace. K databázi máme přístup pomocí jazyka SQL (*Structured Query Language* – strukturovaný dotazovací jazyk). Ten se stal standardem podporovaným většinou databázových systémů. [4]

Pro přístup k údajům v databázi máme na výběr z několika technologií. Nejnovější, nejvýkonnější a v dnešní době podporovaná je technologie nazývaná ADO.NET. Ze starších můžeme uvést například ODBC (*Open Database Connectivity*), OLE DB nebo ADO (*ActiveX Data Objects*).

ADO.NET je soubor typů (tříd, rozhraní, výčetů, ...), které umožňují přístup k datům v různých datových zdrojích z aplikací .NET Framework. Hlavními výhodami jsou možnost odpojené práce s údaji, kdy je použita vyrovnávací paměť pro zrychlení opakovaného přístupu k údajům, relační programovací model, práce s heterogenními zdroji dat, nahrávání a ukládání údajů ve formátu XML, nahrávání a ukládání relačních schémat a umožnění svázání dat s uživatelským rozhraním (*data binding*). [2]

Při vytváření aplikací komunikujícím s datovým zdrojem lze použít dva modely pro přístup k datům. Jedním je tzv. „připojený model“, kdy je aplikace stále připojena (on-line), druhým je tzv. „odpojený model“, kdy je aplikace většinu času od datového zdroje odpojena (off-line).

Přístup k datům v různých datových zdrojích je zajištěn použitím určitých rozhraní (*IDbConnection, IDbCommand, ...*). Dále jsou k dispozici jednotliví zprostředkovatelé přístupu ke konkrétnímu datovému zdroji, pro který je zprostředkovatel (ADO.NET Data Providers) implementován. Použitím tohoto přístupu je dosaženo, že je možné psát kód, který bude fungovat pro jakýkoliv datový zdroj (Microsoft SQL, Oracle, ...). [8]

Příklad ADO.NET data providerů:

- Databáze Microsoft SQL Server – jmenný prostor *System.Data.SqlClient*
- Databáze Microsoft SQL CE – jmenný prostor *System.Data.SqlCeServer*
- ODBC datové zdroje – jmenný prostor *System.Data.Odbc*
- OLE DB datové zdroje – jmenný prostor *System.Data.OleDb*
- Databáze Oracle – jmenný prostor *System.Data.Oracle*

3 Možnosti datové synchronizace

Mobilní zařízení může pracovat buď v režimu on-line, kdy má k dispozici trvalé připojení k síti (například prostřednictvím Wi-fi) a data tak mohou být ukládána na vzdálený počítač, nebo v režimu off-line, kdy jsou data ukládána do mobilního zařízení. Po připojení k počítači pak může dojít k synchronizaci dat.

3.1 Synchronizace a ActiveSync

Synchronizace je proces, který se postará o to, aby byla na počítači stejná data jako v mobilním zařízení. Typicky se používá pro udržování aktuálního seznamu kontaktů, schůzek, e-mailů a dalších osobních informací na obou zařízeních. Existují automatické nástroje, které tyto činnosti provádějí za uživatele. Jedním z takových programů je Microsoft ActiveSync.

ActiveSync je mocný nástroj, který automaticky při každém opětovném připojení zařízení s Windows Mobile zajišťuje synchronizaci zvolených dat v obou směrech (PC – PDA a naopak). Zároveň umožňuje přenos souborů a v okamžiku, kdy je mobilní zařízení připojeno přes USB kabel získává také IP adresu. Je tedy možné sdílet připojení k internetu a komunikovat mezi zařízeními prostřednictvím protokolu TCP/IP.

3.2 Aplikace typu off-line

V zadání bakalářské práce je přímo uvedeno, že výsledná aplikace má pracovat v režimu off-line. Proto se budu detailně zabývat pouze tímto typem aplikace.

V aplikaci typu off-line se předpokládá autonomní způsob práce, pouze s občasným připojením do sítě. Připojení může být realizováno buď pomocí datového přenosu GSM, nebo je mobilní zařízení občas připojeno do sítě. Off-line aplikační režim se hodí pro sběr údajů v terénu, různé ankety, průzkumy apod. [2]

Při návrhu koncepce databázové aplikace můžeme použít několik aplikačních scénářů. Některé z nich se pokusím popsat v následujících kapitolách.

- Replikaci XML souborů pomocí technologie ActiveSync
- ADO.NET Data Provider
- Využití XML webové služby
- SQL CE replikace
- Přístup k SQL Serveru prostřednictvím HTTP

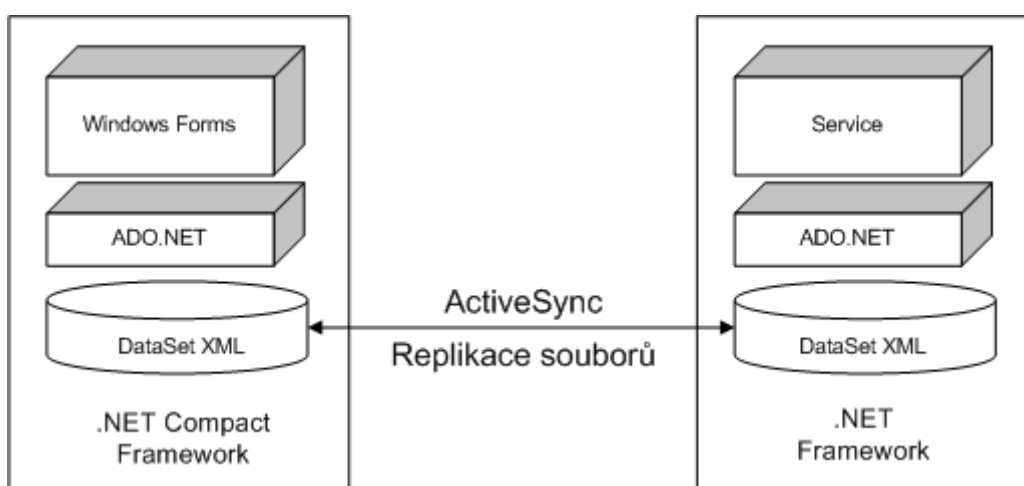
3.3 Replikace dat

Replikace je proces vytváření a správy kopie databázových objektů v distribuovaném databázovém systému. Uživatelé mohou například využít lokální databázi místo vzdáleného serveru a minimalizovat tak síťový provoz.

3.3.1 Replikace souborů XML prostřednictvím ActiveSync

Použití tohoto způsobu replikace je poměrně časté. Setkáme se s ním například při synchronizaci souborů, na kterých uživatelé pracují, při přenášeni elektronických knih pro Microsoft Reader nebo při synchronizaci údajů mezi aplikací Outlook a Pocket Outlook. Výhodou je možnost práce off-line, nevýhodou je možnost současného připojení pouze jednoho mobilního zařízení k jednomu počítači.

Pro funkčnost replikace je nutné nastavit složky a soubory XML, jejichž obsah má být synchronizován. Můžeme nastavit také dominanci jednoho zařízení vůči druhému.

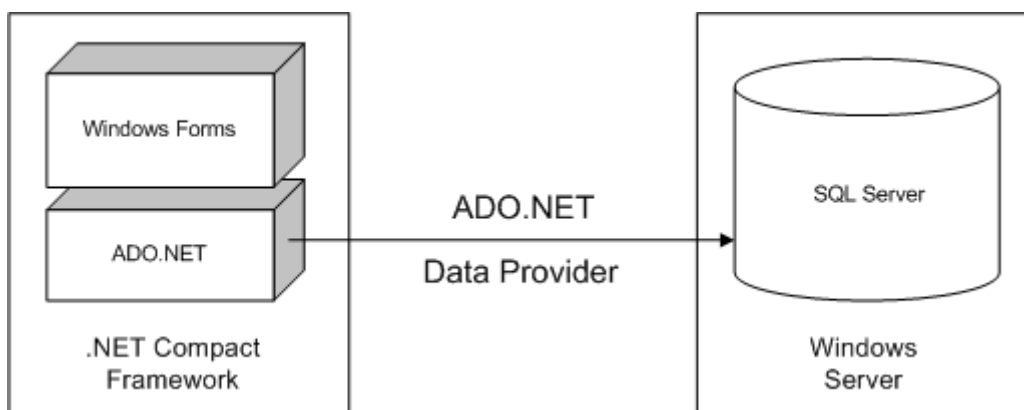


Obrázek 3-1: Replikace souborů pomocí ActiveSync

3.3.2 SQL CE replikace

Microsoft SQL podporuje „merge“ replikaci s Microsoft SQL CE, která se hodí zejména tehdy, kdy má více uživatelů paralelně pracovat s daty v odpojeném režimu. Uživatelé se pak k serveru připojí, doručí svá data na server a ten jim vrátí změny v datech provedené ostatními uživateli. Synchronizace je realizována pomocí HTTP protokolu. Typické použití může být pro podporu obchodních zástupců v terénu.

Instalace a konfigurace tohoto systému je celkem náročná a složitá záležitost. Správa replikací se provádí pomocí Enterprise Manageru, který si uživatel musí po nainstalování SQL Serveru doinstalovat.



Obrázek 3-2 ADO.NET Data Provider

3.3.3 Vzdálený přístup k databázi

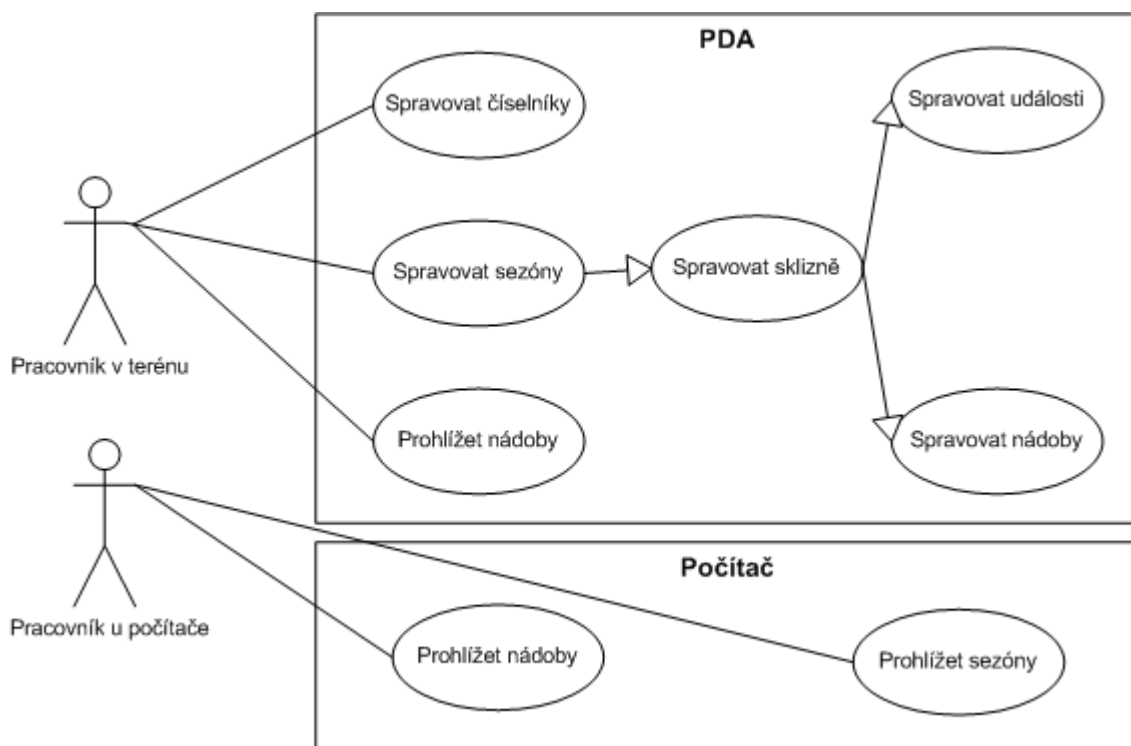
Použití vzdáleného přístupu k databázi je podmíněno použitím trvalého připojení mezi serverem a mobilním klientem. Trvalé spojení může být Ethernet nebo bezdrátová síť s protokolem 802.11.

Aplikace využívá toho, že má Compact Framework integrovanou komponentu *SqlClient*, která umožňuje přístup k databázovému serveru Microsoft SQL Server.

V případě této aplikace je využití „merge“ replikace zbytečné. Databáze na počítači bude využívána pouze pro čtení dat, je tedy nutné zajistit pouze zkopírování dat z mobilní databáze. K tomu postačí vytvořit si vlastní funkce, které budou využívat TCP/IP spojení mezi těmito dvěma zařízeními. Po vzdáleném připojení k databázovému serveru zajistí klient smazání tabulek z příslušné databáze a nahrazení tabulkami novými, přičemž jejich struktura zůstane zachována, změní se pouze data.

4 Návrh aplikace

Informační systém se skládá ze dvou částí. Jedna je umístěna na mobilním zařízení (PDA), které slouží pro zaznamenávání a prohlížení údajů přímo v terénu. Druhá část systému se nachází na stolním počítači a po replikaci dat z databáze umístěné v PDA slouží pro prohlížení a tisk složitějších tiskových sestav. Model případů užití naleznete na obrázku 4-1.



Obrázek 4-1: Use-case diagram

4.1 Aplikace v mobilním zařízení

Při návrhu aplikace jsem zohlednil malé rozměry a výpočetní výkon mobilního zařízení. Systém se skládá z několika hlavních částí, které jsem zvolil podle požadavků potenciálních uživatelů. Jsou to:

- Sezóny
- Sklizně
- Rozšířitelné číselníky
- Katalog nádob

Aplikace využívá grafického rozhraní realizované pomocí *WinForm*. Po spuštění má uživatel na výběr, se kterou částí systému chce pracovat. Aplikace je maximalizována na celou plochu

obrazovky, pouze v dolní části zůstala nabídka pro zobrazení klávesnice. Vytváření nových oken, které zpřístupňují další funkce, jsem řešil jako modální okna. Použití vláken jsem vyhodnotil jako zbytečné.

4.1.1 Výběr programovacího jazyka

Z důvodu použití .NET Framework a Visual Studia 2008 jsem se rozhodl pro programovací jazyk C#. Jde o nově vyvinutý jazyk přímo pro tuto platformu - je navržen pro její maximální využití. Jedná se o silně objektově orientovaný jazyk vycházející z programovacích jazyků Java a C++. Některé důležité vlastnosti jazyka: [8]

- o Třídy – základní stavební prvek při tvorbě objektově orientovaných aplikací obsahující akce (metody) a atributy
- o Struktury – lze je chápat jako zjednodušené třídy, jejich užitím jsou nejčastěji popisovány vlastní datové struktury
- o Výčtové typy
- o Vlastnosti – někdy označované jako chytré proměnné
- o Pole a jejich „chytrá“ verze nazývaná indexery
- o Zástupci – typově bezpečné ukazatele na funkce
- o Události – druh zástupců sloužících ke zpracování asynchronních operací

4.1.2 Jmenné prostory

Na začátku každého souboru, ve kterém jsou umístěny zdrojové kódy v jazyku C# je nutné uvést, které jmenné prostory budeme používat. Příkazem *using* importujeme příslušnou knihovnu. Jmenné prostory slouží pro oddělení k sobě logicky patřících tříd. Pro tento projekt bude vytvořen jmenný prostor *vinar*. Řádek s klíčovým slovem *namespace* určuje, do kterého jmenného prostoru třída patří.

```
using System;
using System.Data.SqlCeClient; // import knihovny pro práci s SQL CE databází
using System.Windows.Forms;   // import knihovny pro práci s formuláři

// používáme jmenný prostor vinar
namespace vinar
{
    // třída vinar dědí po třídě Form
    public partial class vinar : Form
    {
        // konstruktor třídy vinar
        public vinar()
        {
            ...
        }
    }
}
```

4.1.3 Modální a nemoďální okno

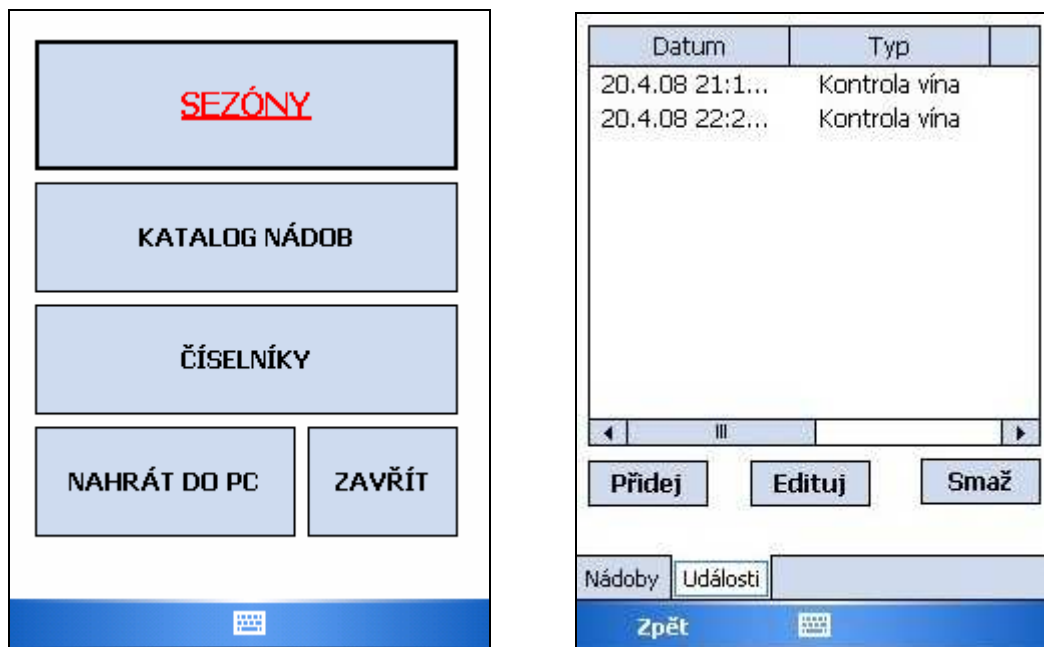
Modální okno je takové okno, které při otevření zůstává neustále navrchu, překryje všechna ostatní okna aplikace. Dokud není uzavřeno, znemožňuje práci s jinými okny aplikace. Modální okna se často používají pro dialogy. Nemoďální okna tuto vlastnost nemají, lze s nimi pracovat bez omezení. [8]

4.1.4 Vlákna

Použitím vláken je umožněno vytvořit aplikaci, která bude některé operace provádět paralelně, respektive pseudo-paralelně – čas procesoru je jednotlivým vláknům předáván střídavě s takovou rychlostí, že v lidském měřítku se operace jeví jako paralelní. Takový přístup je vhodný například u aplikací, které čekají na uživatelské vstupy a zároveň potřebují v pozadí vykonávat další operace. [8]

4.1.5 Sezóny a sklizně

Sezóny tvoří základní dělení, kterého je možné v aplikaci využít. Lze je vytvářet zcela libovolně – nejsou vázány na konkrétní období v roce. Každá sezóna pak může obsahovat libovolné množství sklizní, typicky vázané například k jedné odrůdě.



Obrázek 4-2: Ukázky uživatelského rozhraní aplikace

Každá sklizeň obsahuje základní informace o víně, které reprezentuje, množství a typy dostupných nádob včetně jejich zaplnění a hierarchický strom událostí, který slouží pro uchovávání informací o dané sklizni v průběhu času. Ukázku uživatelského prostředí naleznete na obrázku 4-2.

U sklizně si tedy můžeme nastavit:

- Sezónu, ke které konkrétní sklizeň patří
- Odrůdu vína
- Barvu vína
- Dostupné nádoby – např. Sud, objem 100l, zaplněno 90l, ...
- Události – kontrola vína, přelévání, filtrace, ...

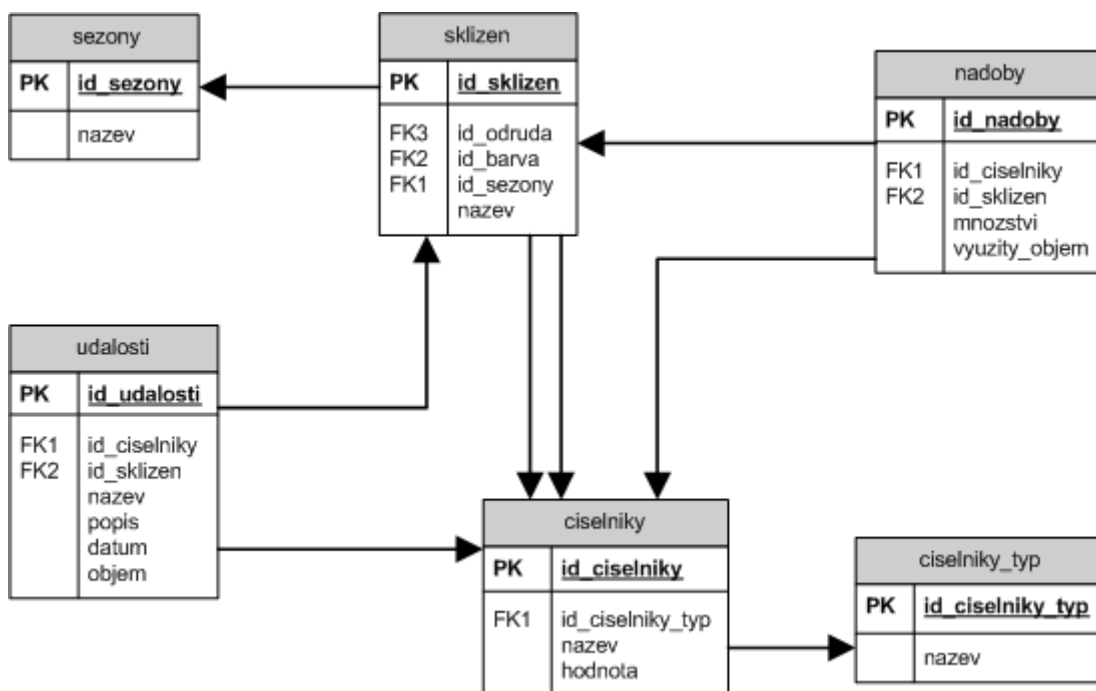
4.1.6 Číselníky

Rozšiřitelné číselníky umožňují vkládat, editovat a mazat informace, které potřebujeme pro popis vlastností, postupů zpracování nebo událostí, které s vínem provádíme. Data vložená do číselníků jsou pak použita například při editaci jednotlivých sezón a sklizní. Některé číselníky jsou pevně dané, některé jdou rozšiřovat pomocí jednoduchého rozhraní.

4.1.7 Databáze

Aplikace využívá na straně mobilního zařízení SQL Server Compact Edition. Výhodou této aplikační architektury je použití lokální databáze, která je vždy přístupná, a možnost komplexního dotazování prostřednictvím jazyka SQL. Ve fyzickém slova smyslu je SQL Server CE realizovaný jako knihovna DLL. Po technické stránce nejde o službu běžící na pozadí, jako je tomu v případě tzv. „velkého SQL Serveru“ – SQL Server CE je vlastně něco jako poskytovatel dat s podporou transakcí, vícenásobných indexů a referenční integrity. Na rozdíl od SQL Serveru nejsou podporovány pohledy, uložené procedury, triggerly ani procedurální jazyk T-SQL. SQL Server CE je navržen tak, aby minimalizoval spotřebu systémových zdrojů, hlavně paměťové karty mobilního zařízení. [2]

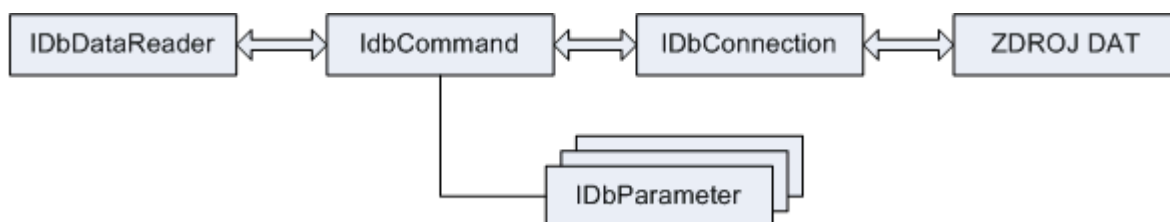
Hlavními problémy při návrhu databáze bylo vymyslet propojení sklizní, sezón a událostí s číselníky. Původně jsem uvažoval, že pro každý číselník vytvořím zvláštní tabulku, ale nakonec jsem z důvodu další rozšiřitelnosti a zbytečně vysokého počtu tabulek od tohoto řešení upustil. Všechny číselníky jsou tedy uloženy v jedné tabulce. Vytvořil jsem ještě druhou tabulku (*ciselniky_typ*), ve které jsou uloženy názvy číselníků. Odkazy na tyto záznamy jsou pak uloženy jako cizí klíče v tabulce *ciselniky*. ER diagram na obrázku 4-3 znázorňuje schéma relační databáze.



Obrázek 4-3: ER diagram

4.1.8 Připojení k databázi a čtení dat

K tomu, abychom mohli z aplikace přistupovat k datovému zdroji potřebujeme aby bylo vytvořeno aktivní spojení, které je v ADO.NET představováno implementacemi rozhraní *IDbConnection*, v našem případě *SqlConnection*. Po vytvoření tohoto spojení k datovému zdroji jsme již schopni nad tímto spojením provádět operace, které jsou představovány implementacemi rozhraní *IDbCommand*, v našem případě *SqlCommand* a v případě potřeby je možné tomuto příkazu předat parametry v podobě implementací rozhraní *IDataParameter*, konkrétně *SqlParameter* (viz obrázek 4-4).



Obrázek 4-4: Komunikace mezi jednotlivými rozhraními

Po vykonání příkazu jsou data čtena pomocí implementace rozhraní *IDataReader*, konkrétně *SqlCeReader*, který čte datový proud představující výsledkovou sadu. S tímto rozhraním souvisí také rozhraní *IDataRecord*, jehož konkrétní implementace představuje jednotlivý datový záznam a pomocí něj jsme schopni přistupovat k jednotlivým hodnotám ve sloupcích.

Následuje příklad, který demonstruje připojení k databázi, vykonání SQL dotazu a čtení dat pomocí *DataReaderu*.

```
// vytvoření spojení k databázi
SqlConnection connection = new SqlConnection("Data source=database.sdf");

// vytvoření SQL dotazu
SqlCommand command = connection.CreateCommand();
command.CommandText = "SELECT nazev FROM ciselniky";

// otevření spojení a spuštění dotazu
connection.Open();
SqlCeDataReader reader = command.ExecuteReader();

// čteme dokud je co číst
while (reader.Read())
{
    Console.WriteLine("název: " + reader["nazev"]);
}

// uzavřeme DataReader a spojení s databází
reader.Close();
connection.Close();
```

4.1.9 Komponenty

Výsledný program je *Winform* aplikace, která tedy používá vizuální komponenty viditelné v okně aplikace a prostřednictvím nich potom komunikuje s uživatelem.

Základní komponentou, kterou používám pro zobrazování údajů načtených z databáze, je *ListView*. Umožňuje kromě zobrazení obsahu v podobě ikon také zobrazení seznamu a hodí se i na zobrazení tabulkových dat. Jednotlivé buňky jsou pouze pro čtení.

Pokud je potřeba zadávat data do databáze nebo měnit stávající je vhodným řešením prvek *TextBox*, který lze pro některé situace nastavit rovněž pouze pro čtení.

Dalšími v aplikaci využívanými komponentami jsou *ListBox* pro výběr jednoho z více prvků, *Label* pro popisky, *Button* pro potvrzovací tlačítka a *TabControl* pro přepínání záložek v rámci jednoho formulářového okna.

4.1.10 Popis tříd

Při návrhu aplikace jsem se snažil oddělit třídy, které popisují chování jednotlivých formulářů aplikace od tříd, které popisují spojení s databází a operace nad ní prováděné. Výsledkem je tedy přehledné rozvrstvení, které umožňuje rychlou editaci zdrojových kódů systému. Každý formulář má

vždy třídu, ve které je definováno jeho chování a pokud je potřeba tak i třídu, která umožňuje editaci záznamů v databázi.

4.1.11 Struktura tříd

Rozvrstvení jednotlivých tříd je následující:

components

dialog – dialogová okna

novaSezona.cs – vytvoření nové sezóny

novaSklizen.cs – vytvoření nové sklizně

novyCiselnik.cs – vytvoření nového číselníku

sklizenNadoby.cs – editace nádoby

udalost.cs – editace události

window – hlavní okna aplikace

ciselniky.cs – zobrazení číselníků

katalogNadob.cs – zobrazení katalogu nádob

sezony.cs – zobrazení sezón

sklizen.cs – zobrazení konkrétní sklizně

sklizne.cs – výběr ze seznamu sklizní

synchronizace.cs – synchronizace s počítačem

comboBox.cs – funkce pro práci s comboBoxy

paramSklizen.cs – pomocné funkce

database

dbCiselniky.cs – funkce pro editaci a mazání dat z číselníků

dbKatalogNadob.cs – funkce pro editaci a mazání dat z katalogu nádob

dbMain.cs – připojení k databázi, základní funkce pro výpisy, zjištění počtu řádků tabulky, ...

dbSezony.cs – funkce pro editaci a mazání sezón

dbSklizen.cs – funkce pro editaci a mazání sklizní

dbSynchro.cs – funkce pro připojení k vzdálenému databázovému serveru

vinar.cs

config.cs – nastavení přístupových údajů k databázím

Všechny třídy z *components.dialog* a *components.window* dědí od třídy *System.Windows.Form*. Funkce pro připojení k vzdáleného databázovému serveru jsou umístěny v souboru *dbSynchro.cs*. V souboru *dbMain.cs* jsou základní funkce pro práci s lokální databází. Všechny ostatní soubory v *database* jsou potomci třídy *dbMain*.

4.1.12 Popis činnosti výsledné aplikace

Po spuštění aplikace je uživateli nabídnuto úvodní menu. Prostřednictvím něj si vybere část systému, se kterou chce pracovat – jedná se o sezóny, číselníky, katalog nádob a synchronizaci. Po kliknutí na příslušné tlačítko se otevře nové modální okno, ze kterého se mohou případně otevřít další modální okna.

Pokud si uživatel vybere tlačítko „Zobraz sezóny“, připojí se aplikace k databázi, která je umístěna v jednom souboru na mobilním zařízení a nabídne uživateli dostupné sezóny. Uživatel si může jednu vybrat. Následně je vybědnut k vybraní příslušné sklizně. V obou případech lze kromě čtení dat provádět i další operace – editovat nebo smazat stávající položky nebo vytvořit nové.

Nejdůležitější částí je bezesporu práce se sklizní. Zde se uživatel přepíná pomocí záložek mezi událostmi a dostupnými nádobami, se kterými může libovolně manipulovat. K dispozici je mu také jednoduchá funkce pro výpočet doslazování moštu.

Číselníky jsou další částí systému. Umožňují uživateli provádět operace s přednastavenými daty, která jsou používána v dalších částech aplikace. Mezi jednotlivými číselníky se lze přepínat klasickou roletkou – *comboBox*.

Pro zobrazení nádob slouží tlačítko „Katalog nádob“. Data vypsána do komponenty *ListView* informují o všech lahvích, které jsou v systému evidovány, včetně informací o jejich obsahu.

Po kliknutí na tlačítko „Synchronizace“ v úvodním menu proběhne v případě dostupnosti vzdáleného databázového serveru replikace dat. O úspěšnosti operace je uživatel informován dialogem.

4.2 Aplikace v počítači

Tato bakalářská práce se aplikací v počítači zabývá jen okrajově, hlavní část se věnuje aplikaci v mobilním zařízení. Hlavním úkolem je přenést data z mobilního zařízení do počítače a umožnit jednoduché zobrazení složitějších tabulek v internetovém prohlížeči.

4.2.1 Databáze

Hlavní částí je nainstalovaný Microsoft SQL Server 2005 Express, který umožňuje vzdálené připojení mobilního zařízení s následnou aktualizací dat. Dále uchovává data, která jsou následně poskytována webové aplikaci. Z tohoto důvodu musí na počítači běžet také webový server. Vhodným řešením by se zdálo použití Microsoft IIS (Internet Information Server), nicméně dostupné verze tohoto produktu jsou celkem omezené. Z tohoto důvodu jsem se rozhodl pro následující kombinaci:

- Microsoft SQL Server 2005 Express
- Webový server Apache, na které je možné spouštět skripty v jazyce PHP

4.2.2 PHP

PHP (rekurzivní zkratka PHP: *Hypertext Preprocessor*, „*PHP: Hypertextový preprocesor*“, původně *Personal Home Page*) je skriptovací programovací jazyk určený pro tvorbu dynamických webových stránek. Nejčastěji se začleňuje přímo do struktury HTML stránek. PHP skripty jsou prováděny na straně serveru, uživateli je zaslán pouze výsledek. PHP je nezávislé na platformě, skripty fungují bez úprav na mnoha operačních systémech. Obsahuje rozsáhlé knihovny funkcí pro zpracování textu, grafiky, práci se soubory, přístup k většině databázových serverů a podporu celé řady internetových protokolů.

Pro tento projekt je důležité, že server, na kterém lze spouštět PHP skripty, je volně dostupný a může běžet pod Windows. Instalace je velmi jednoduchá, pro námi vyžadované funkce není potřeba další konfigurace. Nyní je počítáno s tím, že server bude běžet lokálně na stejném počítači jako SQL Server. Nicméně pokud by bylo v budoucnu potřeba spouštět skripty z internetu, je možné využít podpory PHP na téměř každém webhostingu.

4.2.3 Připojení k databázovému serveru

Programovací jazyk PHP poskytuje funkce pro práci s Microsoft SQL databází, nicméně nejsou standardně povoleny. Proto je použito rozhraní ODBC (*Open Database Connectivity*), které je ve většině distribucí standardně instalováno.

ODBC je standardizované softwarové API (*Application interface*) pro přístup k databázovým systémům. Snahou ODBC je poskytovat přístup nezávislý na programovacím jazyku, operačním systému a databázovém systému.

Prvním krokem je konfigurace přístupu k databázi prostřednictvím ODBC. Jednou z možností je použití *Správce zdrojů dat ODBC*, který je součástí operačních systémů Windows, v němž lze vytvořit *DSN (Data Source Name)*, přes které se pak k databázi přistupuje. Já jsem však použil jiného postupu – vytvořil jsem *DNS-less* spojení, které je podporováno od PHP 4.3.0. Výhodou je, že odpadá nutnost cokoli konfigurovat – stačí mít nainstalovanou příslušnou verzi PHP. Následuje příklad připojení k databázi pomocí *DNS-less* spojení.

```
<?php
...
// nastavení přístupových údajů
$dsn = "Driver={SQL Server};Server=127.0.0.1,1433;Database=vinar1;";
$user = "sa";
$password = "vinar";
$sqlconnect = odbc_connect($dsn, $user, $password);
...
?>
```

5 Problémy při řešení

Některé komplikace při vývoji aplikace jsem uvedl v příslušných kapitolách. V této části se pokusím zmíněné problémy zrekapitulovat a některé podrobněji popsat.

5.1 Uvolňování paměti v PDA

Jedním z nejzávažnějších problémů, které jsem řešil na straně PDA, bylo uvolňování paměti. Ačkoliv společné prostředí pro běh aplikací (CLR) obsahuje *Garbage Collector*, který se stará o automatické uvolňování nepotřebných programových objektů z paměti, při práci s SQL CE se mi po několika dotazech na databázi objevila výjimka „*Not enough storage is available to complete this operation*“.

Řešení jsem hledal v diskuzích na zahraničních serverech. Návodů jak vyřešit tento problém jsem našel hned několik. Prošel jsem si zdrojové kódy vytvářené aplikace a zkontroloval jsem, jestli po každém dotazu na databázi důsledně zavírám spojení. Po několika úpravách aplikace zvládala řádově několik desítek dotazů na databázi.

V některých diskuzích jsem se dočetl, že dalším možným řešením je vytvořit spojení s databází hned po spuštění aplikace a využívat ho po celou dobu životnosti aplikace. Já však používám vícenásobné spojení.

Výsledným řešením, které zaručilo bezproblémovou stabilitu aplikace je volání metody *System.GC.Collect()* v průběhu běhu programu. Tento příkaz jsem umístil za každé uzavření spojení s databází. Uvedená metoda zajišťuje bezprostřední uvolnění nevyužívaných paměťových prostředků.

5.2 Konfigurace SQL Serveru

Určitým zdržením se stala konfigurace SQL Serveru 2005. V aplikaci potřebuji použít autorizaci prostřednictvím uživatele a hesla - *SQL Server Authentication*. Při výchozím nastavení je však podporován pouze *Windows Authentication Mode*, který kontroluje autorizaci k přístupu na základě přihlašovacích údajů do operačního systému lokálního počítače. Pro změnu nastavení je nutné odinstalovat aktuální verzi SQL Serveru, která byla instalována spolu s Visual Studiem a nainstalovat software znovu. Tentokrát jsem zvolil zdarma dostupnou základní verzi – SQL Server 2005 Express.

Během instalace programu proběhla nová konfigurace serveru, kde bylo možné nastavit režim *Mixed mode*, který umožňuje použití obou způsobů autorizace. Následně je potřeba provést povolení připojení prostřednictvím TCP/IP protokolu v aplikaci SQL Server Configuration Manager a nastavit firewall. Podrobný popis konfigurace je uveden v příloze 1.

Po úspěšné konfiguraci SQL Serveru a zprovoznění replikace dat jsem musel vyřešit způsob získávání dat z databáze prostřednictvím webové služby. Po několika neúspěšných pokusech o zprovoznění PHP na IIS a kvůli některým omezením zdarma dostupné varianty tohoto webového serveru jsem se rozhodl pro Apache.

6 Testování

Určitou překážkou při vývoji a testování je fakt, že Visual Studio 2008 je samo o sobě dost náročné na paměť. Nicméně tato nevýhoda je vykoupena výbornou podporou mobilních zařízení. Pro vývoj a ladění jsem měl k dispozici skutečné PDA, emulátoru jsem tedy nevyužíval.

Testování jsem prováděl postupně jakmile jsem dokončil každou část systému. V první fázi jsem navrhl uživatelské rozhraní a otestoval ho na stolním počítači. Netestoval jsem rovnou na PDA z důvodu pomalého sestavení aplikace a nahrání do mobilního zařízení.

Jakmile jsem byl s vytvořenou aplikací spokojen, upravil jsem projekt do podoby, aby mohl běžet na mobilním zařízení. Jelikož je .NET Compact Framework podmnožinou .NET Frameworku, stačilo při vývoji používat komponenty a funkce, které jsou součástí obou frameworků. Tím se zajistila kompatibilita stolní verze s mobilní verzí.

V další fázi jsem testoval spolupráci aplikace s databází umístěnou na mobilním zařízení. Vyskytlo se pár problémů s připojením k databázi, zejména již zmíněné zaplnění paměti při práci s databází, které se nakonec podařilo vyřešit.

Závěrečnou fází bylo testování spolupráce mobilní aplikace s velkou databází umístěnou na stolním počítači. Po překonání problémů, spočívajících ve špatné konfiguraci serveru a špatně nastaveném připojovacím řetězci v mobilní aplikaci, jsem zprovoznil jednosměrnou replikaci dat. Jediným problémem bylo, pokud jsem chtěl překopírovat tabulku, ve které se vyskytovaly řádky s hodnotou *NULL*, které byly do databáze zaneseny při testování. V tomto případě docházelo ke generování výjimky, která pak kopírování přerušila. Po opravě databáze a vložení dalších řádků za účelem testování rychlosti aplikace už vše fungovalo.

6.1 Použitelnost v reálných podmínkách

Program je nyní ve fázi, kdy umožňuje pohodlné ovládání systému prostřednictvím aplikace v mobilním zařízení a prohlížení nasbíraných údajů na stolním počítači. Systém by byl po nezbytných úpravách schopný nasazení v reálných podmínkách.

Pokud by se uvažovalo o reálném provozu, bylo by dobré přemýšlet o možnostech rozšíření v závislosti na konkrétním použití. Bylo by také potřeba vytvořit instalátor, který by systém nakonfiguroval a nahrál do počítače a do PDA. Některé návrhy na rozšíření jsem uvedl v další kapitole.

7 Návrh na rozšíření

Funkčnost vytvořené aplikace se dá rozvíjet mnoha směry. Může se jednat o rozšíření funkčnosti mobilního zařízení, ale také o propracování části aplikace běžící na stolním počítači a s tím související synchronizace. Zajímavou myšlenkou je také napojení na internetový obchod.

7.1 Mobilní zařízení

Aplikace v mobilním zařízení je dle mého názoru na celkem dobré úrovni - ať už se jedná o funkčnost nebo o uživatelské rozhraní. V případě rozšíření funkčnosti bych se zaměřil na implementaci dalších kalkulátorů, které by vinařovi usnadnily práci. V případě potřeby by bylo možné přidat také další vlastnosti a operace s vínem.

7.2 Stolní aplikace

V aktuální fázi je aplikace, běžící na počítači, se kterým je mobilní zařízení synchronizováno, používána pouze pro zobrazování synchronizací přenesených údajů. V budoucnu by však šlo aplikaci rozšířit o možnost editace a přidávání těchto záznamů. Tímto způsobem by se omezila nutnost vkládat do PDA informace, které jsou známy předtím, než vinař vyrazí do terénu. Menší potřeba využívat nedokonalé ovládání mobilního zařízení by značně zvýšila komfort při práci s aplikací. PDA by tedy mohlo být využíváno pouze k účelu, ke kterým je v tomto případě určeno – přidávání informací v terénu a v případě nutnosti jejich čtení.

V případě rozšíření aplikace o tyto funkce by bylo potřeba přepracovat koncept synchronizace obou zařízení. Nyní probíhá pouze jednosměrná synchronizace – z PDA do počítače. Nutností by se stala synchronizace, která využívá obousměrné replikace dat. To by umožnilo také současný provoz více mobilních zařízení. Při každé synchronizaci by byla do zařízení zkopírována i data, která byla nasbírána ostatními zařízeními.

7.3 Internetový obchod

Po rozšíření aplikace popsané v předchozích kapitolách by byla aplikace schopna spolupracovat s internetovým obchodem prodávajícím víno. Obsahuje totiž množství informací, které by bylo možné využít – typ vína, odrůda, množství a typ lahví apod.

8 Závěr

Náplní této bakalářské práce bylo seznámení se základními požadavky amatérského vinaře nebo malovýrobce vína na informační systém, navrhnutí a použití vhodných mobilních technologií s důrazem na práci v terénu a následnou synchronizaci s počítačem. Výsledkem měla být funkční aplikace.

Při vývoji systému jsem prošel několika fázemi. Nejprve jsem si ujasnil požadavky, které by měla aplikace splňovat z pohledu vinaře, poté jsem vybral technologie a programovací jazyk a následně navrhl uživatelské rozhraní a databázi. V další fázi jsem aplikaci implementoval a potom testoval jednotlivé části.

Výsledkem práce je systém, který se skládá ze dvou programů, kde hlavní běží na mobilním aplikaci a umožňuje vinařovi ukládat a číst data v terénu, v tomto případě ve vinném sklípku. Druhý program je jednoduchý skript, který běží na webovém serveru a umožňuje prohlížení zaznamenaných informací. Důležitou částí je databázový server, který propojuje obě aplikace – první na něj data posílá, druhá je čte.

Osobním přínosem byla možnost práce v moderním vývojovém prostředí a objektově orientovaném jazyku. Vyzkoušel jsem si strasti programování aplikací pro mobilní zařízení a propojení se vzdálenou databází. Využil jsem v praxi znalosti získané při předchozím studiu, ale také jsem získal studiem dalších materiálů mnoho nových.

Všechny body zadání, které byly stanoveny, se mi podařilo splnit, stejně tak jako se mi podařilo naplnit své představy, které jsem si před začátkem práce stanovil. Aplikace je funkční, nicméně si myslím, že v případě reálného využití je potřeba zdokonalit část systému běžícím na webovém serveru a také replikaci dat mezi mobilní a velkou databází.

Literatura

- [1] Pavloušek, P. Výroba vína u malovinařů. Grada Publishing, Praha, 2006.
- [2] Lacko, L. Programujeme mobilní aplikace ve Visual Studiu .NET. Computer Press, Brno, 2004.
- [3] Kent, J. Visual C# 2005 bez předchozích znalostí. Computer Press, Brno, 2007.
- [4] Price, J. C# – programování databází. Grada Publishing, Praha, 2005.
- [5] Personal digital assistant. Wikipedie, otevřená encyklopedie, 1.4.2008. Dokument dostupný na URL http://cs.wikipedia.org/wiki/Personal_Digital_Assistant (7.5.2008)
- [6] Windows Mobile. Wikipedie, otevřená encyklopedie, 9.4.2008. Dokument dostupný na URL http://cs.wikipedia.org/wiki/Windows_Mobile (11.5.2008)
- [7] Fajfr, S. Kurz C# - 1. lekce. Programujte.com, 13.6.2006. Dokument dostupný na URL <http://programujte.com/index.php?akce=clanek&cl=2006061103-kurz-c#-1-lekce> (7.5.2008)
- [8] Puš, P. Poznáváme C# a Microsoft .NET. Živě.cz, 23.11.2004.
Dokument dostupný na URL http://www.zive.cz/Programovani/C_CSHARP/sc-74/default.aspx (7.5.2008)
- [9] J2ME vs. .Net Compact Framework. 24.6.2005. Dokument dostupný na URL http://www.must.edu.my/~dwong/resources/mobile_commerce_web/j2mevsnetcf.html (8.5.2008)
- [10] Šeda, J. J2EE, .NET a vývoj rozsáhlých systémů 2. Interval.cz, 11.2.2003. Dokument dostupný na URL <http://interval.cz/clanky/j2ee-net-a-vyvoj-rozsahlych-systemu-2/> (8.5.2008)
- [11] Vinařský průvodce. 2008. Dokument dostupný na URL www.vinarskypruvodce.eu (11.5.2008)
- [12] Webová prezentace vinaře. 2004. Dokument dostupný na URL www.padalikovo.cz (11.5.2008)
- [13] Hřebačka, L. Vinařská stránka – technologie výroby bílého vína. 2008. Dokument dostupný na URL <http://vinar.unas.cz/technolb.html> (11.5.2008)

Seznam příloh

Příloha 1. Instalace a spuštění aplikace

Příloha 2. Manuál k aplikaci na PDA

Příloha 3. CD obsahující zdrojové soubory a technickou zprávu v PDF

Příloha 1 – Instalace a spuštění aplikace

Před samotným spuštěním aplikace je třeba nainstalovat Visual Studio 2008, které umožní spuštění projektu. Zdarma lze stáhnout ze stránek firmy Microsoft variantu Express. Zároveň stáhněte a nainstalujte i program ActiveSync 4.5, který umožňuje spojení s PDA.

Pro provoz databázového serveru je nutný SQL Server 2005. Při instalaci dávejte pozor na správné nakonfigurování serveru. Je nutné povolit *Mixed mode*; autorizace je potom umožněna prostřednictvím integrace bezpečnostního modelu Windows a také prostřednictvím SQL autorizace, kterou budete potřebovat. Ačkoliv lze tuto variantu zabezpečení údajně nakonfigurovat i po instalaci serveru, mně se to nepodařilo, proto doporučuji konfiguraci přímo prostřednictvím instalačního průvodce. SQL Server 2005 Express je rovněž možné stáhnout zdarma ze stránek Microsoftu.

Dalším krokem je povolení vzdáleného přístupu k databázi, který provedete v programu *SQL Server Configuration Manager*, který se nachází pravděpodobně v nabídce *Start – Programy – Microsoft SQL Server 2005 – Configuration Tools*. Klikněte na *Protocols for SqlExpress – Tcp/Ip – IP Adresses – IPAll*. Řádek *TCP Dynamic Port* nastavte na 0 a *TCP Port* na 1433. Poté server restartujte.

Pokud máte povolený firewall, je třeba nastavit výjimky pro soubory (umístění se může lišit v závislosti na tom, kam byl SQL Server 2005 nainstalován):

- o *C:\Program Files\Microsoft SQL Server\90\Shared\sqlbrowser.exe*
- o *C:\Program Files\Microsoft SQL Server\MSSQL.1\MSSQL\Binn\sqlservr.exe*

Aplikace běží nad .NET Compact Frameworkem, ale vzhledem k tomu, že už máte nainstalováno nejnovější Visual Studio, není potřeba ho instalovat. To samé platí i o SQL Compact Edition. Pokud by se i přesto objevily komplikace se spuštěním aplikace, je možné do počítače (do PDA příslušné CE verze) dodatečně nainstalovat nejnovější verze produktů .NET Compact Framework 3.5 a SQL Server 2005 Compact Edition.

Webové rozhraní aplikace zprovozníte nainstalováním volně dostupného serveru Apache a nakopírováním PHP skriptů z příloženého CD do příslušného adresáře. Skripty se nachází v adresáři *server*. V souboru *config.php* je nutné zkontrolovat přístupové údaje k databázi.

Poslední krokem je otevření projektu ve Visual Studiu. Na CD se nachází adresář *klient*, který si překopírujete na lokální disk. Zároveň nakopírujete do PDA databázi SQL CE, kterou naleznete v adresáři *klient_db*. Poté projekt otevřete a v Solution Exploreru najdete soubor *config.cs*, ve kterém dle pokynů změňte příslušné údaje. Nyní již stačí připojit k počítači PDA a například klávesou F5 program spustit.

Příloha 2 – Manuál k aplikaci na PDA

Po spuštění aplikace se zobrazí úvodní nabídka (viz obrázek 1), ze které můžete otevřít katalog nádob, který vám poskytne souhrnné informace o všech lahvích (tlačítko „Katalog nádob“), můžete editovat číselníky (tlačítko „Číselníky“) a po kliknutí na tlačítko „Nahrát do pc“ také synchronizovat s počítačem (nutné připojení kabelem přes USB). Po kliknutí na tlačítko „Zavřít“ aplikaci ukončíte.

Do hlavní části systému se dostanete kliknutím na tlačítko „Sezóny“. Následně jste vyzváni k vybrání sezóny, se kterou budete chtít pracovat (viz obrázek 2). Pokud chcete vytvořit novou sezónu, kliknete na příslušné tlačítko v horní části obrazovky. Pokud si vyberete některou ze stávajících a jednou na ni kliknete, můžete ji po stisknutí příslušného tlačítka editovat, smazat nebo otevřít. Pro otevření můžete použít také „dvoj-klik“.



Obrázek 1: Hlavní nabídka



Obrázek 2: Výběr sezóny

V dalším kroku následuje výběr sklizně (viz obrázek 3), která se váže k otevřené sezóně. Veškeré operace se provádí analogicky předchozímu kroku.

Nová Edituj Smaž

jaro - rulanda 7
podzim - Chardonnay
podzim - Savignon

Otevři vybranou sklizeň

Zpět

Obrázek 3: Výběr sklizně

Sezóna 2008

Název podzim - Chardonnay

Odrůda Chardonnay

Barva Bílé

OK Storno

Obrázek 4: Editace sklizně

Po vybrání a otevření sklizně máte možnost přidávat nádoby (viz obrázek 5) a vytvářet události (viz obrázek 7), které slouží k uchovávání vlastností vína a operací s ním prováděnými. K dispozici je také kalkulátor, pomocí kterého lze vypočítat množství cukru potřebného k doslazení moštu (viz obrázek 8). V celé aplikaci lze používat tlačítko „Zpět“, které vás vrátí do předcházející nabídky. K vyplňování políček je k dispozici virtuální klávesnice, kterou můžete zapnout v dolním menu.

Typ	Množ...	Zapl.	
Sud 100 l	0	0	
Sud 100 l	4	0	
Lahev 0.75 l	2	0	

Přidej Edituj Smaž

Nádoby Události

Zpět Kalkulátor

Obrázek 5: Nádoby

Datum	Typ	Ná
.05 23:35:32	Událost	sk
.05 23:36:15	Kontrola vína	ba

Přidej Edituj Smaž

Nádoby Události

Zpět Kalkulátor

Obrázek 6: Události

Typ události

Datum

Název

Objem (l)

123 + ě š č ř ž ý á í é = [←
 Tab q w e r t z u i o p ú)
 Cap a s d f g h j k l ů ě
 Shift y x c v b n m , . - ←
 Ctl áü ; ? ↓ ↑ ← →

Zpět

Obrázek 7: Editace události, klávesnice

Původní cukernatost (°NM)

Požadovaná cukernatost (°NM)

Objem moštu (hl)

Koeficient (0.8 - 1)

Vypočítej 22 kg

Zpět

Obrázek 8: Kalkulátor - doslazování