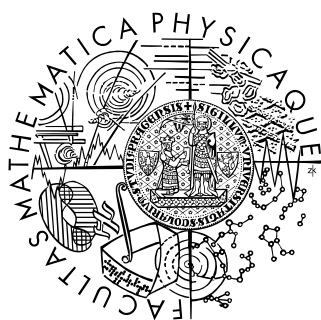


Univerzita Karlova v Praze
Matematicko-fyzikální fakulta

BAKALÁŘSKÁ PRÁCE



Ivan Vacula

Framework pro vývoj webových aplikací

Katedra aplikované matematiky

Vedoucí bakalářské práce: RNDr. Tomáš Tichý

Studijní program: Informatika, Programování

2008

Ďakujem vedúcemu RNDr. Tomášovi Tichému za korektný a chápavý prístup ku mne aj tejto práci.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce.

V Praze dne 15. 5. 2008

Ivan Vacula

Obsah

1. Konvencie a predpoklady	4
2. Motivácia	4
3. Čo je to Emanager?	5
4. Úlohy projektu Emanager	5
4.1. Výhody pre vývojárov	5
4.1.1. Prečo deklaratívna syntax?	5
4.2. Výhody pre používateľov	6
5. Popis súčastí systému	6
5.1. Emanager Controls	7
5.1.1. AddressBox	7
5.1.2. AdvancedGridView	8
5.1.3. AdvancedCheckBoxField	10
5.1.4. AdvancedSqlDataSource	10
5.1.5. Calendar	10
5.1.6. Menu	11
5.1.7. NullableCheckBox	12
5.1.8. SmartLookup	12
5.1.9. TabsView a Tab	13
5.1.10. TubularView	14
5.2. Emanager Security	14
5.2.1. Autorizačný model systému Emanager	15
5.2.2. Dvojitá autentifikácia	15
5.2.3. Hierarchia tried	16
5.3. Emanager Error Handling	17
5.4. Emanager Auto Updater	18
6. Záver	18
7. Použitá literatúra	19

Název práce: Framework pro vývoj webových aplikací

Autor: Ivan Vacula

Katedra (ústav): Katedra aplikované matematiky

Vedoucí bakalářské práce: RNDr. Tomáš Tichý

E-mail vedoucího: tichy@math.cas.cz

Abstrakt: Framework pro rychlý vývoj intranetových webových aplikací zaměřený na příjemné a funkční designovací prostředí s jednoduchou správou, které navíc umožní další rozšiřitelnost nasazených aplikací.

Klíčová slova: ASP.NET Emanager web framework

Title: Framework for web-based application development

Author: Ivan Vacula

Department: Department of Applied Mathematics

Supervisor: RNDr. Tomáš Tichý

Supervisor's e-mail address: tichy@math.cas.cz

Abstract: Framework for rapid intranet database web-based application development focused on pleasant and functional design-time environment and easy administration and scalability of deployed applications.

Keywords: ASP.NET Emanager web framework

1. Konvencie a predpoklady

V tejto bakalárskej práci budem obvykle používať anglické odborné pojmy v ich pôvodnom znení. Mám pocit, že akékoľvek preklady by čitateľa iba zbytočne miatli a mohli spôsobovať nejednoznačnosti v chápaní mojich myšlienok. Ak je však slovenský ekvivalent dostatočne zaužívaný, budem ho s anglickým originálom niekedy obmieňať.

Predpokladám, že čitateľ sa vyzná v základoch problematiky a preto sa k vysvetleniam ohľadom použitých technológií, prostredí apod. budem odkláňať iba ak budem mať pocit, že vyžadujú hlbšie ozrejmienie nejakej ich časti. Často však pôjde skôr o priblíženie môjho pohľadu na danú vec, ako o nejaké odborné definície, preto si myslím, že aj pre expertov nebudú takéto pasáže na škodu.

2. Motivácia

V roku 2003 som začal pracovať vo svojom prvom programátorskom zamestnaní. Jedným z prvých projektov, na ktorom som spolupracoval bola databázová intranetová aplikácia na báze webu.

Odvtedy uplynulo zopár rokov a podieľal som sa na niekoľkých databázových systémoch na rôznych platformách. I keď každá z platforiem, s ktorými som sa zoznámil má určite svoje prednosti, pre svoju komplexnosť ako aj celkovú filozofiu mi najviac učaroval ASP.NET. Mnohé veci, ktoré som pred tým, ako som ho začal používať musel navrhovať a programovať sám, rieši priamo ASP.NET a pre mnohé iné ma vybudovanú perfektnú podporu.

Ani ASP.NET však zďaleka nie je dokonalá technológia a nerieši určite všetko, čo bežný klient od intranetovej aplikácie očakáva. Takisto niektoré predstavy vývojárov musia byť dodatočne implementované.

Už dlhšie som teda uvažoval nad vytvorením akéhosi rozšírenia ASP.NET pre praktické potreby bežnej aplikácie, ktoré vyústilo až do systému Emanager a tejto bakalárskej práce.

3. Čo je to Emanager?

System Emanager je súbor troch .NET knižníc a jednej Windows služby. Ide o framework vylepšujúci a rozširujúci technológiu ASP.NET 2.0 v niekoľkých oblastiach, ako sú web controls, autorizácia, autentifikácia a ošetrovanie chýb v aplikácii využívajúcej framework Emanager.

Celý systém je navrhnutý tak, aby zapadal do ideového modelu ASP.NET a aby popri funkčnosti aj čo najviac uľahčoval prácu vývojárovi.

Zdrojový kód systému Emanager je napísaný kompletne v jazyku C#.

4. Úlohy projektu Emanager

System by mal priniesť výhody ako pre vývojárov, ktorí na ňom stavajú prípadnú aplikáciu, tak aj pre používateľov konkrétnej finálnej aplikácie.

4.1. Výhody pre vývojárov

Ide predovšetkým o zrýchlenie a uľahčenie vývoja. System som sa snažil navrhovať tak, aby sa väčšina bežných funkčností webových aplikácií dala zapísať pomocou deklaratívnej syntaxe a aby bol programátor nútený pristúpiť k písaniu klasického kódu až pri programovaní naozaj pokročilých vecí.

K deklaratívnej syntaxi si dovoľím malú odbočku.

4.1.1. Prečo deklaratívna syntax?

Prikláňanie sa k deklaratívnej syntaxi namiesto písania samotného kódu je v podstate akýmsi trendom už v samotnom ASP.NET. Stačí sa napríklad pozrieť na to, aké zmeny v tomto smere priniesol ASP.NET 2.0 oproti verziám 1.0, či 1.1.

Niet sa však čo diviť. Výhody deklaratívnej syntaxe sú zjavné. Určite je takto zapísaný "kód" čitateľnejší, je z neho veľmi dobre zrejmá štruktúra ASPX stránky a umožňuje robiť kozmetické úpravy na stránke aj menej skúseným vývojárom (dokonca si myslím, že aj takým, ktorí s ASP.NET ešte vôbec neprišli do styku).

Mojou obľúbenou výhodou je však možnosť editovať takýto dokument v Design móde Visual Studia. Jednoducho si z ToolBox-u vytiahnem control, ktorý potrebujem, naklikám jeho jednotlivé vlastnosti, prípadne vyberiem z TaskList-u niektorú z predvolených akcií. Všetko sa mi mení takpovediac pred očami a vidím výsledný efekt. A to všetko za zlomok času a oveľa pohodlnejšie ako pri priamej editácii či už deklaratívneho alebo klasického kódu.

Práve podpora Design prostredia a deklaratívnej syntaxe vôbec bola pre mňa pri navrhovaní systému Emanager jednou z priorit.

Ďalšou podstatnou vecou v systéme Emanager je využívanie pokročilých funkcií Microsoft SQL Serveru. Ide predovšetkým o použitie Common Table Expressions (CTE). Tie urýchľujú aplikáciu tým, že presúvajú mnohé z vecí, z webovej časti aplikácie na databázovú časť, kde správne patria. Za všetky uvediem jeden skvelý príklad - stránkovanie v zoznamoch (GridView a SqlDataSource).

4.2. Výhody pre používateľov

Samozrejماً výhoda, ktorá vyplýva z takéhoto frameworku je unifikované prostredie aplikácie, v ktorej sa tak používateľ ľahšie orientuje.

Mnohé controls sú navrhnuté tak, aby používateľovi umožnili ľahkú orientáciu a poskytovali drobné nápovede (systém hintov).

V neposlednej rade sa stránky rýchlejšie generujú (napríklad vďaka použitiu CTE) a sú menšie (vďaka špeciálnemu menu). Obidva tieto faktory prispievajú k rýchlejšiemu načítavaniu stránok a tak príjemnejšej práci používateľa.

5. Popis súčastí systému

Systém Emanager pozostáva z niekoľkých oddeliteľných súčastí.

V prvom rade ide o class library *Emanager Controls*. Ide o niektoré rozšírené a niektoré úplne nové ASP.NET Web Controls.

Ďalšou vecou sú autentifikácia a autorizácia - tu si myslím, že je ASP.NET (bez dodatočného programovania) pomerne málo flexibilný, preto som navrhol a naprogramoval systém, ktorý okrem iného podstatne uľahčuje integráciu s existujúcimi webovými systémami, ale zároveň sa dobre integruje do pôvodného ASP.NET modelu. Všetky triedy týkajúce sa autentifikácie a autorizácie, ako aj ďalšie komponenty súvisiace s bezpečnosťou sa nachádzajú v class library *Emanager Security*.

Pre nasadenú aplikáciu je dôležitý monitoring prípadných chýb. Ten zastrešuje knižnica *Emanager Error Handling*.

Aby bolo možné všetky vymenované súčasti používať jednoducho spolu, vytvoril som aj *Visual Studio Project Template*, ktorý je možné použiť ako základ na vývoj nových aplikácií nad systémom Emanager.

Trošku oddeleným podproduktom celého systému je Windows Service, ktorá umožňuje automatické aktualizácie nasadenej aplikácie. Je predmetom projektu *Emanager Auto Updater*.

Jednotlivé časti som sa snažil navrhovať tak, aby sa dali používať samostatne a nezávisle, niekedy sa však závislostiam prirodzene vyhnúť nedalo.

5.1. Emanager Controls

5.1.1. AddressBox

Control na zadávanie a zobrazovanie adresy. Keďže adresa je pomerne častá položka v mnohých formulároch, tak control na jej editáciu určite nájde svoje uplatnenie.

Adresu je súčasne možné zbalit' do kompaktnejšej formy, ak nie je práve editovaná.

Interne AddressBox pozostáva z dvoch Panel controlov, ktoré obsahujú jeho kompaktný a rozbalený stav.

Prepínanie medzi stavmi je realizované pomocou JavaScript-u na klientskom prehliadači.

[Zatvoriť](#)

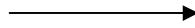
Ulica:
Napr.: Sofijská

Číslo:
Napr.: 6

Mesto:
Napr.: Košice

PSČ:
Napr.: 04013

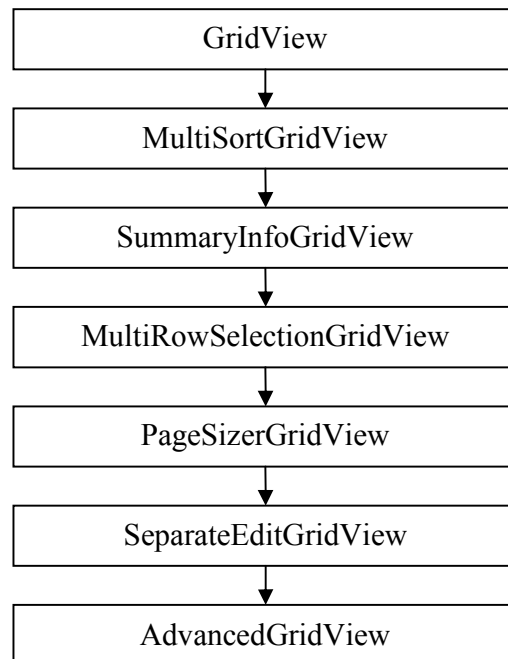
Krajina:
Napr.: 04013



[Ke Karlovu 3, Praha](#)

5.1.2. AdvancedGridView

Jeden z najdôležitejších controlov knižnice Emanager Controls. Samotná trieda AdvancedGridView vlastne iba sprístupňuje celý strom tried, ktorý postupne obohacuje pôvodnú ASP.NET triedu GridView o ďalšie a ďalšie funkčnosti.



Popis jednotlivých tried:

- **MultiSortGridView** – pridáva možnosť zotriediť zoznam podľa viacerých stĺpcov súčasne. Na reprezentáciu svojho stavu používa triedu

MultiSortExpression, ktorá je v konkrétnej inštancii AdvancedGridView prístupná pomocou metódy GetMultiSortExpression. Aby koncept fungoval s pôvodnou funkčnosťou ASP.NET, celý triediaci reťazec je fyzicky uložený vo vlastnosti SortExpression a do MultiSortExpression sa podľa potreby rozoberie alebo z neho naspäť poskladá.

- **SummaryInfoGridView** – pridáva možnosť zobrazovať sumár za stranu a za zoznam v päte GridView. Dáta pre sumárny riadok sú získavané na strane SQL serveru. Ten k výberu podľa potreby pridá jeden riadok pre sumár za aktuálnu stranu daného výberu a jeden pre sumár za celý výber. Tieto dva riadky oddeľuje od dát databázového serveru SummaryDataView, ktorý je rozšírením štandardného ASP.NET DataView.
- **MultiRowSelectionGridView** – pridáva možnosť označiť aj viacero riadkov naraz, napríklad na hromadné spracovanie záznamov. Do každého riadku prehľadu pridá na pozíciu určenú vlastnosťou SelectionColumnIndex jeden checkbox, ktorý má používateľ možnosť zaškrtnúť. Zoznam zaškrtnutých riadkov sa uchováva vo vlastnosti SelectedDataKeys vo ViewState, takže sa zachováva aj medzi jednotlivýmiPostBack-mi, prechodmi na ďalšie stránky zoznamu apod. Aby MultiRowSelectionGridView mohol správne fungovať, je potrebné, aby mal korektne nadefinované DataKeysNames.
- **PageSizerGridView** – pridáva možnosť pre používateľa vybrať si počet záznamov na stránku zoznamu. Tá je realizovaná pomocou DropDownList-u s v hlavičke a päte zoznamu. Jeho hodnoty sú nastaviteľné programátorom cez vlastnosť PageSizerValues. DropDownList je nastavený na AutoPostBack a pri zmene jednoducho zmení vlastnosť PageSize pôvodného ASP.NET GridView na požadovanú hodnotu.
- **SeparateEditGridView** – niekedy je dobré mať editáciu záznamu na inom formulári. Prácu v takejto situácii uľahčuje funkčnosť poskytovaná triedou SeparateEditGridView. Idea je, že po označení riadku pomocou tlačidla, či linku Select je používateľ presmerovaný na stránku definovanú vlastnosťou EditForm s identifikátorom riadku v QueryStringu.

5.1.3. AdvancedCheckBoxField

Štandardný ASP.NET CheckBoxField umožňuje ako svoje hodnoty používať iba typ boolean. To je problém napríklad pri data-bindingu na dáta z Microsoft SQL serveru, ktorý nemá boolean ako štandardný dátový typ a je namiesto neho nutné používať celočíselné typy. CheckBoxField si však s hodnotami 1 a 0 nevie poradiť. Práve tento jeho drobný nedostatok odstraňuje AdvancedCheckBoxField.

Interne AdvancedCheckBox iba mierne pozmeňuje metódu OnDataBindField pôvodného CheckBoxField-u a namiesto metódy Boolean.Parse používa na získavanie hodnoty z dát metódu Convert.ToBoolean, ktorá akceptuje aj výrazy ako "1" a "0".

5.1.4. AdvancedSqlDataSource

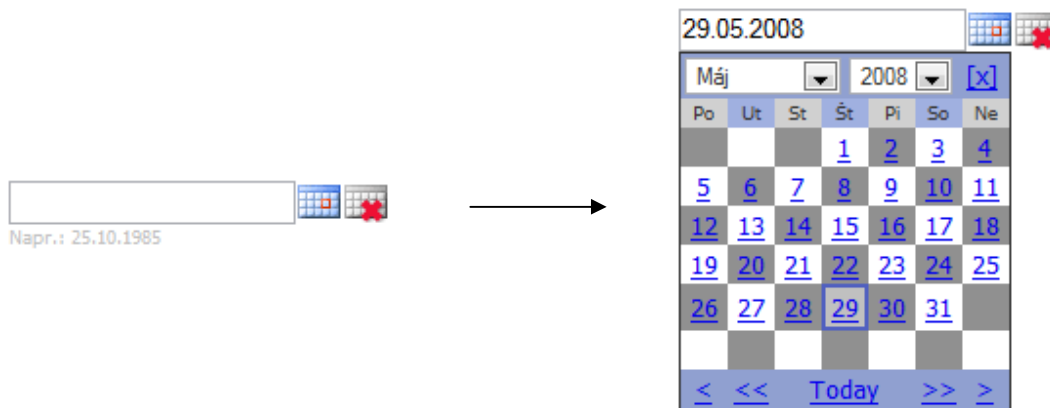
V podstate jeho jedinou, aj keď veľmi podstatnou výhodou oproti SqlDataSource je schopnosť používať Common Table Expressions (CTE) Microsoft SQL serveru na výber presne tých a len tých údajov, ktoré sú práve požadované ako aj ďalších informácií o danom výbere.

Medzi konkrétne služby poskytované AdvancedSqlDataSource patrí triedenie záznamov na strane SQL serveru, filtrovanie záznamov na strane SQL serveru, stránkovanie na strane SQL serveru a vytváranie sumárov (za stránku aj za výber) na strane SQL serveru. V skutočnosti je celá logika AdvancedSqlDataSource ukrytá v triede AdvancedSqlDataSourceView, ktorú používa namiesto štandardnej SqlDataSourceView. Tá pozmeňuje SQL dopyt tak, aby dosiahla požadovaný efekt. Na to sa používa trieda SelectCommandBuilder, ktorá ešte reprezentuje základný SQL dopyt na výber dát pomocou triedy SqlSelectQuery.

5.1.5. Calendar

ASP.NET control Calendar neposkytuje vždy presne tú funkčnosť, akú by sme od neho čakali, navyše zaberá zväčša dosť miesta na stránke. Emanager Calendar je k nemu veľmi vhodnou alternatívou. Využíva JavaScript na vykreslenie kalendára, z ktorého si používateľ ľahko vyberie konkrétny dátum.

Z hľadiska ASP.NET nie je tento control ničím prevratným, ide iba o spojenie niekoľkých Web alebo HTML controlov do jedného celku. Celá logika sa skrýva v JavaScript-e.



5.1.6. Menu

I keď menu je možné vytvoriť v ASP.NET viacerými spôsobmi, tak control Menu pridáva ďalší. Jeho hlavnou výhodou je množstvo dát, ktoré prenáša z klienta na server – celé menu sa totiž prenáša v mierne skomprimovanej podobe a samotné HTML sa generuje až pomocou JavaScriptu na klientovi. Navyše sa skript, s pomocou ktorého sa menu dostáva zo serveru na klientský prehliadač na klientovi cachuje, čím odpadáva dátová réžia pri ďalších requestoch. Obzvlášť pri veľkých aplikáciách, ktoré majú v menu veľa položiek sa tak môže rapídne znížiť objem prenášaných dát na klienta.

Vnútorne fungovanie Menu určite stojí za zmienku. JavaScript obsluhujúci a generujúci Menu na strane klienta je generovaný na požiadanie spolu s dátami v kompaktnej forme, z ktorých sa jednotlivé položky menu vytvoria. Tento skript sa posiela browseru spolu s odporúčaním na uloženie do cache na najbližšiu pol hodinu. Aby sa pri prípadnom odhlásení používateľa a prihlásení ďalšieho predišlo zobrazeniu pôvodného menu, má vygenerovaný súbor v názve aj identifikátor používateľa, ktorý si ho vyžiadal. Práve kvôli tomuto „náhodnému“ elementu v názve je obsluhovanie súborov realizované pomocou samostatného IHttpHandler-u (trieda MenuJavaScriptHandler), ktorý control Menu automaticky pridá do konfiguračného súboru webovej aplikácie, ak v ňom ešte nefiguruje.

5.1.7. NullableCheckBox

V niektorých situáciách je potrebné zadať do HTML prvku checkbox viac než len dve hodnoty (ÁNO / NIE). Takáto potreba vzniká najmä vo filtroch zoznamov, keď na daný príznak nechceme prihliadať. NullableCheckBox pridáva tretiu hodnotu (NULL) a renderuje sa ako HTML element select s tromi hodnotami.

Kvôli obmedzeniam hierarchie tried ASP.NET túto triedu nebolo vhodné odvodiť od žiadnej z tried ako CheckBox (podporuje iba bool hodnoty vlastnosti Selected), DropDownList alebo ListControl (obidve umožňujú data-binding svojich hodnôt, čo nie je v prípade NullableCheckBox žiaduce). Preto je odvodená od najbližšieho možného kandidáta – triedy CompositeControl a snaží sa čo najbližšie priblížiť zoznamom vlastností prvkom, ktorým je charakterom najviac podobná – CheckBox a DropDownList.

5.1.8. SmartLookup

Často je potrebné vybrať vo formulári jednu z hodnôt nejakého číselníka. Vhodnou alternatívou býva DropDownList, ale to iba do určitého počtu záznamov. Potom sa stáva pre používateľa prakticky nepoužiteľný. Práve v takýchto situáciách prichádza na rad SmartLookup, ktorý po zadaní vyhľadávacieho reťazca používateľom nájde vyhovujúce záznamy v databáze a zobrazí ich na výber pre používateľa. Celý proces sa odohráva cez ClientPostBack.

Programátor pri používaní nastaví SmartLookup-u polia, podľa ktorých sa bude vyhľadávať. SmartLookup potom vyhľadáva vo všetkých definovaných poliach buď pomocou podreťazcov (pri textových poliach) alebo na presnú zhodu (ostatné typy). Vlastnosť, ktorá ešte pri SmartLookup-e stojí za zmienku je to, ako funguje data-binding. SmartLookup sa k dátam nebinduje pri klasickom requeste, ale až keď príde ClientPostBack kvôli výsledkom vyhľadávania.

5.1.9. TabsView a Tab

Control TabsView je analógiou k ASP.NET controlu MultiView, s tým rozdielom, že umožňuje jednotlivý pohľad (záložku) vybrať používateľovi a taktiež umožňuje otvoriť viacero záložiek naraz. Funkciu triedu View (používaná v MultiView) plní trieda Tab.

Zobrazovanie a skrývanie záložiek je realizované jednou z dvoch metód – PostBack alebo JavaScript. Výhoda prvej je menej prenášaných dát zo serveru na klienta, výhoda druhej zas rýchle prepínanie bez nutnosti ďalšieho requestu. Zvoliť jednu z metód má vývojár možnosť prostredníctvom vlastnosti Mode.

Každá záložka ďalej umožňuje napojenie na zdroj dát, z ktorého zistí či nejaké záznamy obsahuje, prípadne koľko ich obsahuje. Túto informáciu potom môže zobrazit' v hlavičke zvýraznením nadpisu alebo uvedením počtu záznamov do zátvorky za nadpis. Toto správanie je možné ovplyvniť vlastnosťou RecordCountDisplayMode.

The diagram illustrates the transition from a simple tabbed interface to a more detailed one. At the top, there are two tabs labeled 'Tab1' and 'Tab2'. An arrow points down to a more detailed view of the tabs. The 'Tab1' tab is active and contains the text 'Content in Tab1' and a link 'Zatvorit'. Below this, there are five form fields with labels: 'Ulica:', 'Číslo:', 'Mesto:', 'PSČ:', and 'Krajina:'. Each field has a text input box and a small text label below it: 'Např.: Sofijská', 'Např.: 6', 'Např.: Košice', 'Např.: 04013', and 'Např.: 04013'. The 'Tab2' tab is also shown, containing the text 'Content in Tab2' and a text input box with the value '25.10.1985'. To the right of the input box are two small icons: a grid icon and a red star icon.

5.1.10. TubularView

Klasický GridView zobrazuje dvojrozmerný zoznam dát. Čo ale ak potrebujeme zobrazit' trojrozmerné dáta (hodnoty kartézskeho súčinu množín záznamov)? Presne pre tento prípad som vyvinul control TubularView, ktorý takýto prehľad poskytuje.

V praxi sa dá použiť napríklad na zobrazenie počet najazdených kilometrov a počet nehôd pre všetkých vodičov a všetky autá z nejakej skupiny záznamov. Trojrozmernosť TubularView treba nejakým spôsobom vtesnať do dvojrozmernej HTML stránky. Toto je realizované ukladaním jednotlivých položiek (najazdené kilometre, počet nehôd) každej dvojice auto – vodič horizontálne vedľa seba.

TubularView taktiež umožňuje zobrazovať sumy jednotlivých riadkov a stĺpcov v poslednom stĺpci a riadku, ako i celkovú sumu v pravej dolnej bunke výslednej tabuľky.

Pre dáta, na ktoré sa TubularView binduje musia platiť tieto podmienky:

- Musia byť korektne zadefinované kľúče vertikálnych a horizontálnych záznamov pomocou vlastností `HorizontalDataValueField` a `VerticalDataValueField`. Na základe nich TubularView určuje, kedy má prejsť do generovania ďalšieho riadku tabuľky.
- Dáta musia byť zoradené podľa stĺpca `VerticalDataValueField` a potom podľa stĺpca `HorizontalDataValueField`. Samotný TubularView totiž dáta nijak neusporiada a preto je toto potrebné pre rýchle sekvenčné spracovanie záznamov.
- Aj keď nejde o nutnosť, spravidla je žiaduce, aby boli záznamy zoskupené (SQL príkaz `group by`) podľa `VerticalDataValueField` a `HorizontalDataValueField`.

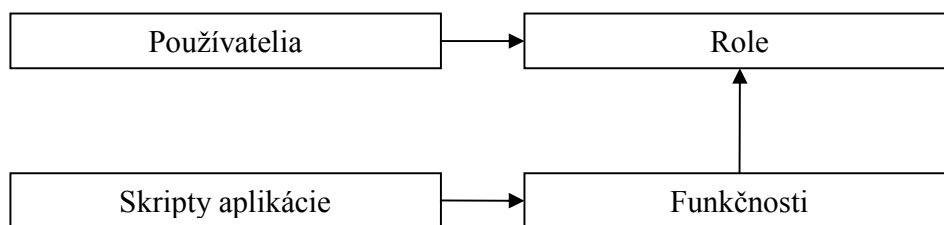
Praktickými funkciami TubularView sú aj skrývanie nulových a prázdnych riadkov a/alebo stĺpcov a zobrazovanie checkbox-ov v každom riadku a/alebo stĺpci.

5.2. Emanager Security

Emanager Security je knižnica zahŕňajúca rozšírenie a nahradenie systémov ASP.NET autentifikácie a autorizácie. Jej dva najpodstatnejšie prínosy sú zavedenie hladiny funkčností (functionalities) ako medzivrstvu medzi rolami a skriptami a odsunutie úložiska autentifikačných a autorizačných dát z konfiguračných XML súborov do SQL databázy.

5.2.1. Autorizačný model systému Emanager

Štandardný ASP.NET autorizačný model pozostáva z troch vrstiev: používatelia, role a skripty aplikácie. Tento model je pre malé aplikácie vyhovujúci, pre veľké však už často na hranici použiteľnosti. Ak je totiž istej skupine používateľov potrebné povoliť prístup do niektorej časti aplikácie, znamená to zväčša povoliť prístup až do niekoľkých desiatok skriptov, čo nie je pre administrátora veľmi pohodlné, ani prehľadné. Navyše sa tým neúnosne zvyšuje veľkosť (a tým aj znižuje prehľadnosť) konfiguračných súborov. Práve funkčnosti sú tým mechanizmom, ktorý zoskupuje jednotlivé skripty do väčších celkov, ktoré je možné naraz povoliť alebo zakázať jednotlivým roliam.

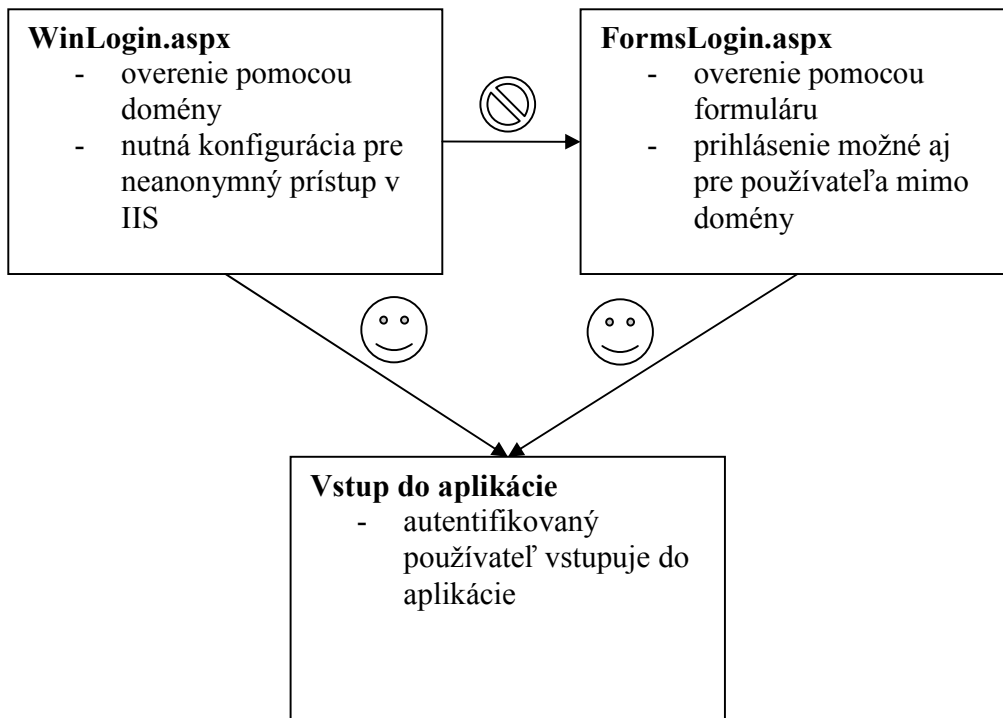


Ukladanie autorizačných dát (zoznamy používateľov, rolí, funkčností a ich vzájomných vzťahov) v SQL databáze odstraňuje problém s veľkosťou konfiguračných súborov a tiež uľahčuje prácu pri migrovaní existujúcich aplikácií zo starších platforiem (napr. ASP), ktoré majú zväčša tieto dáta uložené práve v relačnej databáze.

5.2.2. Dvojitá autentifikácia

Množstvo organizácií a podnikov má vo svojej sieti zavedenú doménovú politiku pomocou a tak by bolo škoda, aby aplikácia postavená na systéme Emanager nepodporovala automatické overovanie používateľov na základe ich doménového mena. Zároveň je však nutné ponechať klasické formulárové prihlasovanie pre tých používateľov, ktorý nie sú zaradený do domény.

ASP.NET však nič ako autentifikáciu dvoma rôznymi spôsobmi nepozná, preto bolo potrebné vyvinúť systém, ktorý z pohľadu používateľa poskytuje tieto dve metódy prihlásenia, z hľadiska .NET platformy sa však tvári ako jeden. Ako základ slúži ASP.NET forms authentication.



Celý proces prebieha v 2 krokoch:

1. Ako hlavná autentifikačná stránka je nastavená WinLogin.aspx. Keď sa používateľovi na ňu podarí prísť, skript ho autentifikuje. Táto stránka je však v IIS nastavená tak, že nepodporuje anonymný prístup a preto sa na ňu dostanú len používatelia overení voči doméne. Ostatní sú presmerovaní na stránku FormsLogin.aspx
2. Zobrazí klasický prihlasovací formulár, ktorý používateľa overí na základe zadaného prihlasovacieho mena a hesla.

5.2.3. Hierarchia tried

Knižnica Emanager Security obsahuje triedy, ktoré možno rozdeliť nasledovne:

- Statické triedy priamo používané vývojárom, ktoré abstrahujú jednotlivé časti autentifikácie a autorizácie:
 - **Authentication** – trieda zoskupujúca logiku na overovanie používateľa. Je vhodná najmä kvôli dvojitej autentifikácii popísanej v predošlej kapitole.
 - **Authorization** – dopĺňa ASP.NET triedu System.Net.Authorization pre potreby Emanager autorizácie.

- **Functionalities** – využíva sa pre prácu s funkčnosťami, ako vytváranie a mazanie funkčností, pridávanie funkčností do rolí apod.
- **Membership** – nahrádza ASP.NET triedu System.Web.Security.Membership hlavne z dôvodu väčšieho počtu položiek pri používateľovi systému Emanager (najpodstatnejšou je doménové prihlasovacie meno).
- Triedy poskytovateľov (providers), ktoré implementujú konkrétnu časť funkčnosti Emanager Security:
 - **AuthorizationProvider** – implementuje Emanager autorizácie. Trieda sa štandardne využíva v statickej triede Authorization.
 - **FunctionalityProvider** – implementuje podporu pre Emanager funkčnosti. Trieda je štandardne využívaná v statickej triede Functionalities.
 - **MembershipProvider** – implementuje funkčnosť okolo používateľov. Je štandardne využívaná v statickej triede Membership.
 - **XmlSiteMapProvider** – je triedou z trošku iného konca. Implementuje vytváranie mapy stránok pre daného používateľa na základe práv, ktoré má pridelené.
- HTTP moduly, ktoré ovplyvňujú spracovanie prichádzajúceho webového dopytu:
 - **AuthorizationModule** – odmieta alebo priznáva používateľovi prístup na požadovanú stránku na základe nastavení rolí, práv, výnimiek atp.

5.3. Emanager Error Handling

Spoliehať sa na to, že všetky chyby aplikácie vám podrobne nahlásia používatelia je prinajmenšom naivné. Preto je dobré mať po ruke mechanizmus, ktorý vám hlásenie o chybách zabezpečí. Takýto mechanizmus zastrešuje knižnica Emanager Error Handling.

Každá výnimka, ktorá sa nezachytila inde v aplikácii sa zachytí v udalosti Application_Error triedy Global application class (Global.asax). Práve tam ju spracuje trieda Emanager.ErrorHandling.ErrorLogging. Táto trieda nezachytené výnimky ukladá do súboru na disku a posieľa v pravidelných intervaloch na nakonfigurované emailové adresy.

Zasielanie v určitých intervaloch je potrebné z toho dôvodu, aby systém pri vážnejšej chybe, na ktorú narazí v krátkom čase veľa používateľov neposielal desiatky až stovky mailov každú minútu.

Ukladanie na disk a nie do pamäte, či databázy som zas zvolil z dvoch dôvodov. Pri výpadku serveru, na ktorom beží webová aplikácia by sa história neodchytených výnimiek nenávratne stratila (v prípade ukladania do pamäte) alebo by nebolo pri výpadku databázového serveru možné chyby ukladať (v prípade ukladania do databázy). Celé fungovanie hlásenia chýb sa konfiguruje v štandardnom konfiguračnom súbore (zväčša je to súbor Web.config). Tu je možné nakonfigurovať interval posielania emailových správ s prípadnými chybami, ako aj ich príjemcov.

5.4. Emanager Auto Updater

Emanager Auto Updater je jednoduchá Windows služba, ktorá synchronizuje dvojice adresárov medzi sebou. Je tak možné ľahko distribuovať zmeny v existujúcej aplikácii do ďalších prostredí.

Služba aktualizuje cieľový adresár vždy pri zmene zdrojového, ale aj v definovaných časových intervaloch.

Konfigurácia prebieha jednoducho pomocou XML súboru a pre jednoduchú inštaláciu som vytvoril setup project.

6. Záver

Na systéme Emanager som pracoval už viac než rok a čas investovaný do neho určite neľutujem. Pri vývoji nových a vylepšovaní už existujúcich webových aplikácií som neraz siahol po niektorej z jeho súčastí a nemusel som tak pracne obchádzať niektoré z nedokonalostí platformy ASP.NET.

Navyše som sa zvýšeným záujmom o túto platformu dozvedel veľa vecí, pre ktoré ju teraz oproti ostatným webovým platformám absolútne preferujem. Svoju snahu naučiť sa o tejto technológii čo najviac som v januári tohto roku korunoval zložením skúšok firmy Microsoft a dosiahnutím titulu Microsoft Certified Technology Specialist: .NET Framework 2.0: Web applications.

7. Použitá literatura

- Tony Northrup, Shawn Wildermuth a Bill Ryan (2006): Microsoft .NET Framework 2.0 – Application Development Foundation
- Glenn Johnson and Tony Northrup (2006): Microsoft .NET Framework 2.0 – Web-Based Client Development
- Dino Esposito (2006): Programming Microsoft ASP.NET 2.0 Applications: Advanced Topics
- Kalen Delaney (2006): Inside Microsoft SQL Server 2005: The Storage Engine
- Solid Quality Learning (2006): Microsoft SQL Server 2005: Database Essentials
- Steve Krug (2003): Web design – Nenuťte uživatele přemýšlet!