

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

UŽIVATELSKÝ NÁVRH WEBOVÝCH DOTAZNÍKŮ

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ŠTEFAN MIKULA

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

UŽIVATELSKÝ NÁVRH WEBOVÝCH DOTAZNÍKŮ

USER DESIGN OF WEB QUESTIONNAIRES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ŠTEFAN MIKULA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. KŘIVKA ZBYNĚK, Ph.D.

BRNO 2008

Abstrakt

Úlohou bakalárskej práce je analýza súčasných možností vytvárania webových dotazníkov a návrh a implementácia webového vizuálneho editoru, pomocou ktorého je možné online vytvárať a následne používať webové dotazníky. Táto úloha pozostáva z dvoch komponent, webového rozhrania, ktorý tvorí kombinácia Javascriptu a HTML a serverového parseru, ktorý analyzuje odoslaný kód a dynamicky vytvára formuláre na serverovskej strane vrátane databázovej štruktúry. Serverová časť je implementovaná v prostredí ASP.NET.

Kľúčové slová

Webový formulár, dotazník, WWW, WYSIWYG editor, FCKeditor, JavaScript, HTML, HTML editor, .NET, ASP.NET, C#, MS SQL

Abstract

The aim of this bachelor thesis is to analyze the contemporary methods of web questionnaires development and to design and implement a visual editor for designing such questionnaires. This task consist of two components, a web interface, which is a combination of JavaScript and HTML and a server side, which parses the submitted design and generates a working web questionnaire. The server side component is implemented in ASP.NET technology.

Keywords

Web Form, Questionnaire, WWW, WYSIWYG editor, FCKeditor, JavaScript, HTML, HTML editor, .NET, ASP.NET, C#, MS SQL

Citovanie

Štefan Mikula: Užívateľský návrh webových dotazníkov, bakalárska práca, Brno, FIT VUT v Brně, 2008

Užívateľský návrh webových dotazníkov

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením

Ing. Zbyňka Křivky, PhD.

Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Štefan Mikula
14.5.2008

Pod'akovanie

Rád by som poďakoval všetkým, ktorí mi pomohli pri realizácii bakalárskej práce. Menovite môjmu vedúcemu bakalárskej práce Ing. Zbyňkovi Křivkovi PhD. a ostatným za podporu pri tvorbe tejto práce.

© Štefan Mikula, 2008.

Táto práca vznikla ako školské dielo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práca je chránená autorským zákonom a jej použitie bez udelenia oprávnenia autorom je nezákonné, s výnimkou definovaných prípadov.

1	Úvod.....	2
2	Použité technológie.....	3
2.1	HTML.....	3
2.1.1	Jazyk HTML.....	3
2.1.2	Dotazníky v HTML.....	4
2.1.3	Prvky dotazníkov.....	4
2.1.3.1	Typy ovládacích prvkov.....	5
2.1.3.2	Popis elementov.....	7
2.2	JavaScript.....	11
2.2.1	JavaScript a HTML dotazníky.....	11
2.2.2	Document Object Model (DOM).....	12
2.3	ASP.NET.....	14
2.3.1	Základné charakteristiky.....	15
2.3.2	MS SQL.....	17
3	Analýza a návrh riešenia.....	18
3.1	Popis problému.....	18
3.2	Teoretický úvod.....	18
3.3	Závislosti prvkov.....	21
3.3.1	Validácia prvkov.....	22
3.4	JavaScript WYSIWYG editor.....	22
3.5	Spracovanie formulára.....	23
3.6	Iné jazyky popisujúce webové dotazníky.....	24
4	Implementácia.....	26
4.1	Vizuálny editor.....	26
4.1.1	Kompatibilita prehliadačov.....	28
4.2	HTML parser.....	28
4.3	Databáza MS SQL.....	29
4.4	Konfigurácia a integrácia.....	30
4.5	Zhrnutie.....	31
5	Záver.....	32

1 Úvod

Svet internetu sa stále vyvíja a mení. World Wide Web, čiže skratka www pod akou väčšina ľudí pozná túto časť internetu, je najvyužívanejšia časť internetu koncovými užívateľmi. Nové webové stránky a aplikácie pribúdajú každý deň a s nimi aj nové požiadavky užívateľov. Vývojári webových aplikácií snažia naplniť použitím nových technológií a techník, či postupov, aby v krátkom čase, efektívne a lacno splnili požiadavky zákazníka alebo klienta.

Internet prechádza rôznymi zmenami. Zo začiatku boli užívatelia len pasívni pozorovatelia, jediná možná tvorivá činnosť alebo spôsob komunikácie bol pomocou elektronickej pošty alebo diskusných skupín. Postupne vznikali rôzne komunitné webové portály, fóra, blogy a iné systémy, pomocou ktorých môžu užívatelia komunikovať ako aj prezentovať sa. Postupne prechádza do rúk užívateľov čoraz viac nástrojov a už nie sú iba pozorovateľmi webu ale sú ich spoluvorcami (tzv Web 2.0).

Táto práca sa zaoberá systémom, pomocou ktorého sú užívatelia schopní vytvárať vlastné webové dotazníky. Tieto dotazníky sa dajú ďalej používať na rôzne ankety, vedomostné kvízy alebo zber osobných či iných dát. Nasleduje popis kapitoly.

V druhej kapitole sú predstavené použité technológie pri tvorbe aplikácie. Predstavený je jazyk HTML a detailne sú popísané formuláre v tomto jazyku, práca s nimi a ich ovládacie prvky. Ďalej sú popísané technológie JavaScript, ASP.NET a MS SQL.

Tretia kapitola sa zaoberá analýzou a návrhom riešenia celej aplikácie. Predstavené sú závislosti medzi ovládacími prvkami formulárov a tieto sú popísané formálnym spôsobom. Ďalej sú rozobraté možnosti súčasných JavaScriptových WYSIWYG editorov a ich využitie pre túto aplikáciu. Kapitola uzatvára stručný úvod do ďalších jazykov popisujúcich webové dotazníky, ich prehľad, výhody a možné využitie v budúcnosti.

Štvrtá kapitola pojednáva o spôsobe implementácie, predstavené sú jednotlivé komponenty aplikácie, ako klientská, tak serverová časť. Rozobrané sú problémy s kompatibilitou a možnosť výstavby formulárov v rôznych internetových prehliadačoch. Záver kapitoly pojednáva o konfiguračných a integračných možnostiach aplikácie.

2 Použité technológie

2.1 HTML

Jazyk HTML je značkovací jazyk pre popis webových dokumentov. Skratka HTML znamená HyperText Markup Language, v preklade hyper-textový značkovací jazyk, čo popisuje primárne určenie jazyka, tj. popis webových dokumentov navzájom prepojených odkazmi.

2.1.1 Jazyk HTML

V roku 1989 v spoločnosti CERN spolupracovali Tim Berners-Lee a Robert Caillau na informačnom systéme pre zdieľanie dokumentov pre CERN (Európska Organizácia pre Jadrový Výskum), so sídlom v Ženeve vo Švajčiarsku. V tej dobe sa na popis dokumentov obvykle používal TeX, Postscript alebo SGML¹. Situácia si však vyžadovala jednoduchší a použiteľnejší spôsob popisu dokumentov a tak bol v roku 1990 navrhnutý jazyk HTML a protokol pre jeho prenos po sieti HTTP (HyperText Transfer Protokol). V nasledujúcich rokoch došlo k veľkému rozvoju web a bolo nutné pre HTML definovať štandardy. Prvý dostupný dokument popisujúci jazyk vyšiel v roku 1991 pod názvom *HTML Tags*, popisoval 22 elementov, 13 z nich sa existuje v poslednej špecifikácii jazyka. Postupne jazyk prechádzal ďalšími špecifikáciami, kde v tomto procese zohrala hlavnú úlohu organizácia W3C (World Wide Web Consortium). Posledná verzia špecifikácie HTML 4.01 vyšla na konci roku 1999. [4]

Jazyk je charakterizovaný množinou značiek, alebo „tagov“ (z anglického tag, tj. označiť, označovať, značka). Tieto značky sú väčšinou párové, pričom rozlišujeme začiatkové a koncové značky, ktoré sú väčšinou párové. Rozlišujeme teda počiatkové a koncové značky, počiatkové značky môžu obsahovať atribúty. Značky uzatvárajú textové časti dokumentu a tým určujú význam textu (sémantiku). Názvy jednotlivých značiek uzatvárajú uhlové zátvorky („<“ a „>“). Časť dokumentu ohraničená týmito značkami tvorí tzv. element (prvok) dokumentu. Súčasťou obsahu elementu môžu byť ďalšie vnorené elementy. Atribúty sú doplnujúce informácie, ktoré upresňujú vlastnosti elementu.

¹ SGML - SGML je metajazyk pre popis značkovacích jazykov, hlavne tých, ktoré sa používajú v elektronických dokumentoch, manažmente dokumentov a publikovaní dokumentov. HTML je príkladom jazyka definovanom pomocou SGML.

Počiatok SGML siaha do roku 1980 a odvtedy je tento jazyk pomerne ustálený. Je to jazyk veľmi bohatý na možnosti a vlastnosti a zároveň je flexibilný. Ale práve vďaka svojej flexibilita a určitej úrovni zložitosti sa nepodarilo presadiť tento jazyk do rôznych aplikačných prostredí, vrátane World Wide Web.

2.1.2 Dotazníky v HTML

Dotazník je výskumný nástroj pozostávajúci z niekoľkých otázok a iných podnetov za účelom zberu informácií od respondentov. [1] Webové dotazníky sa s expanziou internetu stali obľúbeným nástrojom pre zber dát od užívateľov.

Dotazník v HTML je taká časť HTML dokumentu, ktorá obsahuje okrem normálneho obsahu, teda značiek a textu, aj špeciálne elementy, ktoré nazývame *ovládacie prvky* (*controls* - zaškrťavacie tlačítko, rôzne menu, vstupné textové polia a iné) a *označenia* (*labels*) týchto ovládacích prvkov. Užívateľia väčšinou vyplňujú dotazník menením hodnôt jednotlivých ovládacích prvkov (zadanie textu, vybratie položky z menu atď), predtým než dotazník odošlú na spracovanie napr. na webový server, emailový server atď.

2.1.3 Prvky dotazníkov

Dokument špecifikácie webových formulárov organizácie W3C (World Wide Web Consortium), ktorá pôsobí ako štandardizačná inštitúcia vo webových technológiách, presne popisuje vo svojom dokumente HTML 4.01 Specification [2] v sekcii 17 syntax a sémantiku tvorby webových formulárov. Tento dokument je odporúčanou špecifikáciou (W3C Recommendation) od roku 1999.

Každý webový dotazník pozostáva aspoň z jedného z nasledujúcich typov dotazov:

- *otvorené otázky* (*text fields*) – umožňuje užívateľovi zadať ľubovlnú textovú hodnotu, či už reťazec alebo číslo,
- *uzavreté otázky* (*selection fields*) – užívateľ si vyberá jednu alebo viacero z N možných hodnôt, tieto hodnoty môžu byť zobrazené rôznymi formami (viď nižšie).

Spôsob reprezentácie týchto dvoch základných typov dotazov v jazyku HTML (HyperText Markup Language) je niekoľko. Otvorené otázky je možné reprezentovať ako:

- *jednoriadkové textové pole* (*text field*) - zapisovaný reťazec môže mať zhora ohraničenú dĺžku,
- *viacriadkové textové pole* (*text area*) - zapisovaný reťazec nemá z pravidla zhora ohraničenú dĺžku, môže byť zapísaný na viacerých riadkoch.

Uzavreté otázky môžeme v jazyku HTML reprezentovať ako:

- *vyskakovacie menu* (*drop-down list*) - užívateľ vyberá jednu možnosť z N
- *zoznam položiek* (*list box*) - užívateľ vyberá 0 až M možností z N

- *rádiové tlačítko (radio button)* - stav zapnutý/vypnutý
- *zoznam rádiových tlačítok (radio button group)*- užívateľ vyberá jednu možnosť z N
- *zaškrtávacie tlačítko (checkbox)* - stav zapnutý/vypnutý
- *zoznam zaškrtávacích tlačítok (checkbox list)* - užívateľ vyberá 0 až M možností z N

Nasleduje popis ovládacích prvkov dotazníkov tak ako ich definuje špecifikácia. [2]

2.1.3.1 Typy ovládacích prvkov

HTML definuje nasledujúce typy ovládacích prvkov:

Tlačítka (buttons)

Autori môžu vytvárať tri druhy tlačítok:

- *odosielacie tlačítka (submit buttons)*: Pri aktivácii, odosielacie tlačítko odošle dotazník. Jeden dotazník môže obsahovať viacero odosielacích tlačítok.
- *nulovacie tlačítka (reset buttons)*: Pri aktivácii nulovacie tlačítko vynuluje všetky ovládacie prvky dotazníku a nastaví ich hodnoty na ich pôvodnú hodnotu.
- *všeobecné tlačítka (push buttons)*: Všeobecné tlačítka nemajú implicitné správanie. Každé všeobecné tlačítko môže mať asociované svoje *atribúty udalostí (event)* s klientským skriptami. Keď nastane nejaká udalosť, napríklad užívateľ stlačí tlačítko alebo ho pustí, zavolá sa asociovaný skript.

Tlačítka možno vytvárať pomocou elementu `BUTTON` alebo elementu `INPUT`. Element `BUTTON` poskytuje však viac vykreslovacích možností pri tvorbe tlačítka.

Zaškrtávacie políčka (checkboxes)

Zaškrtávacie políčka sú prepínače stavov zapnutý/vypnutý a môžu byť prepnuté užívateľom. Políčko je „zapnuté“ keď atribút `checked` elementu je nastavený. Pri odosielaní dotazníku, sa odošlú na spracovanie iba tie zaškrtávacie tlačítka, ktoré sú zapnuté.

Viacero zaškrtávacích políčok môže zdieľať rovnaké *ovládacie meno (control name)*. Takto umožňujú zaškrtávacie políčka užívateľom vybrať niekoľko hodnôt pre rovnakú vlastnosť. Element `INPUT` sa používa na vytvorenie ovládacieho prvku typu zaškrtávacie políčko.

Rádiové tlačítka (radio buttons)

Rádiové tlačítka sú podobné zaškrťovacím políčkam, okrem situácie, keď viacero rádiových tlačítok zdieľa rovnaké *ovládacie meno* (*control name*), vtedy sa navzájom vylučujú (tvoria disjunktnú množinu): keď je jedno nastavené do stavu „zapnutý“, všetky ostatné sa nastavujú do stavu „vypnutý“. Element `INPUT` sa používa na vytvorenie ovládacieho prvku typu rádiové tlačítko.

Ak ani jedno rádiové tlačítko z množiny tlačítok, ktoré zdieľajú rovnaké ovládacie meno nie je implicitne nastavené na hodnotu „zapnutý“, nastavenie jedného z ovládacích prvkov na „zapnutý“ užívateľským agentom² nie je definované.

Keďže správanie užívateľských agentov sa líši, autori by sa mali uistiť, že v každej množine rádiových tlačítok nastavili práve jedno implicitne na hodnotu „zapnutý“.

Menu (menus)

Menu ponúkajú užívateľovi možnosti, z ktorých si môže vybrať. Element `SELECT` v kombinácii s elementmi `OPTGROUP` a `OPTION` vytvára menu.

Textový vstup (textinput)

Autori môžu vytvárať dva typy ovládacích prvkov, ktoré umožňujú zadávať text. Element `INPUT` vytvára jednoriadkový vstupný ovládací prvok, element `TEXTAREA` vytvára viacriadkový vstupný ovládací prvok. V oboch prípadoch sa zadaný text stáva *aktuálnou hodnotou* ovládacieho prvku.

Výber súboru (file select)

Tento ovládací prvok umožňuje užívateľovi vybrať súbor takže jeho obsah môže byť odoslaný s dotazníkom. Element `INPUT` sa využíva na vytvorenie ovládacieho prvku typu výber súboru.

Skryté ovládacie prvky (hidden controls)

Autori môžu vytvoriť ovládacie prvky, ktoré nie sú vykreslené ale ich hodnoty sú odoslané s dotazníkom. Autori vo všeobecnosti využívajú tento typ ovládacích prvkov na ukladanie informácií medzi jednotlivými komunikáciami medzi serverom a klientom, ktoré by sa v opačnom prípade stratili kvôli bezstavovej povahe protokolu HTTP. Element `INPUT` sa využíva na vytvorenie skrytého ovládacieho prvku.

Objektové ovládacie prvky (object controls)

Autori môžu do dotazníkov vkladať generické objekty tak, že asociované hodnoty sú odoslané spolu s ostatnými ovládacími prvkami. Element `OBJECT` sa využíva na vytvorenie objektového ovládacieho prvku.

2 Užívateľský agent - zobrazovací nástroj, z pravidla sa jedná o webový prehliadač (web browser) určitého typu a konkrétnej implementácie (Mozilla, Firefox, Internet Explorer, Opera, Safari, Konqueror atď)

Elementy, ktoré vytvárajú ovládacie prvky, sú zväčša umiestnené vo vnútri elementu FORM, ale môžu sa vyskytovať aj mimo FORM elementu. V tomto prípade sa používajú na stavbu užívateľských rozhraní. Ovládacie prvky, ktoré sú umiestnené mimo dotazníka nemôžu byť *úspešne odoslanými ovládacími prvkami*.

2.1.3.2 Popis elementov

Element **INPUT** sa využíva na výstavbu väčšiny ovládacích prvkov v dotazníkoch. Príklad elementu podľa špecifikácie [2]:

```
<INPUT type="text" name="first_name" >
```

Počiatková značka je vyžadovaná, ukončovacia značka je zakázaná.³

Atribút `type` (názov) nie je povinný a ak chýba, implicitná hodnota je `text`.

Atribút `name` (názov) je povinný pre všetky okrem typu `submit` (odošli) a `reset` (vynuluj).

Zoznam atribútov definovaných elementom:

```
value, checked, disabled, readonly, size, maxlength, src, alt,  
usemap, ismap, tabindex, accesskey, onfocus, onblur, onselect,  
onchange, accept
```

Zoznam atribútov definovaných inde⁴:

```
id, class, lang, dir, title, style, alt, align, accept, readonly,  
disabled, tabindex, accesskey, usemap, ismap, onfocus, onblur,  
onselect, onchange, onclick, ondblclick, onmousedown, onmouseup,  
onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup
```

Definícia atribútov

`type (typ) = { text, password, checkbox, radio, submit, reset, file, hidden, image, button }`

Tento atribút špecifikuje *typ* ovládacieho prvku. Implicitná hodnota je `"text"`.

`name (názov)`

Tento atribút priradzuje *ovládacie meno* danému prvku.

3 Syntax elementu INPUT ako aj iných elementov upravuje špecifikácia XHTML 1.0, tento element je ukončovaný znakmi „/>“ a nie iba „>“ a tak približuje obsah XHTML dokumentu XML štandardu. [5]

4 Atribúty, ktoré sú definované v iných častiach špecifikácie HTML a nie priamo týmto elementom. [7]

`value` (hodnota)

Tento atribút špecifikuje implicitnú hodnotu ovládacieho prvku. Je to voliteľný atribút okrem prípadu, keď ako typ je definovaný "radio" (rádiové tlačítko) alebo "checkbox" (zaškrávacie políčko).

`size` (veľkosť)

Tento atribút hovorí užívateľskému agentovi aká je implicitná šírka tohto ovládacieho prvku. Šírka je udávaná v pixloch⁵ okrem prípadov keď typ (`type`) je "text" alebo "password" (heslo). V tom prípade táto hodnota udáva počet znakov.

`maxlength` (maximálna dĺžka)

Ak je hodnota atribútu `type` "text" alebo "password" tak tento atribút udáva maximálny počet znakov, ktoré užívateľ môže zadať. Toto číslo môže prekročiť hodnotu definovanú atribútom `size`, v tomto prípade užívateľský agent by mal poskytnúť posúvanie. Hodnota tohto atribútu je explicitne neobmedzená.

`checked` (zvolený)

Ak je typ ovládacieho prvku zadefinovaný ako "radio" alebo "checkbox", táto pravdivostná hodnota udáva, že prvok je v polohe "zapnutý". Užívateľský agent ignoruje tento atribút pri iných typoch ovládacích prvkov.

`src` (zdroj)

Ak má atribút `type` hodnotu „image“ tak tento atribút udáva adresu obrázku, ktorý je použitý ako dekorácia v grafickom odosielačom tlačítku.

Element `BUTTON` sa používa na vytvorenie generického tlačítka, jedná sa o podobný element ako v prípade `INPUT` s typom definovaným ako „button“ ale element `BUTTON` má vyššiu vykreslovaciu schopnosť, t.j. môže mať obsah.

Počiatočná a koncová značka sú vyžadované.

Zoznam atribútov definovaných elementom:

`name, value, type`

Zoznam atribútov definovaných inde:

`id, class, lang, dir, title, style, disabled, tabindex, accesskey, onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup`

5 Pixel - základná jednotka v zobrazovaní v počítačovej grafike, predstavuje jeden zobraziteľný bod.

Definícia atribútov

name (názov)

Tento atribút priradzuje *ovládacie meno* danému prvku.

value (hodnota)

Tento atribút špecifikuje implicitnú hodnotu tlačítka.

type (typ) = {submit, button, reset}

Tento atribút deklaruje typ tlačítka. Možné hodnoty sú:

- submit: Vytvorí odosielacie tlačítko. Toto je implicitná hodnota.
- reset: Vytvorí *mulovacie tlačítko*.
- button: Vytvorí *všeobecné tlačítko*.

Elementy SELECT, OPTGROUP a OPTION sa používajú na tvorbu vyskakovacích (pull-down) a výberových (list box) menu. Element SELECT:

Počiatočná a koncová značka sú vyžadované.

Musí obsahovať aspoň jeden element OPTION.

Zoznam atribútov definovaných elementom:

name, size, multiple, onfocus, onblur, onchange

Zoznam atribútov definovaných inde:

id, class, lang, dir, title, style, accesskey, onfocus, onblur, onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup

Definícia atribútov

name (názov)

Tento atribút priradzuje *ovládacie meno* danému prvku.

size (veľkosť)

Ak sa element SELECT zobrazí ako posúvateľný zoznam (listbox), tak táto hodnota udáva počet riadkov v tomto zozname, ktoré sú viditeľné užívateľovi.

multiple (viacero)

Ak je tento atribút nastavený, element podporuje viacnásobný výber. V opačnom prípade povoľuje element SELECT vybrať iba jednu položku zo zoznamu.

Element OPTGROUP umožňuje logicky zhlukovať možnosti výberu. Toto je obzvlášť vhodné, pokiaľ je zoznam možností príliš dlhý, užívateľ sa tak jednoduchšie zorientuje v ponúkaných možnostiach.

Element `OPTGROUP` musí byť priamym potomkom elementu `SELECT`, vnorené skupiny nie sú povolené.

Nula a viac hodnôt môžu byť predvyznačené pre užívateľa. Vyznačenie sa zaistí zahrnutím atribútu `selected` do deklarácie elementu `OPTION`. Jeden zoznam môže obsahovať viac predvyznačených možností ak element `SELECT` obsahuje atribút `multiple`, sú implicitne vyznačené všetky možnosti s atribútom `selected`.

Definícia atribútov

`label` (označenie)

Tento atribút dáva označenie tejto skupine výberu.

Element `OPTION`

Počiatočná a koncová značka sú vyžadované.

Definícia atribútov

`selected` (vyznačený)

Tento atribút udáva pravdivostnú hodnotu, či je táto možnosť výber predvyznačená.

`value` (hodnota)

Tento atribút predstavuje *implicitnú hodnotu* prvku. Ak tento jazyk nie je prítomný, *implicitná hodnota* prvku je nastavená na obsah prvku `OPTION`.

`label` (označenie)

Tento atribút poskytuje autorom vytvoriť kratšie označenie tohto prvku.

Element `TEXTAREA` sa používa na vytvorenie viacriadkového *textového ovládacieho prvku*.

Implicitná hodnota prvku je použitá ako implicitný text v prvku.

Počiatočná a koncová značka sú vyžadované.

Definícia atribútov

`name` (názov)

Tento atribút priradzuje *ovládacie meno* danému prvku.

`rows` (riadky)

Tento atribút udáva počet riadkov, ktoré sú viditeľné užívateľovi. Užívateľ by mal byť schopný zadať viac riadkov textu ako toto číslo, takže užívateľský agent by mal poskytovať posúvací mechanizmus.

`cols` (stĺpce)

Tento atribút udáva viditeľnú šírku ovládacieho prvku založenú na priemernej šírke jedného znaku. Užívatelia by mali byť schopní zadať viac znakov do jedného riadku ako je toto číslo, užívateľský agent by mal poskytnúť posúvací mechanizmus.

2.2 JavaScript

JavaScript je programovací jazyk, ktorý sa používa v internetových stránkach. Zapisuje sa buď priamo do HTML kódu alebo môže byť uložený v samostatných súboroch (väčšinou s príponou `.js`) a pripojený k webovej stránke. JavaScript je klientský programovací jazyk, tj. celý program sa odosiela spolu so stránkou na stranu klienta a potom je týmto klientom, prehliadačom, spracovávaný, interpretovaný⁶.

Existujú aj iné skriptovacie jazyky využívané vo webových stránkach, napríklad VB Script alebo JScript, sú však málo používané. JavaScript bol vyvinutý spoločnosťou NetScape v kooperácii so spoločnosťou Sun Microsystems a bol uvedený na trh v decembri roku 1995 [6]. Jedná sa o dynamický, typovo slabý, prototypovo orientovaný jazyk. Napriek svojmu menu je tento jazyk málo podobný jazyku Java, aj keď oba vychádzajú zo spoločnej syntaxe jazyka C a JavaScript kopíruje veľa menných priestorov Javy. Kľúčový dizajn tohto jazyka bol odvodený od programovacieho jazyka Self. [7]

2.2.1 JavaScript a HTML dotazníky

JavaScript ako klientský programovací jazyk je dôležitým nástrojom v súčasnom svete World Wide Web na manipuláciu s webovými dokumentami. Poskytuje programový prístup ku všetkým elementom dokumentu, dokáže dynamicky vytvárať, mazať alebo meniť elementy a ich atribúty. Pri webových dotazníkoch môže byť použitý na kontrolu zadaných hodnôt, porovnávanie určitých elementov (napríklad potvrdzovanie zadaného hesla atď), znepřístupnenie alebo sprístupnenie určitých ovládacích prvkov, zmena viditeľnosti ovládacích prvkov a iné. JavaScript môže byť použitý aj na vytváranie komplexných programov ako sú napríklad textové či iné editory. V tejto práci je JavaScript použitý na implementáciu vizuálneho editoru na výstavbu webových dotazníkov (viď kapitola 3).

6 Protikladom klientských skriptov sú skripty serverové, ktoré sa vykonávajú na serveri a na stranu klienta sú odoslané už len výsledky

2.2.2 Document Object Model (DOM)

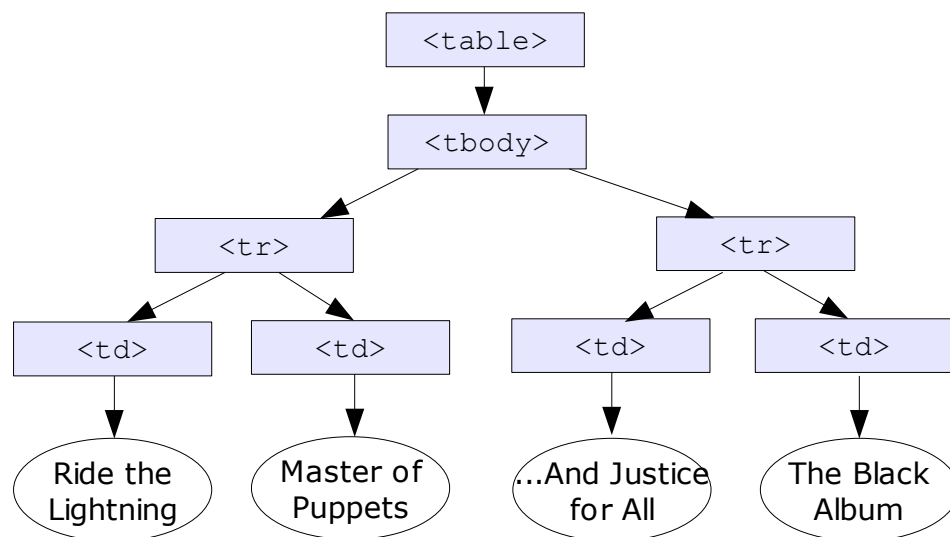
Document Object Model (ďalej len DOM) je objektový model dokumentu. Jedná sa o platformovo a jazykovo nezávislý objektový model pre reprezentáciu HTML, XML a iných príbuzných formátov z dokumentov.

Webový prehliadač nemusí používať DOM na vykreslenie HTML dokumentu, ale prítomnosť DOMu je vyžadovaná pri použití JavaScriptu, teda ak chceme extrahovať určité informácie z HTML dokumentu alebo manipulovať s ním. DOM definuje programové rozhrania na manipuláciu s dokumentom, posúvanie sa po jednotlivých elementoch ako aj štruktúru, názvoslovie a jednotlivé prvkové typy dokumentu.

```
<table id="favourite_albums_table">
  <tbody>
    <tr>
      <td>Ride the Lightning</td>
      <td>Master of Puppets</td>
    </tr>
    <tr>
      <td>...And Justice for All</td>
      <td>The Black Album</td>
    </tr>
  </tbody>
</table>
```

Ukážka kódu 2.1: Príklad HTML kódu popisujúceho tabuľku

Tento kus zdrojového HTML kódu by bol pomocou DOM transformovaný do nasledujúcej štruktúry:



Obr. 2.2: Stromová štruktúra kódu 2.1 v DOM

Pomocou JavaScriptu by sme text v tabuľke v jednej bunke zmenili nasledovným spôsobom:


```
//získam element tabuľky z elementu document použitím funkcie
//getElementById
var tableElement = document.getElementById("favourite_albums_table");

//získam prvý tr element cez odkazy na prvého potomka
var firstTdElement = tableElement.firstChild.firstChild;

//odkaz na druhý td element, zoznam potomkov je číslovaný od 0
var secondTdElement = firstTdElement.childNodes[1];

//odkaz na obsah textového elementu vnútri elementu td
var tableCell = secondTdElement.firstChild;

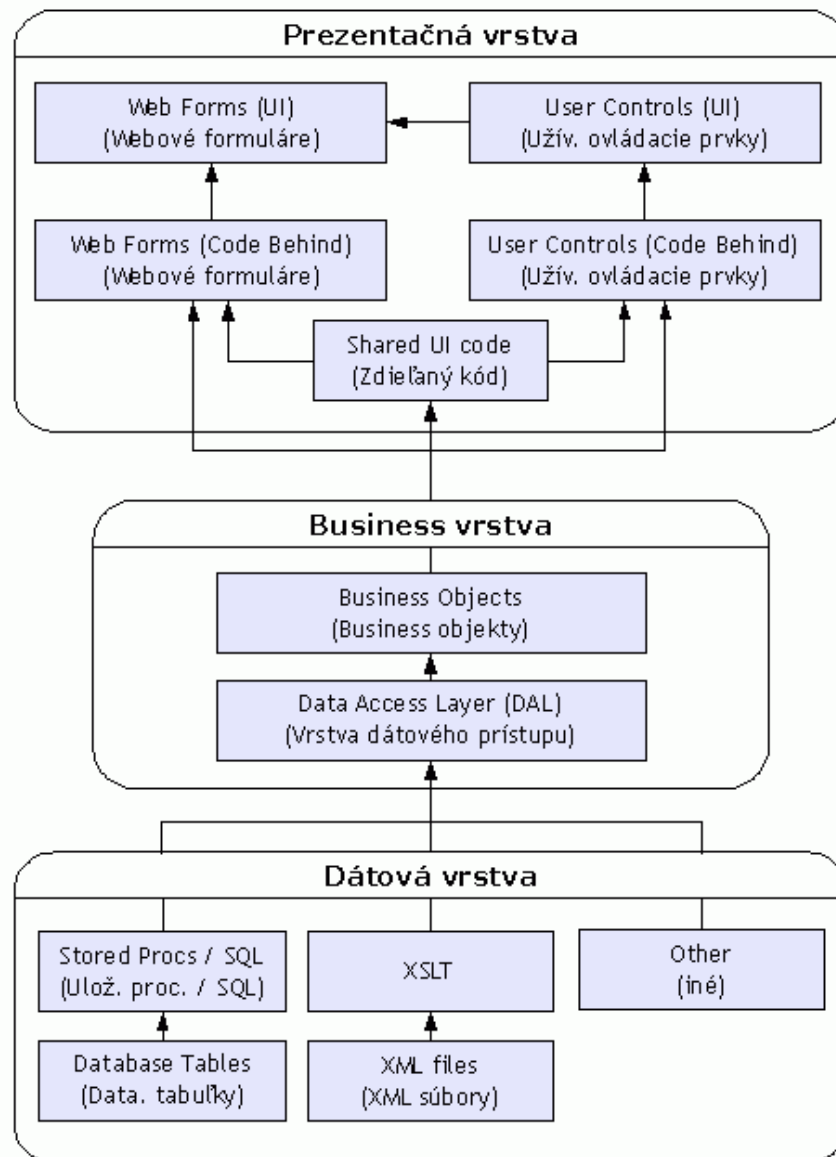
//zmeníme hodnotu bunky z "Master of Puppets" na "Garage Inc."
tableCell.data = "Garage Inc.";
```

Ukážka kódu 2.3: Zmena textu bunky v tabuľke pomocou JavaScriptu

DOM definuje jednotlivé typy prvkov, rozhrania funkcií a názvoslovie a menný priestor. Nie je to však implementácia týchto funkcií alebo prvkov, iba odporúčanie pre prehliadače, ktoré chcú nasledovať normu. Špecifikácia DOMu je vyvíjaná spoločnosťou W3C a poslednou odporúčanou špecifikáciou je dokument DOM Level 3 z apríla roku 2004 [7].

2.3 ASP.NET

ASP.NET je webový aplikačný rámec (framework) vyvíjaný a propagovaný spoločnosťou Microsoft, ktorý je určený na vývoj dynamických webových stránok, webových aplikácií a webových služieb. Prvá verzia bola uvedená na trh v roku 2002 (verzia 1.0) ako súčasť aplikačného rámca .NET Framework⁷ a ako nástupca technológie Active Server Pages (ASP). Rámec .NET je postavený na vrstve zvanej „Common Language Runtime“, ktorá umožňuje programátorom využívať akéhokoľvek jazyka z rodiny .NET jazyky.



Obr. 2.4: Architektúra ASP.NET Frameworku [3]

⁷ .NET Framework je aplikačný rámec, ktorý integruje celý rad Microsoft technológií. Od prezentačnej vrstvy vo webových stránkach a klientských aplikáciách na systémoch Windows až po serverové služby, bezpečnostné služby a databázové služby.

2.3.1 Základné charakteristiky

Webové stránky ASP.NET , známe tiež ako webové formuláre (web forms) sú základným stavebným kameňom pre výstavbu aplikácií. Tieto stránky sú uložené v súboroch s príponou ASPX, a nadväzujú tak na pôvodnú technológiu, kde stránky majú príponu ASP. V týchto súboroch je uložená prezentačná väčšinou vo forme HTML alebo XHTML kódu spolu so špeciálnymi elementmi, tzv. Webovými ovládacími prvkami (Web Controls) a Užívateľskými ovládacími prvkami (User Controls), ktoré vytvárajú dynamické časti stránok. Je tu možnosť vloženia dynamického kódu priamo do stránky medzi značky označujúce dynamický blok <% nejaký dynamický kód %> tak ako tomu je u iných webových technológií (PHP, JSP alebo ASP). Tento prístup sa však neodporúča kvôli zneprehľadneniu kódu ako tendencii k vytváraniu zlých logických štruktúr.

```
<%@ Page Language="C#" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">

    protected void Page_Load(object sender, EventArgs e)
    {
        testLabel.Text = "Dobré ráno!";
    }

</script>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Sample page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Label id="testLabel" runat="server" />
    </div>
    </form>
</body>
</html>
```

Ukážka kódu 2.5: Príklad ASPX stránky v s dynamickým skriptom v jazyku C# (Example.aspx):

Takýto prístup tvorby aplikácií nie je však ideálny ani odporúčaný. Microsoft vyvinul koncept zvaný Code-Behind (Kód na pozadí), ktorý rozdeľuje aplikáciu na funkčnú a prezentačnú vrstvu, kde prezentačná vrstva sa sústreďuje do súborov s príponou .aspx a programová vrstva do asociovaných súborov s príponou .cs. Predchádzajúci súbor Example.aspx by sa tak rozdelil na súbory Example.aspx a Example.aspx.cs. Vo vývojových prostrediach ako Microsoft Visual Studio a iných sa toto deje automaticky.

Rozdelenie súboru Example.aspx na dva súbory v rámci konceptu Code-Behind:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Example.aspx.cs" Inherits="WebSite.Example" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
  <title>Example.aspx</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <asp:Label ID="timeLabel" runat="server" />
    </div>
  </form>
</body>
</html>
```

Ukážka kódu 2.6: Vizualná časť ASPX stránky (súbor Example.aspx)

```
using System;
using System.Web.UI;

namespace WebSite
{
    public partial class Example : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            timeLabel.Text = "Aktuálny čas:" +
                DateTime.Now.ToLongTimeString();
        }
    }
}
```

Ukážka kódu 2.7: Code-Behind časť ASPX stránky (súbor Example.aspx.cs)

Okrem konceptu Code-Behind sa technológia ASP.NET vyznačuje radom ďalších možností ako je manažment stavu aplikácie (Application state) , manažment užívateľského stavu (Session state), manažment pohľadu (View state). Ponúka rôzne spôsoby autentifikácie a autorizácie (LDAP, Active Directory) a napojenia na rôzne databázové systémy (MS SQL, MySQL, PostgreSQL, DB2, Sybase).

2.3.2 MS SQL

Microsoft SQL Server je hlavná relačná databáza spoločnosti Microsoft. Hlavným dotazovacím jazykom je Transact-SQL, čo je implementácia ANSI/ISO štandardu jazyka SQL (Structured Query Language).

Prvá verzia produktu MS SQL Server (verzia 4.21) pre systémy Windows vyšla v roku 1994, ale počiatky tejto databázovej aplikácie siahajú do roku 1989, keď sa firmy Microsoft, Sybase a Ashton-Tate spojili pri vývoji aby prerazili na databázový trh veľkých spoločností. Postupom času sa pozícia tohto produktu na trhu ustálila a posledná verzia (Microsoft SQL Server 2008) vyšla vo februári 2008. Okrem komerčných verzií pre veľké spoločnosti existuje aj verzia tohto serveru, ktorá je zdarma (SQL Server Express 2005) a prístupná pre voľné využitie. Táto verzia bola použitá pri tvorbe tejto práce.

MS SQL bola zvolená pre jednoduchú integráciu s technológiou ASP.NET, ale pre potreby tejto práce by bolo možné použiť aj iné typy databáz (MySQL, PostgreSQL).

3 Analýza a návrh riešenia

Obsahom tejto kapitoly je analýza problému a možnostiach riešenia. Je tu rozobratá problematika vytvárania webových formulárov, popis závislostí ovládacích prvkov, dynamické spracovanie formulárov. Koniec kapitoly uzatvára popis iných jazykov pre vytváranie webových formulárov.

3.1 Popis problému

Cieľom práce je vytvoriť užívateľsky príjemný a použiteľný vizuálny editor, ktorý by umožňoval užívateľovi výstavbu webových dotazníkov bez znalosti HTML kódu. Primárny zámer je umožniť užívateľovi vytvárať jednoduché dotazníky na rôzne ankety, prieskumy, zber rôznych komentárov, postrehov alebo aj súborov. Užívateľské prostredie by malo byť intuitívne a priateľské s možnosťami vytvárania nielen ovládacích prvkov formulárov ale aj iných elementov ako sú paragrafy, zoznamy, tabuľky a pod. Presúvanie textu, znema hrúbky, skosenia alebo typu textu by mali byť samozrejmosťou.

Ďalšia úloha pozostáva z vytvorenia serverovej časti a teda sevreverého skriptu, ktorý užívateľom vytvorený webový formulár spracuje, vyextrahuje ovládacie prvky, na základe nich vytvorí vlastný formulár ako aj vhodnú databázovú štruktúru (tabuľku), do ktorej sa výsledky z formulára budú ukladať.

Posledným krokom je vytvorenie vhodného užívateľského prostredia pre vytváranie nových formulárov, používanie už vytvorených formulárov, prezeranie dát obsiahnutých v databáze ako aj mazanie formulárov spolu s ich asociovanými dátami.

3.2 Teoretický úvod

Prvky dotazníkov popísané v kapitole 2.1.3 môžu medzi sebou vytvárať určité závislosti. Nasleduje formálny popis prvkov a závislostí medzi nimi na základe matematického aparátu.

Webový formulár (dotazník) pozostáva z prvkov, ktoré sú určitým spôsobom zapísané. Svojim zápisom vytvárajú určitú štruktúru. Táto štruktúra je súčasťou HTML dokumentu, stránky. HTML dokument vytvára strom prvkov, kde koreňovým uzlom je vždy element `<html>`. Vo formulári je koreňovým prvkom element `<form>` a tiež vytvára koreň stromu. Pre formálne definície je treba siahnuť po teórii grafov:

Orientovaný graf je trojica $G=(V, E, \varepsilon)$ tvorená neprázdnu konečnou množinou V , ktorej prvky nazývame *vrcholy*, konečnou množinou E , ktorej prvky nazývame *orientovanými hranami*,

a zobrazením $\varepsilon: E \rightarrow V^2$, ktoré nazývame *vzťahom incidencie*. Toto zobrazenie priraduje každej hrane $e \in E$ usporiadanú dvojicu (x, y) . Prvý z nich, x , nazývame *počiatočným vrcholom hrany* a značíme ho $Pv(e)$. Druhý nazývame *koncovým vrcholom hrany* a značíme ho $Kv(e)$.

O hrane e hovoríme, že *vedie z vrcholu x do vrcholu y* a tiež, že *spojuje vrcholy x a y* . O vrchoch x, y potom hovoríme, že sú *incidentné* (alebo že *incidujú*) s hranou e a tiež naopak hrane e je *incidentná* s vrcholmi x, y . Oba vrcholy x, y tiež súhrnne nazývame *krajnými vrcholmi hrany e* . [9]

Graf G' je *podgrafom* grafu G , ak vznikne z grafu G vynechaním nejakých (alebo žiadnych) vrcholov alebo hrán. Podgraf musí byť grafom, to znamená, že spolu s každou hranou, ktorá je v podgrafe, tam musia byť aj oba jej krajné vrcholy.

Graf G' nazývame *podgrafom indukovaným množinou vrcholov* $S \subseteq V(G)$, ak podgraf G' má množinu vrcholov S a obsahuje všetky hrany grafu G , ktorých vrcholy ležia v množine S .

Koreňový strom je orientovaný graf, v ktorom existuje význačný vrchol r , tzv. *koreň*, a to taký, že do koreňa nevedie žiadna hrana, do každého iného vrcholu vedie presne jedna hrana a navyše sú všetky vrcholy z koreňa r dostupné. Koreňový strom býva niekedy nazývaný aj *vetvením*.

Ak vedie v koreňovom strome hrana z vrcholu x do vrcholu y , tak vrchol x nazývame *otcom vrcholu y* a vrchol y nazývame *synom vrcholu x* . Vrchol, ktorý nemá žiadneho syna nazývame *listom* [10] V HTML dokumentoch ako aj v špecifikácii DOM sa ďalej hovorí o vrchoch, ktoré majú rovnakého *otca* ako o *súrodencoch* (siblings), zavádzajú sa ďalšie pojmy ako *praotec*, *vnuk*, *predchodca* (ancestor), *potomok* (descendant).

Usporiadáný koreňový strom je koreňový strom, v ktorom je pre každý vrchol určené usporiadanie jeho synov. Usporiadáný koreňový strom kreslíme spravidla tak, aby synovia boli usporiadaný zľava doprava. [10]

Ak si uvažujeme akýkoľvek validný HTML dokument, dokážeme ho popísať usporiadaným koreňovým stromom. Potom časť dokumentu, ktorá reprezentuje formulár (obsah dokumentu uzavretý počiatočnou a koncovou značkou elementu `form`) môžeme popísať ako indukovaný podgraf tohto koreňového stromu (podstrom).

Vrcholy sú definované názvami elementov, atribúty vrcholov tvoria množinu A , ktorá obsahuje usporiadané dvojice (x, y) , také, že $x \in A$ a $y \in N$, kde A je množina názvov atribútov a N je množina hodnôt. Nasledujúci príklad ilustruje situáciu.

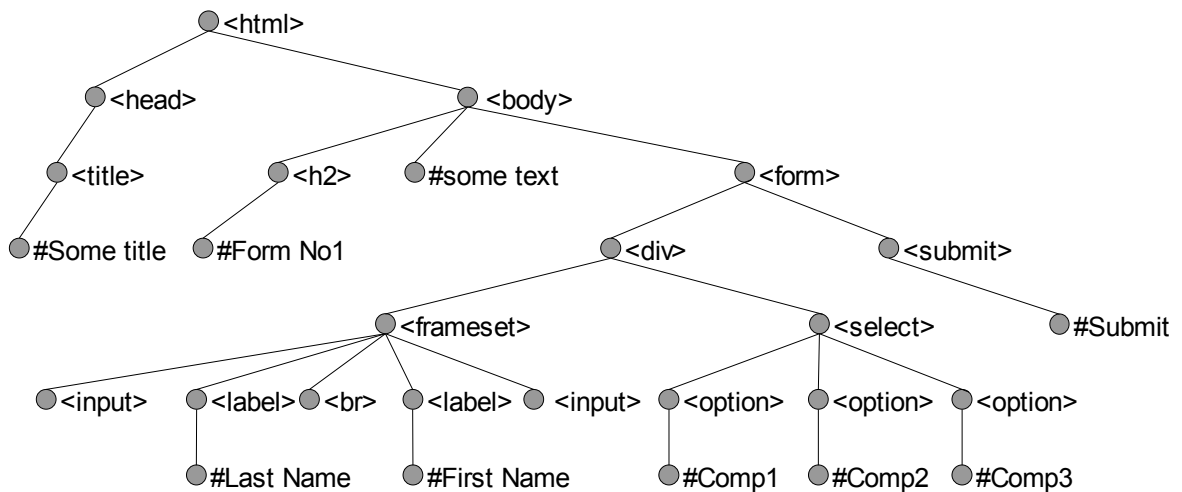
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head><title>Example</title></head>
  <body>
```

```

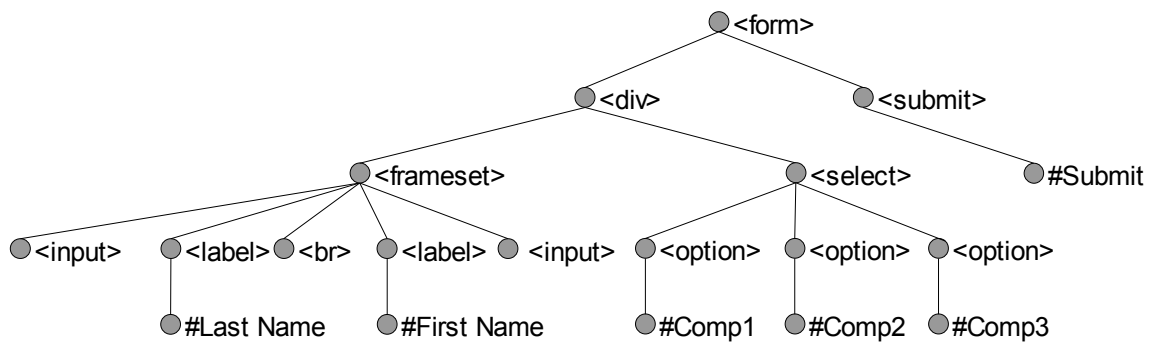
<h2>Form No1</h2>
some text
<form id="form1" name="form1">
  <div>
    <frameset>
      <label for="last_nameID">Last Name:</label>
      <input id="last_nameID" name="last_name" type="text"
size="20"><br>
      <label for="first_nameID">First Name:</label>
      <input id="first_nameID" name="first_name" size="20">
    </frameset>
    <select multiple size="3" name="components">
      <option value="c1">Comp1</option>
      <option value="c2">Comp2</option>
      <option value="c3">Comp3</option>
    </select>
  </div>
<button type="submit">Submit</button>
</form>
</body>
</html>

```

Ukážka kódu 3.1: Príklad HTML dokumentu s formulárom



Obr.3.2 Stromová reprezentácia html dokumentu



Obr. 3.3 Indukovaný podstrom s koreňom vo vrchole <form>

3.3 Závislosti prvkov

Pri textovej alebo inej editácii štruktúry HTML dokumentu dochádza aj k modifikácii koreňového stromu, ktorý tento dokument reprezentuje. *Vytvorením nového prvku* dochádza k vytváraniu nového vrcholu a k nemu vedúcej hrany od otca tohto prvku. *Mazaním prvku* z dokumentu dochádza k odstráneniu tohto vrcholu, celého indukovaného podstromu tohto vrcholu ako aj hrany vedúcej od otca tohto vrcholu. *Premiestnením* prvku rozumieme jeho zmazanie a vytvorenie na inom mieste v dokumente.

Prvky webového formulára v HTML dokumente môžu vytvárať rôzne závislosti medzi sebou. Jedna z typov je *rodinná závislosť*. Znamená to, že pre správne fungovanie formulára je dôležité, aby prvok *A* bol otcom prvku *B*, alebo aby prvok *C* bol potomok prvku *D*. Napríklad všetky ovládacie prvky, ktoré majú byť pri odoslaní dotazníku úspešné musia byť potomkami prvku `<form>`, prvok `<option>` musí byť synom prvku `<optgroup>` alebo prvku `<select>`, prvok `<optgroup>` musí byť synom prvku `<select>`, prvok `<textarea>` nesmie mať syna iný prvok okrem textového literálu atď.

Atribútová závislosť znamená, že hodnota atribútu nejakého prvku musí byť totožná s hodnotou atribútu iného prvku niekde v dokumente. Napríklad prvok `<label>` v prípade, že nie je otcom ani praotcom nijakého ovládacieho prvku môže vo svojom atribúte `for` uviesť identifikátor iného prvku, pre ktorý má slúžiť ako popisná značka. V dokumente musí teda existovať taký prvok, ktorého atribút `id` má túto hodnotu.

Funkčné závislosti popisujú funkčnosť, schopnosť prvku zisťovať alebo meniť hodnoty atribútov iného alebo iných prvkov. *Vypnutie* je taká funkčná závislosť, ktorá umožňuje prvku znepriístupniť iný ovládací prvok alebo všetky ovládacie prvky, ktoré sú potomkami vypínaného prvku. Naopak *zapnutie* je taká funkčná závislosť, ktorá umožňuje prvku sprístupniť ovládacie prvky, ktoré sú v podstrome vybraného prvku, ak boli v stave vypnuté.

Zneviditeľnenie je taká funkčná závislosť, ktorá umožňuje prvku zneviditeľniť iný prvok a celý všetkých jeho potomkov. Daný podstrom sa z dokumentu nevymaže, len nastaví atribúty jednotlivých prvkov tak, aby neboli vykresľované. *Zviditeľnenie* je teda taká funkčná závislosť, ktorá zviditeľní prvok a celý jeho podstrom, v prípade, že bol v stave neviditeľný.

Kontrolná závislosť je taká funkčná závislosť, ktorá určuje akú hodnotu má mať atribút iného prvku v momente odoslania formulára. Jedná sa hlavne o schopnosť kontrolovať obsah vyplnený užívateľom, tj. validácia potrebných ovládacích prvkov (vyplnenie povinných polí, kontrola zadaných hesiel apod.).

3.3.1 Validácia prvkov

Validácia hodnôt je dôležitou súčasťou návrhu užívateľsky príjemných dotazníkov. Ide o proces kontroly zadaných hodnôt, overovanie povinných polí, overovanie formátu alebo typu zadanej hodnoty (číslo, reťazec).

Vo webových stránkach sa používajú rôzne techniky pre implementáciu validácie, ale v zásade ich môžeme rozdeliť na dva hlavné typy. Klientská validácia, prebieha priamo v internetovom prehliadači, predtým ako sú dáta odoslané na server. Na klientskú validáciu sa používa JavaScript. V čase zobrazenia formuláru musí mať prehliadač všetky informácie na validáciu. Ak nejaká informácia chýba alebo je v zlej forme, prehliadač nedovolí užívateľovi odoslať dáta na server. Hlavná výhoda tohto prístupu je rýchlosť odozvy a efektívnosť, pretože dáta zbytočne neputujú po sieti. Naproti tomu serverová validácia prebieha na serveri, až po odoslaní formulára užívateľom. V prípade nedostatkov sa odosiela formulár späť užívateľovi na doplnenie. Tento prístup kladie väčšie nároky na server, ale v niektorých prípadoch je žiadaný (napr. nechceme aby užívateľ videl obmedzenia hodnôt, bezpečnosť aplikácie apod.) Ako zaujímavý prístup sa v posledných rokoch javí zmiešaná validácia, kde niektoré hodnoty sa validujú na strane klienta, iné na serveri, s využitím Ajaxu⁸.

V tejto práci je použitá validácia na strane klienta. Síce aplikačný rámec ASP.NET poskytuje validáciu na ovládacích prvkoch na strane serveru, ale táto validácia je určená len pre ovládacie prvky vytvorené v rámci ASP.NET (WebControls) a nie štandardné ovládacie prvky v HTML. WebControls totiž spájajú popisy (label) s vlastným ovládacím prvkom, čo znemožňuje ich použitie v užívateľsky navrhovaných dotazníkoch. Z tohto dôvodu nie je možné použiť ani validáciu poskytovanú v ASP.NET.

3.4 JavaScript WYSIWYG editor

Vizuálnych editorov pre editáciu textu alebo priamo návrh webových stránok existuje hneď niekoľko. Existujú komerčné ako aj riešenia s otvoreným zdrojovým kódom (open source), napríklad FCKeditor, TinyMCE, FreeRichTextEditor atď. V tejto práci sa samozrejme zaoberáme iba tými z druhej kategórie. Pri rozhodovaní pre výstavbu úplne nového editoru alebo rozšírenie už existujúceho som sa rozhodol rozšíriť už existujúci editor tak, aby spĺňal potreby mojej práce. Rozhodol som sa pre FCKeditor [15], ktorý je dostupný pod niekoľkými licenciami pre open source projekty ako aj pre komerčné využitie spoločností, ktoré ho chcú začleniť do svojho softvéru. Pre FCKeditor som sa rozhodol, pre jeho rýchlosť, výkonnosť, jednoduchú integráciu do viacerých

⁸ Ajax (Asynchronous JavaScript and Xml) je technika implementácie webových aplikácií založená na prenose menšieho množstva dát po sieti, s cieľom zvýšiť odozvu týchto aplikácií.

webových technológií (ASP.NET, PHP, JSP, Perl atď) a jednoduchý spôsob ovládania. Začiatkové tri písmená v názve tohto editoru značia Frederico Caldeira Knabben, teda meno pôvodcu a vedúceho vývojára tohto softvéru.

Tento editor poskytuje niektoré základné ovládacie prvky, s ktorými sa dá pracovať vo WYSIWYG móde. Medzi ne patria *zaškrtávacie tlačítko*, *rádiové tlačítko*, *textová pole*, *vstupné pole*, *element form*, *výberové menu*, *skryté pole* a *tlačítko* (Verzia FCKeditoru 2.6 z aprílu 2008). Ostatné ovládacie prvky (`label`, `fieldset`, `legend`, `file upload` a iné) tento vizuálny editor neobsahuje a je potrebné ho o tieto ovládacie prvky rozšíriť. Takisto nie sú implementované závislosti medzi ovládacími prvkami, takže je potrebné editor rozšíriť o túto funkcionality.

3.5 Spracovanie formulára

Spracovanie formulára je dôležitou súčasťou celej aplikácie. Vizuálny editor reprezentuje vizuálne len obyčajný HTML kód ale nijak ďalej tento kód nespracováva. Pri odoslaní navrhnutého formulára editorom je na server odoslaný HTML kód. Tento kód je potrebné zanalyzovať (rozparsovať) a na jednotlivé ovládacie prvky adekvátne reagovať.

Spracovávané ovládacie prvky sú len tie, ktoré sa vyskytujú vo vnútri prvku `form`. Ak sa v HTML kóde nevyskytuje tento prvok, nevytvára sa dynamický formulár ani pridružená databázová štruktúra. Ak sa ale v odoslanom dokumente vyskytujú ovládacie prvky, je potrebné ich jednoznačne identifikovať (atribút `id` s unikátnym identifikátorom v rámci dokumentu), rozpoznať typ, poprípade začleniť do množiny asociovaných ovládacích prvkov (v prípade skupiny rádiových tlačítok alebo zaškrtávacích tlačítok).

Pre rozpoznávanie jednotlivých ovládacích prvkov sa použijú regulárne výrazy (regular expressions). Pri rozpoznaní sa zisťuje typ ovládacieho prvku, jeho atribúty, validujú sa atribúty `name` a `id` v prípade chýbajúcich atribútov sa tieto vygenerujú. Popis formulára s jeho HTML kódom, identifikovanými ovládacími prvkami, závislosťami a príslušnými SQL kódmi uloží do databázy. Táto časť databázy obsahuje popisné informácie k ďalším tabuľkám, a preto ju nazývame meta časťou.

Zaujímavá situácia nastáva pri návrhu viac ako jedného formulára na jednej stránke. V tomto prípade je potrebné jednoznačne rozlíšiť formuláre od seba identifikátorom (`id`) ako aj odosiacím tlačítkom, presnejšie hodnotou atribútu `name` v tomto prvku. Len na základe tejto hodnoty je totiž možné identifikovať, ktorý formulár bol odoslaný k spracovaniu.

3.6 Iné jazyky popisujúce webové dotazníky

Popri webových formulároch definovaných v špecifikácii HTML 4.01 existujú aj rozšírenia tejto špecifikácie ako aj ďalšie jazyky popisujúce formuláre. Zameriame sa na rozšírenie Web Forms 2.0 a XForms.

Web Forms 2.0 je plánovanou súčasťou špecifikácie HTML5, dokumentu, ktorý je vyvíjaný skupinou WHATWG⁹ a v januári roku 2008 nadobudol stav Working Draft. Špecifikácia rozširuje možnosti formulárov popísaných v špecifikácii HTML4. Web Forms 2.0 sú platné pre ako pre jazyky HTML tak aj pre XHTML. Poskytujú silne typované vstupné polia, nové atribúty pre definovanie obmedzení (constraints), opakovací model pre deklaratívne opakovanie častí formulára, odoslanie a inicializáciu formulárov v XML a iné. Tiež štandardizujú a kodifikujú súčasné postupy tvorby formulárov v oblastiach, ktoré neboli doteraz dokumentované a objasňujú vzťahy medzi formulárovými ovládacími prvkami a CSS¹⁰.

XForms je jazyk založený na jazyku XML. Konzorcium W3C navrhlo špecifikáciu tohto jazyka a schválilo verziu 1.0 ako W3C Recommendation v roku 2003. Jazyk XForms nebol navrhnutý pre samostatný výskyt, ale vždy sa vyskytuje ako súčasť dokumentu iného jazyka XML, HTML, XHTML.

XForms bude súčasťou špecifikácie XHTML 2.0, teda dokumentu, na ktorom konzorcium W3C ešte stále pracuje (máj 2008). XForms sa líšia vo svojej podstate od súčasných webových formulárov, existuje teda určitá miera úsilia, ktorú vývojári budú musieť vynaložiť pri osvojení si tejto technológie, avšak pre vývoj kvalitných, rozsiahlych a komplexných formulárov poskytuje viac možností a rýchlejši ako súčasné webové dotazníky. [11] XForms 1.1 zavádza niekoľko zlepšení oproti pôvodnej verzii 1.0 a v súčasnosti má tento dokument štatút W3C Candidate Recommendation.

Formulár v XForms je organizovaný do *inšancií* dátových schém. Takýto formulár umožňuje spracovávať dáta pomocou troch mechanizmov. Deklaratívny model (model) pozostáva z rovníc pre kalkuláciu dát a obmedzení na nich, dátových typoch a parametrov pre odoslanie dát. Zobrazovacia vrstva (view) popisuje spôsob zobrazenia ovládacích prvkov v užívateľskom rozhraní. Kontrolór (controller) vykonáva manipuláciu dát, kontroluje vzťahy medzi modelom a zobrazovacou vrstvou a odosielením dát.

9 WHATWG - Web Hypertext Application Technology Working Group, skupina založená vývojármi zo spoločností Apple, Mozilla Foundation a Opera Software v roku 2004 po W3C workshope. Apple, Mozilla a Opera boli nespokojní so smerovaním XHTML 2.0 špecifikácie a tak vytvorili vlastnú pracovnú skupinu s úmyslom vytvárať špecifikácie bližšie potrebám súčasných webových tvorcov.

10 CSS - Cascading Style Sheet - jazyk pre popis štýlových definícií je značkový jazyk pre vizuálny popis elementov. Používa sa hlavne v kombinácii s jazykmi HTML a XHTML ale môže byť použitý spolu s jazykom XML

Hlavné výhody XForms spočívajú v robustnosti a rozdelenia formulárov na niekoľko úrovní. Vďaka svojmu popisu v XML je táto technológia použiteľná aj mimo svet WWW a môže byť nájst' využitie v mobilných zariadeniach (PDA, mobilné telefóny atď'). [11]

V súčasnosti (máj 2008) neexistuje webový prehliadač, ktorý by podporoval XForms. Existujú však niektoré rozšírenia a samostatné klientské implementácie. Firefox XForms Extension je rozšírenie pre webový prehliadač Firefox a Mozilla. Implementácia nepokrýva celú špecifikáciu ale pokrýva väčšinu z nej. FormFaces je čisto JavaScriptový procesor XForms schopný pretransformovať XForms v kombinácii s HTML na štandardné HTML formuláre. FormFaces je kompatibilný s väčšinou súčasných hlavných internetových prehliadačov ako Internet Explorer, Mozilla, Firefox, Opera, Konqueror, Safari a NetFront. Medzi hlavné klientské implementácie tejto technológie patrí IBM Lotus Forms, OpenOffice, PicoForms a DataMovil. V blízkej budúcnosti sa implementácia do hlavných prehliadačov kvôli rozsiahlosti a komplexnosti špecifikácie nepredpokladá, avšak kvôli spomenutým výhodám je XForms technológiou budúcnosti.

```
<xforms>
  <model>
    <instance>
      <person>
        <fname/>
        <lname/>
      </person>
    </instance>
    <submission id="form1" action="submit.asp" method="get"/>
  </model>
  <input ref="fname">
    <label>First Name</label>
  </input>
  <input ref="lname">
    <label>Last Name</label>
  </input>
  <submit submission="form1">
    <label>Submit</label>
  </submit>
</xforms>
```

Ukážka kódu 3.4: Formulár definovaný v XForms [12]

First Name	<input type="text"/>
Last Name	<input type="text"/>
<input type="submit" value="Submit"/>	

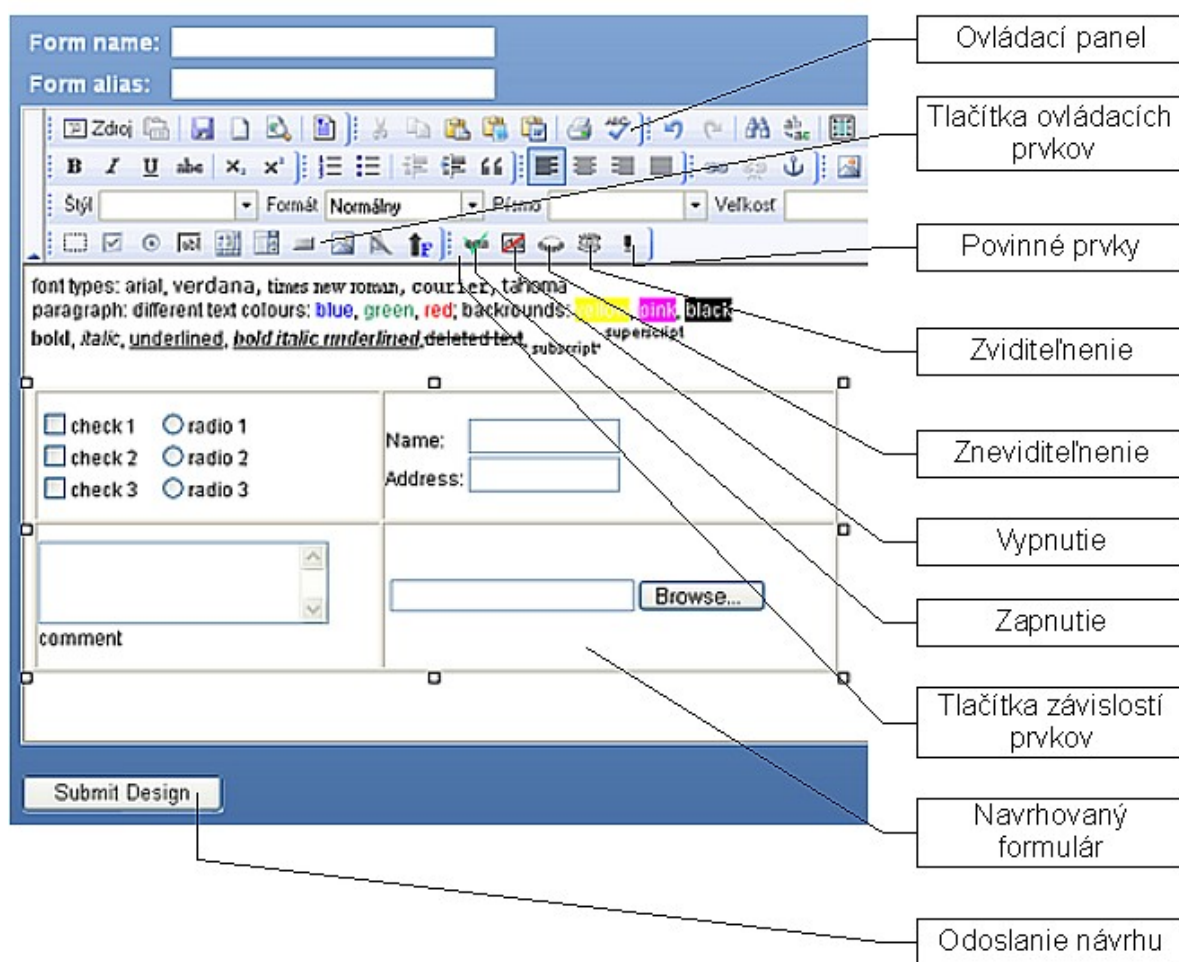
Obr 3.5: Zobrazenie XForms formuláru [12]

4 Implementácia

Táto kapitola popisuje implementáciu jednotlivých častí aplikácie. Popíšem funkčnosť a rozšírenia vizuálneho editoru, možnosti vytvárania formulárov vo vizuálnom móde ako aj nasledovné spracovanie dotazníku a prepojenie na databázu. Naznačím problémy, na ktoré som narazil počas implementácie a popíšem ich riešenie.

4.1 Vizuálny editor

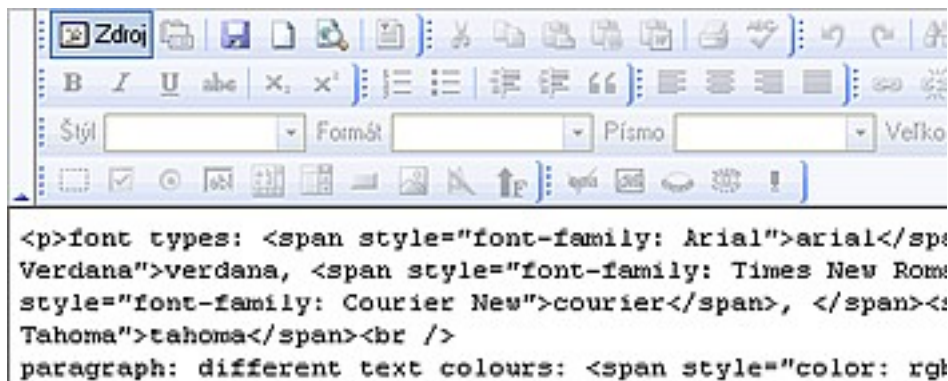
Vizuálny editor je základnou časťou aplikácie pre navrhovanie webových dotazníkov. Je rozšírením FCKeditoru s doplnenou funkčnosťou pre ovládacie prvky ako aj závislosti týchto prvkov.



Obr. 4.1: Vizuálny editor pri návrhu komplexného formuláru

Pri návrhu formulára je dôležitý element `<form>`, ktorý je povinný pre vytvorenie funkčného formulára a spracovávané sú len tie ovládacie prvky, ktoré sa nachádzajú vo vnútri tohto elementu.

Formulár môže obsahovať rôzne ovládacie prvky, ale pre jeho funkčnosť je dôležitý práve jeden prvok typu *submit*, tento prvok totiž odosiela užívateľom zadané informácie na server. Bez tohto prvku je formulár nefunkčný. Vizualný návrh formuláru je možný vkladaním jednotlivých ovládacích prvkov do formuláru, slúžia k tomu tlačítka ovládacieho panelu. Zároveň je možné editovať obsah, rozmiestnenie a formátovanie prvkov. HTML kód je možné editovať aj v zdrojovom pohľade.



Obr. 4.2: Upravovanie zdrojového HTML kódu

Do formulára je vizuálne možné vkladať tieto ovládacie prvky:

- zaškrtávacie políčko (`checkbox`),
- rádiové tlačítko (`radio button`),
- textové pole (`input text`),
- textové viacriadkové pole (`text area`),
- výberové menu (`select`),
- značka (`label`),
- tlačítko (`button`),
- skryté pole (`hidden field`),
- výber súboru (`file select`).

Pri vkladaní ovládacieho prvku sa zobrazí modálne okno so vstupmi, potrebnými k správne vygenerovaniu ovládacieho prvku. Počet a typ vstupov sa líši v závislosti na type ovládacieho prvku. Spoločný vstup pre každý ovládací prvok je názov prvku (`name`). Je potrebné uvedomiť si vplyv tohto prvku na správanie sa celého formuláru. Rôzne ovládacie prvky budú mať rôzne názvy, ale ak chceme zlúčiť rádiové tlačítka do jednej skupiny, zvolíme pre každé takéto tlačítko rovnaký názov. To isté platí pre zaškrtávacie tlačítka. Je vhodné dávať ovládacím prvkom vhodné názvy, pretože tieto názvy budú názvami stĺpcov príslušnej tabuľky v databáze (viď kapitolu 4.2). Pri vkladaní prvku editor kontroluje či názov prvku nekoliduje s inými názvami prvkov, v prípade nutnosti upraví názov tak, aby bolo v rámci formulára unikátne. Jednoznačné identifikátory (atribút `id`) sú generované automaticky.

Nad prvkami zaškrťavacie políčko (`checkbox`), rádiové tlačítko (`radio button`) a tlačítko (`button`) je možné definovať závislosti. Aplikácia umožňuje definovať tieto závislosti:

- zapnutie
- vypnutie
- zviditeľnenie
- zneviditeľnenie

Pre celý formulár je možné určiť množinu povinných prvkov. Ak sa užívateľ pokúsi odoslať formulár bez vyplnenia týchto prvkov formulár sa neodošle a užívateľ bude upozornený na chýbajúce vstupné informácie.

4.1.1 Kompatibilita prehliadačov

Vizuálny editor bol testovaný na prehliadačoch Mozilla Firefox (verzia 2.0.0.14), Mozilla (verzia 1.7.7), Internet Explorer (verzia 6), Internet Explorer (verzia 7) a Opera (verzia 9.50 beta). FCKeditor, ktorý je použitý ako základ vizuálneho editoru sa síce snaží pokrývať čo najviac prehliadačov, napriek tomu je v niektorých funkčnosť lepšia ako v iných. Najlepšie obstáli prehliadače Internet Explorer vo oboch testovaných verziách. Všetka funkčnosť poskytovaná FCKeditorom fungovala v týchto prehliadačoch bezchybne. Ostatné prehliadače, Firefox, Mozilla i Opera prejavili zníženú funkčnosť pri manipulácii s už vloženými ovládacími prvkami. Nefungujú kontextové menu nad prvkami formuláru ako aj presúvanie prvkov vo WYSIWYG móde. V prehliadačoch Mozilla a Firefox nie je možné vo vizuálnom móde označiť tlačítko, takže nad týmto prvkom nie je možné v týchto prehliadačoch definovať ani závislosti. Spomenuté nedostatky prehliadačov však vážnym spôsobom neznižujú použiteľnosť vizuálneho editoru na výstavbu užívateľsky príjemných formulárov.

4.2 HTML parser

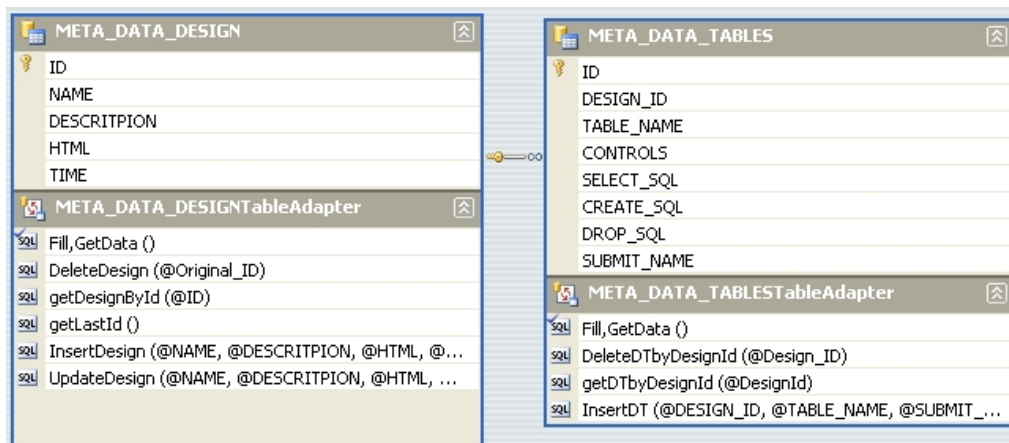
HTML parser (analyzátor) je kľúčovou časťou aplikácie. Napísaný je v jazyku C#. Pri odoslaní návrhu sa HTML kód dostane na server a tam je parserom zanalyzovaný a uložený do databázy. Identifikované sú formuláre, ovládacie prvky a ich atribúty, a tieto sú zapísané do meta tabuliek databázy. Na parsovanie a identifikáciu jednotlivých významných častí kódu sú použité regulárne výrazy. Validujú sa názvy ovládacích prvkov, vytvárajú sa unikátne identifikátory prvkov, vytvoria sa SQL skripty na vytvorenie, zmazanie tabuľky a na výber všetkých dát z tabuľky. Uchováva sa zvyšná časť HTML. Parser je pozostáva z niekoľko tried: `ControlParser`, ktorý spracováva ovládací prvok, `FormParser`, ktorý spracováva formulár a `DesignParser`, ktorý spracováva celý návrh.

Pomocná trieda `WebCapt`, je pôvodne prevzatá [13], bola však upravená a rozšírená pre použitie v tomto parseri.

Trieda `ControlParser` spracováva jeden formulárový dátový prvok¹¹. Upravuje HTML kód prvku ako aj vytvára XML kód, ktorý sa neskôr zapíše do tabuľky (stĺpec `CONTROLS`). Trieda `DesignParser` analyzuje celý odoslaný dokument, preposiela jednotlivé prvky triede na `FormParser` spracovanie ako aj zapúzdruje a zjednodušuje manipuláciu s už vytvoreným formulárom. Trieda obsahuje dva konštruktory a niekoľko verejných metód. Za povšimnutie stojí metóda `submitForm(HttpContext context)`, ktorá prevezme od webovej stránky kontext a zanalyzuje odoslaný formulár na základe prvkov uložených v meta tabuľkách, vytvorí SQL skript pre vkladanie záznamu (`INSERT`) a tento skript odošle na databázový server.

4.3 Databáza MS SQL

Databáza MS SQL je použitá na uchovávanie dát z odoslaných pomocou vytvorených dotazníkov. Dynamicky sa v nej vytvárajú alebo mažu tabuľky podľa operácií, ktoré sú vykonané na webovom rozhraní. *Meta* informácie o dátach a tabuľkách sú uložené v dvoch tabuľkách `META_DATA_DESIGN` a `META_DATA_TABLES` s kardinalitou 1 k N. Na prístup a manipuláciu s týmito tabuľkami je využité dátové rozhranie rámca ASP.NET.



Ukážka kódu 4.3: Vrstva dátového rozhrania nad meta tabuľkami aplikácie

Tabuľka `META_DATA_DESIGN` je primárna, obsahuje uložený HTML kód a popisné položky návrhu (`name`, `description`). Tabuľka `META_DATA_TABLES` je sekundárna, obsahuje cudzí kľúč z primárnej tabuľky a dátový popis formuláru, vrátane názvu dynamicky vytvorenej tabuľky, ovládacích prvkov, SQL skriptov atď. Táto štruktúra umožňuje návrh jedného alebo aj viac

¹¹ Dátový formulárový prvok je taký, ktorý nesie hodnotu a táto hodnota sa v konečnej fáze ukladá do databázy. Prvok typu tlačítko (`button`) nie je dátovým prvkom.

formulárov na jednej stránke. Dátové tabuľky dynamicky vytvárané v rámci aplikácie sú manipulované SQL skriptami (vytvor tabuľku, vlož záznam, znič tabuľku). Tieto príkazy sú vykonávané štandardným SQL pripojením na databázu

Server MS SQL využíva dotazovací jazyk typu Transact-SQL, ktorý je rozšírením štandardu SQL. Transact-SQL je rozšírený o vylepšenia ako kontrola toku, lokálne premenné, rôzne matematické funkcie a funkcie na spracovávanie textu, atď. [14]

Práve pre tieto odlišnosti nie je možné bez zásahu do kódu bezproblémovo prepojiť aplikáciu s databázou iného typu (MySQL, PostgreSQL a iné).

4.4 Konfigurácia a integrácia

Aplikácia je navrhnutá a implementovaná ako štandardná ASP.NET aplikácia. Pozostáva z dvoch projektov `FormDesigner` a `FCKeditorV2`. `FormDesigner` je hlavná časť, ktorá zahŕňa celú aplikačnú logiku, `FCKeditorV2` je pomocný projekt, slúži na integráciu `FCKeditoru` do ktoréhokoľvek `.aspx` súboru.

Projekt `FormDesigner` obsahuje celý projekt vrátane `WebForms`, `JavaScriptov`, parseru a konfiguračných súborov. Medzi význačné adresáre projektu patrí adresár `classes`, ktorý obsahuje triedy HTML parsera. Adresár `forms` obsahuje súbory webovej prezentačnej vrstvy. Jednotlivé aplikačné prezentácie sú prepojené jednou `MasterPage.Master`, ktorá ich združuje a zároveň definuje vzhľad celej aplikácie. V adresári `scripts` sú uložené súbory JavaScriptu vrátane rozšíreného `FCKeditoru`. Adresár `uploads` uchováva všetky súbory odoslané užívateľmi na server pomocou vygenerovaných dotazníkov a pre správne fungovanie je potrebné nastaviť na serveri práva na tomto adresári pre zápis. Tieto súbory sú vždy uložené pod adresárom s menom asociovaného dotazníku (názov tabuľky v databáze).

Pre úspešnú konfiguráciu a integráciu projektu je dôležitá konfigurácia databázového pripojenia a nastavenie vizuálneho editoru. V koreňovom adresári projektu `FormDesigner` sa nachádza súbor `Web.config`, ktorý obsahuje okrem iného nastavenie ciest a reťazcov na pripojenie do databáze. Pre správne fungovanie aplikácie je dôležité nastaviť v sekcii `<appSettings>` hodnoty kľúčom `FCKeditor:BasePath` (relatívne umiestnenie `FCKeditoru`) a `FormDesigner:uploadsDir` (relatívna cesta k užívateľmi odoslanými súbormi). V sekcii `<connectionStrings>` je dôležité nastaviť správnu hodnotu pre pripojenie do databázy pre kľúč `FormDesignerConnString`. Pre inicializáciu databázy je potrebné vytvoriť dve meta tabuľky spustením SQL skriptu `init.sql` v koreňovom adresári aplikácie.

Pre konfiguráciu vizuálneho editoru je dôležitý súbor `fckconfig.js` (umiestnený v `/scripts/fckeditor/editor/`), ktorý definuje vzhľad a funkcionality editoru. V tomto súbore je možné definovať rôzne formy ovládacích panelov pre vizuálny editor. Pre úplnú funkcionality editoru je potrebné zvoliť hodnotu `FormDesigner` z poľa `FCKConfig.ToolbarSets`. Rozšírenia editoru o závislosti prvkov sú nastavené takto: `FCKConfig.Plugins.Add("pluginName")`, kde `pluginName` je názov adresáru, kde je zásuvný modul umiestnený.

Celý projekt je pripravený na integráciu do väčšej aplikácie ASP.NET, stačí prekopírovať súbory a nastaviť potrebné atribúty v konfiguračných súboroch na správne hodnoty. Projekt nijakým spôsobom nerieši práva ani bezpečnosť aplikácie, pretože tieto otázky sú nad rámec tejto práce.

4.5 Zhrnutie

System pre užívateľský návrh webových formulárov pracuje na rozšírení vizuálneho JavaScript editoru vo webových stránkach. Vizuálny návrh generuje validný XHTML kód. Tento je spracovaný tak, že dotazníky navrhnuté editorom sú funkčné, umožňujú vkladanie všetkých základných ovládacích prvkov, podporujú dokonca aj jednoduchú validáciu. Implementované sú niektoré závislosti prvkov popísané v kapitole 3.3.

Aplikácia beží na adrese <http://xmikul29.aspone.cz> v plnej funkčnosti, takže vytváranie či mazanie formulárov, odosielanie dát alebo súborov, či akákoľvek ďalšia aktivita nie sú nijako obmedzené.

5 Záver

Úlohou práce bolo zoznámiť sa s možnosťami tvorby internetových dotazníkov, zanalyzovať ovládacie prvky týchto dotazníkov a identifikovať možné závislosti medzi nimi a vytvoriť webový vizuálny editor.

Pri analýze a návrhu riešenia sme sa detailne zamerali na viaceré dokumenty špecifikácií skupín W3C a WHATGW a do hĺbky sme sa zoznámili so smerovaním a súčasným vývojom jazyka HTML a príbuzných jazykov a formulárov definovaných pomocou týchto jazykov. Bolo treba pochopiť závislosti atribútov ako aj prvkov vo formulároch a navrhnúť vhodné riešenie na ich vizuálnu a databázovú reprezentáciu a manipuláciu.

Obtiažnou časťou implementácie bolo rozširovanie FCKeditoru, trvalo nejakú dobu, kým som sa v rozsiahlom kóde zorientoval a bol schopný editor ďalej rozširovať. HTML parser bol tiež náročný, kvôli množstvu stavov a atribútov, ktoré bolo treba kontrolovať. Veľkou pomocou v týchto úlohách mi bolo vývojové prostredie Visual Studio 2008, ktoré obsahuje vstavaný debugger ako pre C# tak aj pre JavaScript. Bez týchto nástrojov by bola práca ak nie nemožná, tak prinajmenšom veľmi zdĺhavá.

Vzniknutý projekt je možné ďalej rozširovať a pridávať novú funkcionálnosť. Ako zaujímavé rozšírenie sa javí generovanie zložitejšej databázovej štruktúry v prípade niekoľkých združených ovládacích prvkov, alebo pokročilá editácia už vytvorených formulárov. V súčasnej verzii je vytvorený formulár možné editovať iba čiastočne, ovládacie prvky, ich názvy a atribúty meniť nemožno. Upravovanie existujúcich ovládacích prvkov by bez ďalšej analýzy znamenala dátovú inkonzistenciu.

Práca na tomto projekte mi priniesla mnoho nových znalostí v návrhu a spracovávaní webových dotazníkov, v použití JavaScriptu pri tvorbe komplexného klientskeho editoru a v poslednej rade som sa zoznámil s ďalšími jazykmi pre popis webových formulárov. Programovanie webovej aplikácie v prostredí ASP.NET bolo príjemnou skúsenosťou a očakávam, že nadobudnuté znalosti a poznatky využijem pri ďalšej práci.

Literatúra

- [1] Wikipedia, the free encyclopedia. *Questionnaires* [online]. 2008 [cit. 2008-04-01]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Questionnaires>>.
- [2] W3C consortium. *HTML 4.01 Specification* [online]. 1997-1999 [cit. 2008-05-07]. Dostupný z WWW: <<http://www.w3.org/TR/html4/interact/forms.html>>.
- [3] Armstrong Damon. .NET Application Architecture: the Data Access Layer [online]. 2006 [cit. 2008-04-05]. Dostupný z WWW: <<http://www.simple-talk.com/dotnet/.net-framework/.net-application-architecture-the-data-access-layer/>>
- [4] Wikipedia, the free encyclopedia. *HTML* [online]. 2008 [cit. 2008-04-01]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Html>>
- [5] W3C consortium. *XHTML™ 1.0 The Extensible HyperText Markup Language* [online]. 2000 [cit. 2008-04-01]. Dostupný z WWW: <<http://www.w3.org/TR/xhtml1/>>.
- [6] Wikipedia, the free encyclopedia. *HTML* [online]. 2008 [cit. 2008-04-01]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Javascript>>
- [7] KŘIVKA, Zbyněk, KOLÁŘ, Dušan. *Principy programovacích jazyků a obiektovo orientovaného programování : IPP-2 Studijní opora* [online]. 2003-2006 [cit. 2008-04-15]. Dostupný z WWW: <<https://www.fit.vutbr.cz/study/courses/IPP/private/Opory/IPP-II.pdf>>.
- [8] W3C consortium. *Document Object Model (DOM) Level 3 Core Specification* [online]. 2004 [cit. 2008-04-07]. Dostupný z WWW: <<http://www.w3.org/TR/DOM-Level-3-Core/>>
- [9] DEMEL, Jiří. *Grafy a jejich aplikace*. Praha : Academia, 2002. 255 s.
- [10] KOSTELECKÝ, Ivo. *Generování formulářů definovaných v jazyce XForms*. 2006. FIT VUT v Brně. Diplomová práce.
- [11] Wikipedia, the free encyclopedia. *XForms* [online]. 2008 [cit. 2008-04-20]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Xforms>>.
- [12] W3Schools. *XForms Model* [online]. 1999-2008 [cit. 2008-04-20]. Dostupný z WWW: <http://www.w3schools.com/xforms/xforms_model.asp>.
- [13] COUTHURES, Alain. *Light HTML to XML Converter* [online]. 2008 [cit. 2008-04-20]. Dostupný z WWW: <<http://sourceforge.net/projects/light-html2xml/>>.
- [14] Wikipedia, the free encyclopedia. *Transact-SQL* [online]. 2008 [cit. 2008-04-25]. Dostupný z WWW: <<http://en.wikipedia.org/wiki/Transact-SQL>>.
- [15] KNABBEN, Frederico K. *FCKeditor, the text editor for Internet* [online]. 2003-2008 [cit. 2008-04-20]. Dostupný z WWW: <<http://www.fckeditor.net/>>.

Zoznam príloh

Príloha 1. CD