



FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY  
UNIVERZITA KOMENSKÉHO  
BRATISLAVA

---

Bakalárska práca

SYSTÉM NA EVIDENCIU A KATEGORIZÁCIU  
ŠTANDARDIZAČNÝCH MATERIÁLOV

Eva Porvazníková

vedúci bakalárskej práce: Doc. RNDr. Daniel Olejár, PhD.

Bratislava, 2008

Čestne prehlasujem, že túto prácu som vypracovala  
samostatne s použitím uvedenej literatúry a zdrojov.

.....

Chcela by som sa poďakovať vedúcemu mojej bakalárskej práce doc. RNDr. Danielovi Olejárovi PhD. a Martinovi Kralovičovi za rady a pripomienky pri písaní tejto práce, Mgr. Tiborovi Bajzíkovi za to, že stál pri mne, za veľkú trpezlivosť a podporu, a tiež všetkým ostatným, ktorí mi verili a povzbudzovali ma.

# Abstrakt

Cieľom bakalárskej práce bolo vytvoriť systém na evidenciu a kategorizáciu materiálov z oblasti informačnej bezpečnosti. Táto webová aplikácia je určená hlavne členom technickej komisie SÚTN ako podporný nástroj pre udržanie prehľadu o materiáloch v štandardizačnom procese a ich jednoduché vyhľadávanie podľa kritérií a kategórií.

kľúčové slová: webová aplikácia, ASP.NET, štandardy

# Obsah

<b>1. Úvod .....</b>	<b>7</b>
<b>2. Štandardizácia.....</b>	<b>9</b>
2.1 Podstata a význam štandardizácie.....	9
2.2 Štandardizačný proces.....	10
2.3 Štandardizačný materiál.....	12
<b>3. Analýza.....</b>	<b>14</b>
3.1 Používatelia a ich oprávnenia.....	14
3.2 Štandardizačný materiál.....	16
3.2.1 Kritériá a kategórie štandardizačných materiálov.....	17
<b>4. Návrh riešenia.....</b>	<b>19</b>
4.1 Výber technológií.....	19
4.2 Štandardy a kompatibilita s prehliadačmi.....	21
4.3 3 - vrstvomá architektúra .....	22
4.4 Návrh databázy.....	25
<b>5. Vývoj aplikácie.....</b>	<b>28</b>
5.1 ASP.NET.....	28
5.2 Architektúra.....	33
<b>6. Požiadavky na software a hardware.....</b>	<b>35</b>
<b>7. Možnosti rozšírenia.....</b>	<b>36</b>
<b>8. Záver.....</b>	<b>38</b>
<b>Zoznam použitej literatúry.....</b>	<b>39</b>
<b>Zoznam príloh.....</b>	<b>40</b>
<b>Príloha A – Špecifikácia bakalárskej práce.....</b>	<b>41</b>
A.1 Úvod.....	41
A.2 Aplikované štandardy a kompatibilita s prehliadačmi.....	42
A.3 Štandardizačný materiál.....	42
A.4 Kategórie štandardizačných materiálov.....	43
A.5 Vzťahy medzi štandardizačnými materiálmi.....	44
A.6 Podpora procesu vývoja ŠM.....	45
A.7 Možnosť vývoja aplikácie vo viacerých etapách.....	46

A.8 Bezpečnostné požiadavky na systém.....	47
A.8.1 Prístupové práva používateľov.....	47
A.8.2 Vytváranie auditných záznamov.....	50
A.8.3 Bezpečnostné požiadavky webovej aplikácie .....	51

# 1. Úvod

Vstupom do medzinárodných spoločenstiev Slovenská republika prijala nielen záväzok zosúladiť svoju legislatívu s legislatívou EÚ, ale aj zaviesť do STN európske technické normy. Na Slovensku za štandardizáciu podľa kompetenčného zákona zodpovedá Slovenský úrad pre metrológiu a skúšobníctvo, ktorý povinnosť zastupovať Slovenskú republiku v medzinárodných štandardizačných organizáciách a zabezpečovať plnenie z toho vyplývajúcich úloh delegoval na Slovenský ústav technickej normalizácie, SÚTN. SÚTN v zmysle tohto poverenia pôsobí ako národný normalizačný orgán a zastupuje SR v organizáciách ISO, ETSI, CEN, CENELEC. Medzinárodných noriem, ktoré by SR mala prevziať do STN je značný počet. Aby to SÚTN pri svojich obmedzených finančných a personálnych kapacitách stihol, zriadil rad technických komisií, zodpovedajúcich za jednotlivé oblasti štandardizácie. Zameranie komisií kopíruje do značnej miery technické výbory ISO.

Informatika a informačné technológie majú v štandardizácii aj na medzinárodnej úrovni zvláštne postavenie. Vývoj v oblasti informačných a komunikačných technológií napreduje takým tempom, že ISO spojila svoje sily s IEC a vytvorila JTC 1 (Joint technical committee 1) Information technology, ktorý sa člení na viacero podvýborov. Informačnú bezpečnosť v ňom zastrešuje podvýbor JTC 1/SC 27 IT Security techniques, ktorej v SÚTN zodpovedá Subkomisia SK 27 technickej komisie TK 37 (ďalej len SK 27). V pôsobnosti JTC 1/SC 27 je v súčasnosti síce len asi 80 štandardizačných materiálov, ale pri ich správe (udržiavanie, úpravy existujúcich a vývoj nových štandardov) vzniká ročne cca 700 ďalších dokumentov. Navyiac, SK 27 by mala sledovať aj činnosť ďalších štandardizačných organizácií, skúmať nimi vydávané štandardy a navrhovať ich prípadné zaradenie do STN.

Udržať si prehľad v takom množstve dokumentov, registrovať termíny pripomienkovania, konzultovať pripomienkované materiály s rozličnými odborníkmi, sledovať vzťahy medzi jednotlivými štandardami je časovo veľmi

náročná úloha. Preto vznikla idea vytvoriť systém, ktorý by pomáhal členom SK 27 pri spracovaní štandardizačných materiálov. Okrem možnosti vkladať a vyhľadávať materiály by systém mal podporovať aj priraďovanie materiálov do kategórií podľa určených kritérií, vyhľadávanie materiálov na základe týchto kritérií, vytváranie a vizualizáciu vzťahov medzi materiálmi, sledovanie a upozornenie používateľa na dátumy odovzdania pripomienok, koniec hlasovania a podobne.

Pôvodná základná špecifikácia sa v priebehu vývoja rozrastala a z jednoduchej aplikácie vznikol pomerne zložitý systém, ktorého vývoj, zdá sa, bude ešte nejaký čas pokračovať. Aby sme nemuseli čakať na definitívnu špecifikáciu a mohli členom SK 27 poskytnúť aspoň základný nástroj na uľahčenie činnosti, sme systém koncipovali modulárne a snažili sme sa, aby jednotlivé moduly boli funkčné a mohli byť postupne zavádzané do testovacej prevádzky. Predpokladáme, že testovacia prevádzka umožní odhaliť chyby v systéme, ako aj identifikovať ďalšiu funkcionálnosť, ktorú by systém mal poskytovať.

Predkladaná bakalárska práca nadväzuje na diplomovú prácu Martina Kraloviča [1], v ktorej je popísaný štandardizačný proces a definovaná základná funkčná špecifikácia systému.

Bakalárska práca pozostáva z úvodu, záveru a 6 kapitol. Druhá kapitola uvádza čitateľa do problematiky štandardizácie. Nezaoberáme sa obsahovou, ale procedurálnou stránkou štandardizácie a pridržame sa postupov používaných v ISO.

Jadro práce tvoria nasledujúce 4 kapitoly. V tretej analyzujeme požiadavky na systém, štvrtá kapitola obsahuje návrh riešenia, v piatej je stručne popísaná implementácia systému a v šiestej požiadavky na HW a SW.

V závere práce sa vraciame k problémom, na ktoré sme pri návrhu a vývoji systému narazili a stručne rozoberáme ďalšie možnosti rozširovania systému.

V prílohe práce je základná špecifikácia systému, ktorú ako súčasť zadania tejto bakalárskej práce vypracoval Martin Kralovič [Príloha A].



## 2. Štandardizácia

Pre priblíženie prostredia, v ktorom by mala pôsobiť naša aplikácia sa v tejto kapitole v základných rysoch oboznámime so štandardizáciou, presnejšie so správou technických noriem. Následne v krátkosti popíšeme štandardizačný proces ISO a jeho jednotlivé etapy. Na záver kapitoly sa budeme venovať štandardom, normám a ich vlastnostiam.

### 2.1 Podstata a význam štandardizácie

V tejto práci sa zaoberáme technickými štandardami. Technický štandard je obvykle formálny dokument, ktorý definuje jednotné výrobné alebo technické kritériá, metódy, procesy a postupy. Štandardy zaisťujú, aby výrobky a služby, ktoré sú vytvárané, resp. poskytované v súlade so štandardami, mali želané charakteristiky (kvalita, bezpečnosť, spoľahlivosť, efektívnosť, zameniteľnosť), nepoškodzovali životné prostredie a boli ekonomicky dostupné.

Najvýznamnejšia medzinárodná štandardizačná organizácia ISO (International Organization for Standardization) definovala význam vydávaných ISO štandardov nasledovne <sup>1</sup>

- make the development, manufacturing and supply of products and services more efficient, safer and cleaner
- facilitate trade between countries and make it fairer
- provide governments with a technical base for health, safety and environmental legislation, and conformity assessment
- share technological advances and good management practice
- disseminate innovation

---

<sup>1</sup> [http://www.iso.org/iso/about/discover-iso\\_what-standards-do.htm](http://www.iso.org/iso/about/discover-iso_what-standards-do.htm)

- safeguard consumers, and users in general, of products and services
- make life simpler by providing solutions to common problems

Pochopiteľne, uvedené ciele sledujú aj iné štandardy, ako tie, ktoré vydáva ISO. Popri záväzných (de jure) štandardoch existujú aj tzv. de facto štandardy a dobrovoľné štandardy. Prvý typ (de facto štandardy) sú prevládajúce, všeobecne akceptované riešenia, ktoré však neboli formálne publikované v podobe štandardov. Druhým typom sú štandardy, ktoré majú charakter odporúčaní. To, aký charakter majú vydávané štandardy závisí od legislatívy upravujúcej postavenie technických noriem. Podľa slovenskej legislatívy majú normy vo všeobecnosti charakter odporúčaní. Povinné sú normy, ktorých záväznosť upravuje niektorý zákon. Napriek tejto zdanlivo slabej vynúiteľnosti noriem zohrávajú normy a štandardy v spoločnosti významnú úlohu a ich význam bude zrejme narastať. Normy a štandardy<sup>2</sup> sa vytvárajú v priebehu štandardizácie.

Štandardizácia je proces vývoja technických štandardov a rokovania vedúcemu k prijatiu navrhovaného štandardu. Súčasťou štandardizácie je aj správa štandardov, ktorá spočíva v udržovaní aktuálnosti prijatých štandardov (revízia štandardov, návrh na doplnenie, prepracovanie, zrušenie štandardu, nahradenie štandardu iným štandardom a pod.).

## 2.2 Štandardizačný proces

Schvaľovanie a vydávanie noriem je zložitý a administratívne značne náročný proces, ktorý sa v jednotlivých štandardizačných organizáciách môže (vzhľadom na charakter problematiky, organizačnú štruktúru a i.) odlišovať. My budeme vychádzať z dokumentov ISO[2] a dokumentu ISO/IEC JTC 1 Directives, 5th Edition Version 3.0 [3]. Podľa nich sa štandardizačný proces

---

<sup>2</sup> v medzinárodnej terminológii sa používa pojem štandard vo význame norma, v slovenskej terminológii má pojem štandard ešte aj užší význam – podzákonný dokument upresňujúci požiadavky stanovené zákonom (napr. štandardy ISVS). My budeme v tejto práci chápať pojmy štandard a norma ako synonymá.

skladá z nasledujúcich troch základných fáz:

- 1.** *Prvotný podnet (požiadavka) na tvorbu nového štandardu* väčšinou vychádza z priemyselného sektoru, teda od koncového používateľa. Ten podá žiadosť na národný štandardizačný útvar, ktorý následne predloží návrh organizácii ISO. Keď je potreba tohto štandardu preukázaná a formálne odsúhlasená, pracovné skupiny, zložené z odborníkov v danej oblasti, vypracujú prvú formálnu definíciu budúceho štandardu.
- 2.** *Fáza rokovaní* medzi jednotlivými pracovnými skupinami a národnými štandardizačnými útvarmi o jednotlivých detailoch daného štandardu. Rokovania prebiehajú, až kým sa nenájde najlepšie technické riešenie problému, ktorý má daný štandard vyriešiť. Výsledkom týchto rokovaní je *návrh medzinárodného štandardu (Draft International Standard)*. Ten je pripomienkovaný zvyčajne po dobu 5 mesiacov. Následne prebehnú ďalšie hlasovania oprávnených členov ISO a pokiaľ skončia úspešne (aspoň 2/3 pozitívnych hlasov), návrh medzinárodného štandardu prejde do etapy *konečný návrh medzinárodného štandardu (Final Draft International Standard)*. V prípade neúspešného hlasovania sa opakujú etapy rokovaní a pripomienkovaní, až kým nie sú splnené podmienky postupu do ďalšieho štádia (teda odstránené všetky nedostatky, vyriešené všetky problémy a úspešný výsledok hlasovania).
- 3.** *Oficiálne schválenie konečného návrhu medzinárodného štandardu, a jeho publikovanie ako medzinárodného štandardu ISO.*

Vidíme, že najmä fáza rokovaní zahrňuje viacero etáp. Počas nich sa vyprodukuje mnoho dokumentov rôzneho typu. Dokumenty vytvárajú tak členovia JTC 1 (jednotlivci, národné štandardizačné inštitúcie) ako aj pracovné skupiny, podvýbory, riadiace, resp. administratívne orgány JTC a i. Každý z dokumentov má označenie, podľa ktorého možno určiť etapu, na ktorú sa vzťahuje. Prehľad týchto etáp a ich označení nájdeme na *Obr. 2.1*, prebratom zo stránky ISO [4]. Celý tento proces od požiadavky po schválenie normy môže trvať 3-5 rokov a nie všetky štandardy ho skončia úspešne. Niekedy sa stáva,

že až v neskorších fázach vývoja sa ukáže jeho neopodstatnenie, alebo iný dôvod, pre ktorý sa vývoj štandardu pozastaví, prípadne úplne ukončí.

## **2.3 Štandardizačný materiál**

Podľa definície ISO/IEC Guide 2: 1996 norma (štandard) je dokument, vytvorený na základe dohody a schválený uznaným orgánom, ktorý poskytuje na bežné a opakované používanie pravidiel, usmernenia (návody alebo charakteristiky) pre činnosti alebo ich výsledky tak, aby sa dosiahol optimálny stupeň usporiadania v danom kontexte (v danej súvislosti) [5].

Pre potreby bakalárskej práce však budeme používať termín štandardizačný materiál (ďalej ŠM) na každý publikovaný dokument, ktorý vznikne ako výsledok alebo medzivýsledok štandardizačného procesu. Teda nie len konečná publikovaná podoba normy, ale tiež všetky jej pracovné formy a návrhy.

## International harmonized stage codes

STAGE	SUBSTAGE						
				90 Decision Substages			
	00 Registration	20 Start of main action	60 Completion of main action	92 Repeat an earlier phase	93 Repeat current phase	98 Abandon	99 Proceed
00 Preliminary stage	00.00 Proposal for new project received	00.20 Proposal for new project under review	00.60 Close of review			00.98 Proposal for new project abandoned	00.99 Approval to ballot proposal for new project
10 Proposal stage	10.00 Proposal for new project registered	10.20 New project ballot initiated	10.60 Close of voting	10.92 Proposal returned to submitter for further definition		10.98 New project rejected	10.99 New project approved
20 Preparatory stage	20.00 New project registered in TC/SC work programme	20.20 Working draft (WD) study initiated	20.60 Close of comment period			20.98 Project deleted	20.99 WD approved for registration as CD
30 Committee stage	30.00 Committee draft (CD) registered	30.20 CD study/ballot initiated	30.60 Close of voting/ comment period	30.92 CD referred back to Working Group		30.98 Project deleted	30.99 CD approved for registration as DIS
40 Enquiry stage	40.00 DIS registered	40.20 DIS ballot initiated: 5 <i>months</i>	40.60 Close of voting	40.92 Full report circulated: DIS referred back to TC or SC	40.93 Full report circulated: decision for new DIS ballot	40.98 Project deleted	40.99 Full report circulated: DIS approved for registration as FDIS
50 Approval stage	50.00 FDIS registered for formal approval	50.20 FDIS ballot initiated: 2 <i>months</i> . Proof sent to secretariat	50.60 Close of voting. Proof returned by secretariat	50.92 FDIS referred back to TC or SC		50.98 Project deleted	50.99 FDIS approved for publication
60 Publication stage	60.00 International Standard under publication		60.60 International Standard published				
90 Review stage		90.20 International Standard under periodical review	90.60 Close of review	90.92 International Standard to be revised	90.93 International Standard confirmed		90.99 Withdrawal of International Standard proposed by TC or SC
95 Withdrawal stage		95.20 Withdrawal ballot initiated	95.60 Close of voting	95.92 Decision not to withdraw International Standard			95.99 Withdrawal of International Standard

Obr. 2.1: Prehľad kódov pracovných dokumentov pre jednotlivé štádiá vývoja medzinárodného štandardu ISO [4]

## 3. Analýza

Pred samotným vývojom aplikácie je potrebné zodpovedať na základné otázky o systéme, ako napríklad kto s ním bude pracovať, čo od systému očakáva a aké informácie mu budú poskytnuté. Taktiež bude potrebné riešiť spôsob ochrany prístupu k dátam, t.j. ako zabezpečiť, aby sa používateľ nedostal k informáciám, na ktoré nemá oprávnenie a definovať potrebné bezpečnostné požiadavky.

Špecifikáciu pre túto aplikáciu som čerpala z diplomovej práce Martina Kraloviča [1]. Spolu s doc. Danielom Olejárom, vedúcim mojej bakalárskej práce, ma uviedli do problematiky šandardizácie a šandardizačného procesu. Vďaka osobným skúsenostiam s prácou pre technickú komisiu SÚTN mi pomohli pochopiť, akú funkcionality by mala aplikácia mať.

### 3.1 Používatelia a ich oprávnenia

Náš systém je predovšetkým určený členom technickej komisie TK 37 SÚTN. No vzhľadom na to, že sa očakáva jeho prístupnosť pre ďalších záujemcov o normy a štandardy z oblasti informačnej bezpečnosti na internete, vzniká potreba autentifikácie používateľov a kontrola ich oprávnení. V systéme sú nasledujúce 4 základné skupiny používateľov, ale v prípade potreby sa môžu pridať ďalšie.

- administrátor aplikácie - má oprávnenie na všetky akcie, vrátane prehliadania auditných záznamov, teda po prihlásení do systému má prístupné všetky druhy menu:
  - ➔ administrácia používateľov
  - ➔ administrácia obsahu
  - ➔ prehľad všetkých pracovných skupín

- prehliadanie auditných záznamov
- administrátor používateľov - má na starosti správu používateľov, teda prístupné má len menu pre administráciu používateľov:
  - vytváranie nových používateľských účtov
  - úprava existujúcich, uzamknutie neaktívneho účtu, otvorenie uzamknutého účtu
  - vytváranie pracovných skupín
  - úprava existujúcich pracovných skupín
- administrátor obsahu - má na starosti činnosti spojené so správou ŠM, teda má prístupné len menu pre administráciu obsahu:
  - pridávanie a úprava materiálov
  - vyhľadávanie materiálov
  - vytváranie a úprava vzťahov
  - pridávanie a úprava kritérií a kategórií pre ŠM
- tvorca obsahu - môže manipulovať s materiálmi na základe pridelených právomocí a pracovných skupín, teda má prístupné len používateľské menu:
  - pridávanie materiálov
  - vyhľadávanie materiálov
  - úprava svojich a iných pridelených materiálov

Pre prácu v systéme musí byť používateľ prihlásený. Nové používateľské konto môže vytvoriť len administrátor používateľov, príp. administrátor aplikácie. Rovnako môže upravovať údaje o používateľoch, ich zaradenie do pracovnej skupiny a v prípade potreby deaktivovať používateľský účet.

Tvorcom obsahu je každý používateľ, ktorý je zaradený aspoň do jednej pracovnej skupiny. Členovia pracovnej skupiny majú prístup len k dokumentom, ktoré sú priradené k ich pracovnej skupine.

## 3.2 Štandardizačný materiál

Ako sme už spomínali, poslaním aplikácie je práca so štandardizačnými materiálmi. S pojmom štandardizačný materiál sme sa v stručnosti oboznámili v kapitole 2.3. Teraz sa zameriame na tie vlastnosti štandardizačných materiálov, ktoré sú podstatné pre našu aplikáciu.

Každý štandardizačný materiál má základné vlastnosti, ako napr:

- Označenie
- Názov
- Dátum vydania
- Počet strán
- Abstrakt

Tieto vlastnosti treba vyplniť pri vkladaní nového štandardizačného materiálu do systému. Materiál je tiež priradený k pracovným skupinám, ktoré s ním môžu pracovať. Administrátor aplikácie a administrátor obsahu môžu materiál priradiť ku ktorejkoľvek pracovnej skupine. Ak však materiál vkladá člen pracovnej skupiny, môže ho priradiť len ku pracovným skupinám, v ktorých sa sám nachádza. Používateľ, ktorý vkladá nový ŠM do systému je zároveň určený ako jeho vlastník. Vlastníctvo sa dá zmeniť na iného používateľa rovnako, ako je možné upraviť ostatné vlastnosti štandardizačného materiálu.

Keď hovoríme o vkladaní materiálu do systému, nemáme na mysli len informácie o ňom, ale samozrejme aj uloženie súboru s obsahom daného štandardizačného materiálu. Stiahnuť si ho môže používateľ s príslušnými právami (teda vlastník materiálu, člen pracovnej skupiny, administrátor obsahu alebo administrátor aplikácie).

Ďalej je materiál zaradený do kategórií v jednotlivých kritériách, čo umožňuje vyhľadať ŠM podľa týchto kritérií.



### 3.2.1 Kritériá a kategórie štandardizačných materiálov

Kritériá a kategórie sú dôležitým prvkom pri kategorizovaní materiálov a ich vyhľadávaní. Štandardizačný materiál môže byť do kategórií zaradený buď pri vytváraní nového, alebo neskôr pri úprave existujúceho materiálu. Príklad kritérií a ich kategórií sa nachádza v *Tab. 3.1*.

<i>Kritérium</i>	<i>Kategórie</i>
Vydavateľ	<ul style="list-style-type: none"><li>● ISO</li><li>● NIST</li><li>● BSI</li><li>● ....</li></ul>
Obsahové zameranie	<ul style="list-style-type: none"><li>● kryptológia</li><li>● audit</li><li>● vývoj systémov</li><li>● certifikácia produktov</li><li>● ...</li></ul>

*Tab. 3.1: Príklad kritérií a ich kategórií*

Každé kritérium má dve nastaviteľné vlastnosti, ktoré sme nazvali mandatory a multiple. Vlastnosť mandatory znamená, že každý materiál musí byť zaradený aspoň do jednej kategórie tohto kritéria. Vlastnosť multiple zas určuje pre materiál maximálny počet kategórií v rámci kritéria, teda či môže byť materiál zaradený do jednej alebo viacerých kategórií tohto kritéria. Pre lepšie pochopenie opäť nazrime do *Tab. 3.1*. Kritérium "Vydavateľ" bude určite mandatory, lebo je dôležité vedieť, kto vydáva daný štandard. Zároveň môžeme predpokladať, že tento štandard vydáva len jedna organizácia, teda kritérium nebude multiple. Naproti tomu kritérium "Obsahové zameranie" nemusí byť nutne mandatory a keďže nie je vždy možné jednoznačne určiť zameranie štandardu, povolíme jeho zaradenie do viacerých kategórií z hľadiska obsahu označením kritéria ako multiple.

Kritériá a kategórie do systému vkladá administrátor obsahu, resp.

administrátor aplikácie. Prvým krokom je vytvoriť kritérium s unikátnym názvom. Ďalej rozhodne, či kritérium je multiple. Vlastnosť mandatory je implicitne nastavená na "nie je mandatory" a možno ju upravovať až po uložení kritéria. Toto riešenie sme zvolili z dôvodu udržania konzistentnosti dát v databáze. Tá by totiž bola narušená, keďže kritérium v čase ukladania ešte nemá vytvorené kategórie a teda ani žiadny z existujúcich štandardizačných materiálov do nich nemôže byť zaradený.

Po úspešnom uložení kritéria mu administrátor vytvorí kategórie, môže ich upravovať a vymazávať. Administrátor tiež môže upravovať vlastnosti vytvorených kritérií, ale len pod podmienkou, že nebude narušená konzistentnosť databázy. To znamená, že kritérium môže byť označené ako mandatory len pokiaľ sú všetky existujúce materiály zaradené aspoň do jednej kategórie tohto kritéria. Podobne maximálny počet kategórií materiálu môže byť určený na "jednu" (nie je multiple) len pokiaľ žiadny materiál nie je zaradený do viacerých kategórií tohto kritéria. Na všetky neželané situácie je administrátor upozornený chybovým hlásením, spolu s vypísaním zoznamu materiálov, ktoré porušujú tieto pravidlá.

Rovnako dôslední musíme byť aj pri vymazávaní kritérií a kategórií. Pred vymazaním kritéria musí administrátor potvrdiť svoje rozhodnutie vymazať toto kritérium a všetky jeho kategórie. Kritérium sa dá vymazať len v prípade, že žiadny z materiálov nie je zaradený do žiadnej kategórie tohto kritéria. V opačnom prípade bude administrátor upozornený chybovou správou spolu so zoznamom ovplyvnených materiálov. Taktiež pri vymazávaní kategórie sa zobrazí upozornenie, pokiaľ niektorý z materiálov je zaradený do vymazávanej kategórie.

## 4. Návrh riešenia

Pred samotnou implementáciou systému treba vybrať vhodné nástroje. Nemôžeme zabudnúť na dôležitosť štandardov, z používateľského hľadiska je potrebné, aby sa stránky správne zobrazovali v najrozšírenejších webových prehliadačoch.

Po určení platformy, na ktorej bude aplikácia stavaná, treba vybrať vhodnú architektúru. Naším cieľom je čo najprehľadnejší kód tak, aby pri dopĺňaní ďalšej funkcionality v neskorších fázach vývoju aplikácie neboli potrebné rozsiahle zmeny v existujúcom kóde. Z vlastných skúseností viem, že veľkú pozornosť treba venovať aj správne návrhu databázy, nakoľko jej neskoršie zmeny môžu mať nepriaznivý vplyv na už implementovanú časť aplikácie.

### 4.1 Výber technológií

Keďže pracujeme na webovej aplikácii, pri výbere platformy sme zvažovali viacero možností. Každá z nich má svoje výhody aj nevýhody. My sme sa rozhodovali najmä medzi nasledujúcimi dvomi:

- PHP
- ASP.NET

PHP je všestranný OpenSource skriptovací jazyk s podporou objektového programovania. Navyše je multiplatformový, možno ho prevádzkovať na väčšine webovských serverov a operačných systémov. Taktiež vie spolupracovať s rôznymi databázovými servermi. Asi najznámejšia je kombinácia Apache + PHP + MySQL, prístupná okrem iného aj v inštalačnom balíku známom ako PHP Triad.

Na rozdiel od PHP, ktorý sa obyčajne používa pri vývoji jednoduchších aplikácií, ASP.NET je silný nástroj, vhodný aj pre rozsiahle systémy. Používa diametrálne odlišný prístup, ktorý je podobný písaniu desktopových aplikácií pre systém Windows. Pracuje so serverovými komponentmi, ktoré majú svoje vlastnosti, metódy a udalosti (napríklad `TextBox.Text`, `Button_OnClick()`), ale aj zložitejšími komponentmi pre prácu s dátami z databázy ako GridView, Repeater a mnoho ďalších.

Ďalšou výhodou ASP.NET oproti PHP je, že aplikácie sú rýchlejšie, lebo kód je predkompilovaný. ASP.NET beží na platforme .NET Framework, ktorý podporuje široký výber klasických programovacích jazykov ako C#, VisualBasic, J# a po stiahnutí kompilátorov možno použiť taktiež množstvo ďalších jazykov. Rovnako podporuje aj spoluprácu týchto jazykov. Konfiguračné údaje sú uložené v XML súboroch, sú jednoducho prístupné a modifikovateľné. Napriek tomu, že ASP.NET bol vyvíjaný pre systém Windows, v súčasnosti sa dá sprevádzkovať aj v operačných systémoch ako Linux, Mac OS X a podobne. Taktiež podporuje rôzne webovské servery, (IIS, Apache, ...) a databázové servery (Microsoft SQL Server, MySQL, Oracle, PostgreSQL, ...).

Pre mňa ako vývojára je tiež dôležité, že ASP.NET podobne ako PHP, je možné používať bez dodatočných licencií. Navyše poskytuje príjemné vývojové prostredie *Visual Web Developer 2005 Express Edition*, ktoré je možno stiahnuť na stránkach Microsoftu.

Po zvážení všetkých výhod, nevýhod a odporúčaní som sa napriek slabým osobným skúsenostiam rozhodla pre ASP.NET 2.0 na platforme Microsoft .NET Framework 2.0. Z programovacích jazykov som dala prednosť jazyku C#, ktorý je syntaxou podobný jazykom C++ a Java. Ako webový a databázový server sme zvolili Microsoft Server IIS (Internet Information Services) a Microsoft SQL Server 2005.

## 4.2 Štandardy a kompatibilita s prehliadačmi

Na úvod treba spomenúť, že ani dôsledné dodržiavanie štandardov zo strany vývojárov webových aplikácií nezaručuje, že sa ich stránky zobrazia vo všetkých prehliadačoch rovnako. V prvom rade záleží od výrobcov prehliadačov, akú podporu venujú danému štandardu. Táto podpora prichádza s určitým oneskorením niekedy až niekoľkých rokov. Niektorí si naopak existenciu štandardov nevšímajú a vyrábajú si vlastné. Nedodržiavanie štandardov vedie k nekompatibilite produktov, s čím má v konečnom dôsledku najviac roboty samotný vývojár, lebo musí venovať dodatočných čas optimalizácii svojej aplikácie pre rôzne prehliadače.

Štandardizácií pre World Wide Web sa venuje medzinárodná organizácia The World Wide Web Consortium, známa ako W3C [6]. Zaoberá sa aj vzdelávaním, vývojom softwaru a slúži ako fórum pre diskusie o webe. Vyprodukovala viacero štandardov, ktoré pokrývajú rôzne oblasti webu, od základných (*HTML*, *XHTML*, *CSS*, ... ), cez graficky orientované (*SVG*, *WebCGM*, *MathML*, ...) až po *WCAG*, *ATAG* zameriavajúce sa na hendikepovaných užívateľov.

Pre potreby našej aplikácie sme vybrali nasledujúce štandardy:

- XHTML 1.0 Strict
- CSS 2.1

Štandard XHTML 1.0 Strict vyžaduje čistý validný kód podobného typu ako XML. V praxi to znamená, že každý tag musí mať uzatvárací tag (vrátane nepárových tagov ako `<br />`), nepovoľuje sa prelínanie tagov, názvy tagov a parametrov sa píšú malými písmenami a hodnoty parametrov musia byť v úvodzovkách.

Aby prehliadač vedel, s akým typom dokumentu má do činenia a podľa ktorého štandardu ho má zobrazíť, je dobré definovať to na začiatku dokumentu:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML 1.0 Strict nepovoľuje definovanie layoutu (vzhľadu) priamo v kóde. Na tento účel sa používa CSS (Cascading Style Sheets), a práve tu vzniká spomenutý problém s kompatibilitou rôznych prehliadačov. Na druhej strane CSS je efektívne riešenie, lebo umožňuje jednoducho upravovať design stránky bez nutnosti zasahovať do kódu. Taktiež je možné úplne vymeniť celý .css súbor a nastaviť stránke odlišný vzhľad, prípadne dynamicky striedať viacero súborov. CSS súbor treba taktiež definovať v hlavičke dokumentu:

```
<link href="Style.css" rel="stylesheet" type="text/css" />
```

Pokiaľ ide o kompatibilitu našej aplikácie s prehliadačmi, zameriame sa najmä na tie s najväčším zastúpením na trhu:

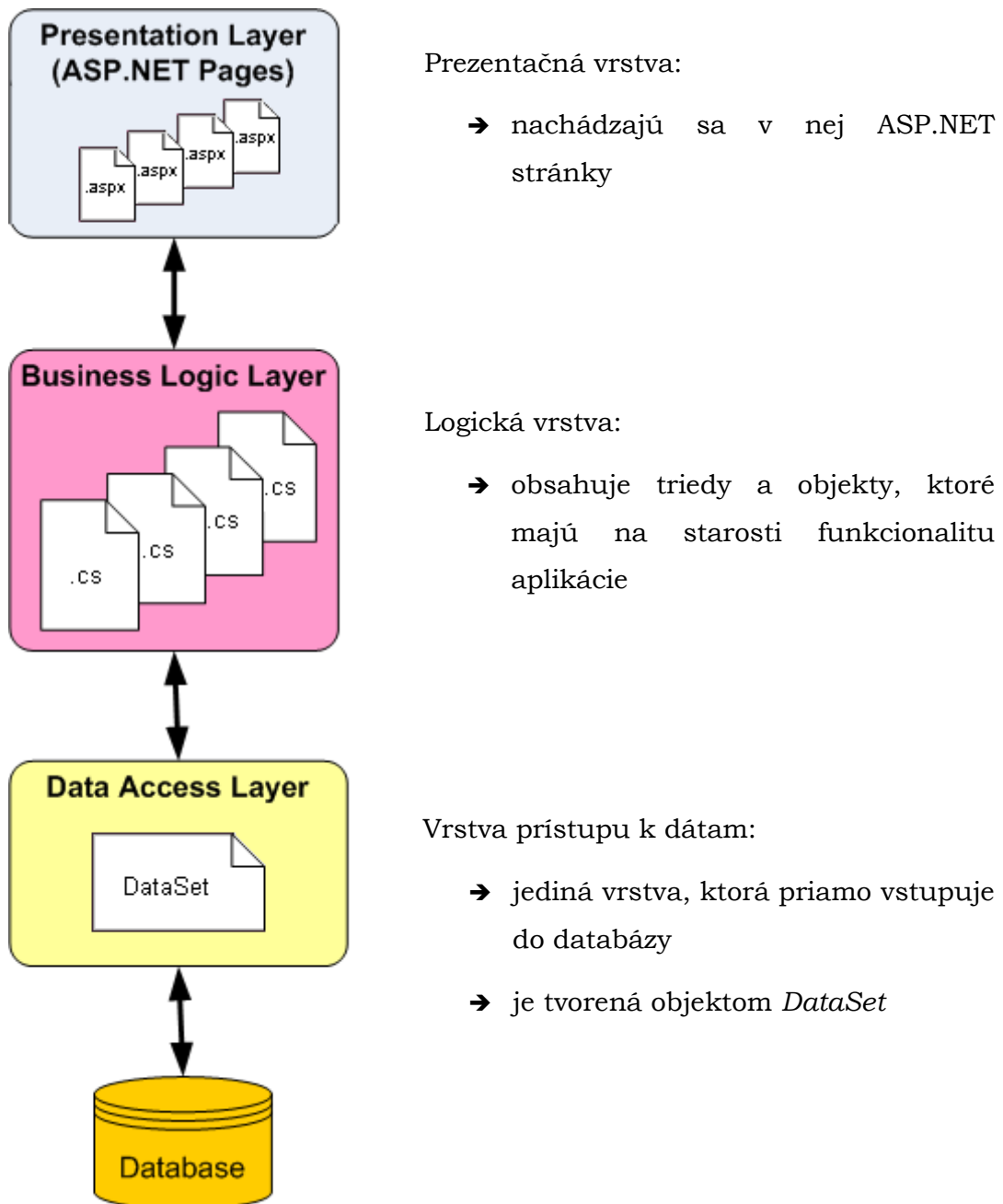
- Microsoft Internet Explorer 6.0
- Windows Internet Explorer 7.0
- Mozilla Firefox 2.0

Nakoľko stránky obsahujú diakritiku, bolo potrebné vybrať kódovanie s ich priamou podporou. Ako najlepšie sa javí kódovanie UTF-8.

## **4.3 3 - vrstvá architektúra**

Pri návrhu aplikácie sme sa snažili dodržiavať zásady 3-vrstvovej architektúry [7]. Hlavným princípom architektúry pomocou vrstiev je, že vrstva môže priamo komunikovať len s vrstvou pod ňou a odpovedať na požiadavky vrstvy nad ňou. Rozdelením aplikácie na vrstvy sme dosiahli vyššiu prehľadnosť systému a jeho väčšiu flexibilitu.

Pri 3-vrstvovej architektúre sa udávajú rozličné prístupy. Niektoré považujú databázu priamo za jednu z vrstiev, iné medzi vrstvy zaraďujú len vytvorené súbory s kódom. Pri vývoji aplikácie som sa priklonila k druhému prístupu, zobrazenému na Obr. 4.1.



Obr. 4.1: Schéma 3-vrstvovej architektúry

ASP.NET poskytuje komponenty, ktoré priamo vstupujú do databázy a pomocou komponentov ako *GridView*, *DataList* a podobne zobrazujú zvolené dáta na stránke. Pomocou týchto komponentov je možné rýchlo vytvoriť jednoduchú aplikáciu pre prácu s údajmi v databáze. Avšak takéto komponenty narúšajú princípy 3-vrstvovej architektúry. Preto sme sa rozhodli použiť objekt *DataSet*. Tento objekt obsahuje objekty dvoch typov - *DataTable* a *TableAdapter*. Pomocou nich sú zapuzdrené jednotlivé SQL dotazy (SELECT, INSERT, UPDATE, DELETE) do metód. Objekt *DataSet* má ako jediný prístup do databázy. V našej aplikácii tvorí Data Access Layer, teda Vrstvu prístupu k dátam.

Druhá vrstva je logická, tzv. Business Logic Layer. Mala by obsahovať celú aplikačnú logiku systému. Vykonáva všetky rozhodnutia aplikácie, spracováva dáta a sprostredkúva výmenu dát medzi dátovou a prezentačnou vrstvou. Obvykle sa logická vrstva implementuje ako samostatný projekt, ale v našej aplikácii ju tvorí sada tried pre prácu s *DataSetom* a ďalšie pomocné triedy. Pre každý *DataTable* v Data Access Layer je vytvorená samostatná trieda, ktorá sprostredkuje metódy pre obsluhu jednotlivých dotazov tohto *DataTablu* prezentačnej vrstve. Zároveň poskytuje ďalšie funkcie. Napríklad trieda *Users*, určená pre obsluhu *UserTableDataTable*, obsahuje okrem iných aj nasledujúce funkcie:

*IsUnique(string username)* - zistí, či sa zadané používateľské meno už nachádza v databáze,

*IsActive(string username)* - overí, či je užívateľský účet aktívny

Poslednou vrstvou je už spomenutá prezentačná vrstva, tzv. Presentation Layer. V našom prípade pozostáva z ASP.NET stránok (majú príponu .aspx). Každá z nich sa skladá z 2 častí: HTML kódu so serverovými komponentami a Code behind, tzv. kód v pozadí. Code behind môže byť písaný v jednom z podporovaných jazykov, ktorým som sa venovala v kapitole 4.1. Tento jazyk treba definovať na začiatku .aspx stránky. Ja som používala jazyk C#:



```
<%@ Page Language="C#" %>
```

Code behind reaguje na udalosti (eventy) komponentov a stará sa o interaktivitu stránky. Môže byť v tom istom súbore ako HTML a komponenty, v špeciálnej sekcii ohraničený tagmi:

```
<script runat="server">  
    //....  
</script>
```

Taktiež ho môžeme vyčleniť do samostatného súboru. Súbory v jazyku C# majú príponu .cs.

Pri Code behind sa nie vždy dá jednoznačne určiť, či je súčasťou prezentačnej alebo logickej vrstvy. Väčšinou Code behind spravuje webové komponenty, reaguje na ich udalosti a teda sa zaraďuje do prezentačnej vrstvy. Niekedy sa však nevyhneme zložitejšiemu rozhodovaniu, prípadne môže obsahovať bussiness objekt (v ideálnom prípade by ho obsahovať nemal) a vtedy ho možno považovať za súčasť logickej vrstvy.

## 4.4 Návrh databázy

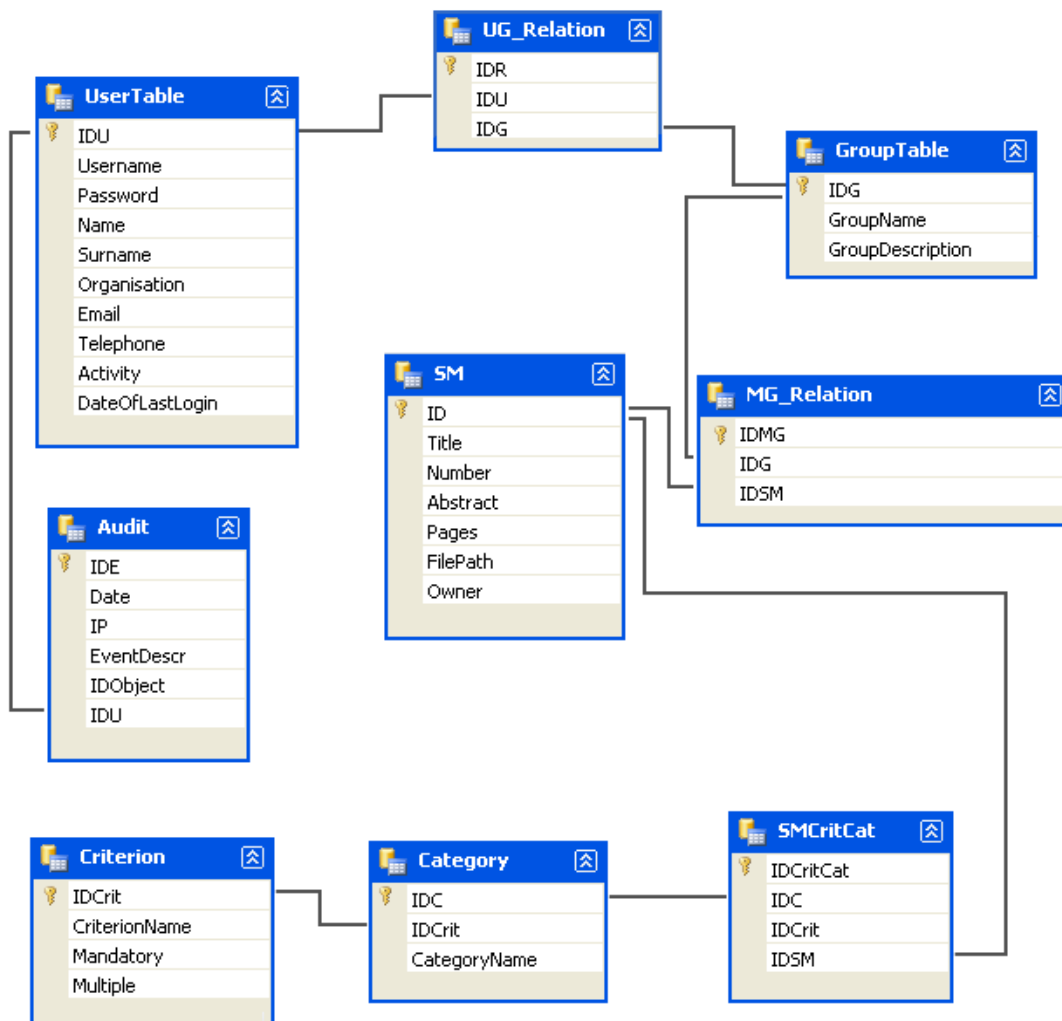
Nevyhnutným krokom pri návrhu aplikácie na správu štandardizačných materiálov je správny návrh databázy. Databáza obsahuje 9 tabuliek, ale predpokladá sa, že tento počet ešte narastie s pribúdajúcou funkcionalitou. Schéma databázy sa nachádza na *Obr. 4.2*.

Stručný popis jednotlivých tabuliek:

- **UserTable** - obsahuje všetky potrebné údaje o používateľoch, teda používateľské meno, heslo, krstné meno a priezvisko, organi-

záciu, email, telefónne číslo, informáciu, či je konto aktívne a taktiež dátum posledného prihlásenia.

- **GroupTable** - obsahuje informácie o pracovných skupinách, teda názov pracovnej skupiny a jej popis. Prvé tri pracovné skupiny obsahujú informácie o skupinách pre Administrátora aplikácie, Administrátora používateľov a Administrátora obsahu.
- **UG\_Relation** - v tejto tabuľke je zaznamenaná príslušnosť používateľov ku skupinám. Používateľ môže byť vo viacerých skupinách zároveň.
- **SM** - obsahuje informácie o štandardizačných materiáloch, ako ich názov, označenie, abstrakt, počet strán, dátum vydania, názov súboru a vlastník materiálu.
- **MG\_Relation** - táto tabuľka zaznamenáva príslušnosť materiálu ku prac. skupine, teda ktorá prac. skupina môže s materiálom pracovať.
- **Criterion** - obsahuje prehľad všetkých kritérií a ich vlastností, konkrétne názov kritéria, informáciu, či je kritérium mandatory a multiple.
- **Category** - v tejto tabuľke sa nachádza prehľad všetkých kategórií a ich príslušnosť ku kritériu.
- **SMCritCat** - táto tabuľka zaznamenáva, do ktorých kategórií v ktorých kritériách sú zaradené štandardizačné materiály.
- **Audit** - do tejto tabuľky sa zapisujú auditné záznamy, kedy došlo k udalosti, z akej IP prišla požiadavka, ak je používateľ prihlásený, zaznamená sa jeho ID, taktiež ID objektov, ktorých sa týka daná udalosť a samozrejme popis udalosti. Jedná sa napríklad o úspešné i neúspešné prihlásenie do systému, pridanie nového objektu (používateľ, ŠM, ..), úprava objektu, vymazanie objektu ...



Obr. 4.2: Schéma tabuliek v databáze

## 5. Vývoj aplikácie

Až pri samotnej implementácii systému sa naplno prejaví výhody i nedostatky zvolených prostriedkov, architektúry a návrhu. Pri používaní nového prostredia ASP.NET sa často odhalili zázračne jednoduché riešenia, ale inokedy sa bolo treba potrápiť s drobnosťami, ktoré sa na iných platformách riešili bez najmenších problémov.

### 5.1 ASP.NET

ASP.NET 2.0 prináša viacero výhod, ktoré zjednodušujú a urýchľujú vývoj webových aplikácií. V prvom rade sú to tzv. Master Pages, Themes a Web Parts, pomocou ktorých sa dá jednoducho nastaviť jednotný vzhľad celej aplikácie. Master Page sa používa ako šablóna pre ostatné stránky. Prvky, ktoré obsahuje sa následne zobrazia na každej ďalšej stránke. Je to jednoduché riešenie pre zobrazenie hlavičky, pätičky a navigácie. Master Page môže obsahovať aj základnú funkcionálnosť a skripty platné pre všetky stránky.

Themes, alebo Motívy taktiež pomáhajú prispôbiť vzhľad aplikácie. Môžu sa použiť na jednotlivé serverové komponenty, stránky, prípadne celú aplikáciu. Ja som ich použila na zjednotenie vzhľadu všetkých použitých komponentov GridView (budeme sa mu venovať neskôr).

Ďalším prínosom ASP.NET je komponent pre navigáciu. Dá sa vybrať z viacerých typov navigácie:

- *Menu* - po prechode kurzorom ponad položku menu sa vyvolá podmenu danej položky
- *TreeView* - zobrazí menu v podobe klasickej stromovej štruktúry

- *SiteMapPath* - slúži na jednoduchšiu orientáciu, zobrazuje používateľovi polohu aktuálne navštívenej stránky a cestu od domovskej stránky po aktuálnu

Pre našu aplikáciu mi však žiadny zo spomenutých komponentov nevyhovoval, tak som využila inú možnosť, ktorú poskytuje ASP.NET. Vyrobila som si vlastné komponenty, tzv. User Web Controls. Výhodou tohto prístupu je, že dovoľuje používať pôvodné serverové komponenty, združovať ich a upravovať podľa vlastnej potreby. Výsledok sa dá použiť ako rovnocenný webový komponent, ale navyše má vlastnosti, ktoré pôvodné komponenty nemajú.

Týmto spôsobom som vyrobila menu spĺňajúce potreby aplikácie:

- *mainMenu* - obsahuje úvodné menu, je zobrazené všetkým používateľom, prihláseným aj neprihláseným
- *adminMenu* - obsahuje menu pre správu používateľov (pozri kapitolu 3.1 - administrátor používateľov)
- *smMenu* - obsahuje menu pre správu štandardizačných materiálov (pozri kapitolu 3.1 - administrátor obsahu)
- *auditMenu* - umožňuje zobrazíť auditné záznamy, prístup k nim má len administrátor aplikácie
- *groupMenu* - zobrazuje zoznam pracovných skupín a ku každej je priradené podmenu pozostávajúce z položiek "Novinky" a "Prehľad dokumentov"

Každý z týchto komponentov sa skladá z dvoch častí - názov daného menu (text uzavretý v tagu `<p>` `</p>`) a serverový komponent *Menu* s potrebnými položkami pre dané menu. Jednoduchým príkladom je *auditMenu*:

```
<%@ Control Language="C#" AutoEventWireup="true"
    CodeFile="auditMenu.ascx.cs"
    Inherits="WebControls_auditMenu" %>
```

```

<p class="title"> Audit </p>

<asp:Menu ID="Menu1" runat="server">

    <Items>

        <asp:MenuItem Text="Auditné záznamy" Value="Audit"
            NavigateUrl="~/admin/audit.aspx">

        </asp:MenuItem>

    </Items>

</asp:Menu>

```

Výnimkou je len posledný web control *groupMenu*, ktorý síce tiež obsahuje názov, no položky menu nie sú len jednoducho priradené ako parametre webového komponentu. Celé *groupMenu* je generované z databázy na základe pracovných skupín, do ktorých je zaradený používateľ.

Riešenie vytvoriť vlastné menu pre každú skupinu užívateľov má aj tú výhodu, že na stránke sa zobrazí len to menu, na ktoré má používateľ oprávnenie. To sa rozhodne hneď pri prihlasovaní používateľa a po dobu používania aplikácie sa prenáša pomocou Session.

Pri úprave vlastností existujúcich komponentov je aj iný prístup ako vytvoriť User Web Control. Vývojár môže napísať vlastnú triedu a pomocou dedenia na ňu preniesť vlastnosti rodičovského komponentu. Týmto spôsobom som vyrobila *MyCriterionListBox*. Tento web control sa používa pri zaraďovaní ŠM do kategórií v rámci kritérií. Každé kritérium je zobrazené ako jeden *MyCriterionListBox* a jeho hodnoty sú kategórie daného kritéria.

Potreba obdobného komponentu vyplynula z vlastností kritérií, ktoré môžu byť *mandatory* a *multiple* (pozri kapitolu 3.2.1). V praxi to znamená, že pri *mandatory* kritériu daný ListBox musí mať označenú nejakú položku. Pokiaľ je kritérium *multiple*, aj ListBox musí dovoliť označenie viacerých položiek.

Vyrobila som teda triedu *MyCriterionListBox* a oddedila ju od triedy

*ListBox*. Oproti klasickému *ListBoxu* má navyše atribúty *mandatory*, *multiple* a *criterion*. Atribút *criterion* je vyplnený pri vytvorení *MyCriterionListBoxu*. Následne sa z databázy vyberie príslušné kritérium a nastaví sa ostatné dva atribúty. Hodnota pre *criterion* sa nastaví serverovým komponentom *Repeater*, ktorý prechádza všetkými kritériami a pre každé vyrobí príslušný *MyCriterionListBox*.

Za zmienku ešte stojí spôsob, akým je na stránku vkladáný *MyCriterionListBox*. Obvykle sa totiž pre serverové komponenty používa tag *asp*, napríklad pre *Label* by vyzeral nasledovne:

```
<asp:Label ID="LabelID" runat="server" Text="text">
</asp:Label>
```

Avšak pre *MyCriterionListBox* bolo potrebné na začiatku dokumentu zadať namespace, čím mu bol vytvorený aj vlastný tag *My1*:

```
<%@ Register TagPrefix="My1" Namespace="MyWebControls" %>
```

Po definovaní namespace sa dá *MyCriterionListBox* vkladať na stránku podobne ako ktorýkoľvek iný web control.

```
<My1:MyCriterionListBox runat="server"
    DataSourceID="RepeatCatDataSource"
    DataTextField="CategoryName"
    DataValueField="IDC"
    criterion='<%# Eval("IDCrit") %>' >
</My1:MyCriterionListBox>
```

Dôležitou súčasťou aplikácie je aj prístup do databázy a forma

zobrazenia týchto dát užívateľovi. ASP.NET 2.0 v tomto prípade ponúka viacero možností ako sa s týmto problémom vysporiadať. Medzi ne patrí aj už spomenutý komponent *GridView*, ktorý zobrazuje dáta formou tabuľky. Jeho použitie je jednoduché. Stačí mu určiť zdroj dát (v našom prípade jeden z objektov logickej vrstvy) a funkciu, podľa ktorej vyberie a zobrazí príslušné dáta, prípadne zdroj pre parametre tejto funkcie (v prípade parametrizovaného výberu, t.j. výber určený podmienkou). *GridView* umožňuje jednoduché filtrovanie a po implementácii príslušných metód aj editáciu a vymazanie záznamov z databázy.

Jeden fakt ma ale pri práci s *GridView* nepríjemne prekvapil. Zatiaľ čo pre potreby aplikácie nemusia byť vždy zobrazené všetky stĺpce tabuľky (napríklad zobrazenie stĺpca ID by odkrylo časť dát, ktorá má ostať pred užívateľom utajená), *GridView* vždy vyžadoval načítať z databázy všetky stĺpce a až dodatočne umožnil nežiaduce stĺpce skryť. Pokiaľ som však chcela využiť editáciu alebo vymazanie záznamu, ani toto riešenie nestačilo. Pomohlo až ukrytie nežiaducich stĺpcov tabuľky pomocou CSS.

Ďalším z použitých komponentov pre zobrazovanie dát je serverový komponent *Repeater*. Je to veľmi flexibilný nástroj, avšak z toho vyplýva vyššia náročnosť pri používaní. Neobmedzuje sa len na zobrazovanie dát pomocou tabuľky ako napríklad *GridView*. Funkcionalita a vzhľad *Repeateru* závisí od šablony (templatu), ktorý použije pri zobrazení každého prvku, tzv. *RepeaterItem*. Template definuje vývojár podľa vlastných potrieb. Pri návrhu šablony môže použiť ľubovoľné usporiadanie serverových komponentov. Rovnako ako *GridView*, aj *Repeater* musí mať určený objekt a funkciu, podľa ktorej získa požadované dáta a vytvára *RepeaterItemy*. Postupne prechádza všetkými záznamami a na každý aplikuje definovaný template.

Práca s *Repeaterom* je ale sťažaná, pokiaľ dáta nechceme na stránku len zobraziť, ale ďalej s nimi pracovať. Príkladom je zistenie kritérií a kategórií pri ukladaní štandardizačného materiálu. Pre každé kritérium je použitím *Repeatera* vyrobený jeden *MyCriterionListBox*. Ten obsahuje kategórie, do ktorých má byť zaradený daný štandardizačný materiál. Po stlačení tlačítka



"Uložiť" by tieto dáta mali byť uložené do databázy. Problémom ale bolo zistiť, ktoré hodnoty jednotlivých *MyCriterionListBox*ov boli označené. Totiž v čase písania kódu tieto komponenty neexistujú. Nedá sa na ne odvolať ako na ostatné web controly, pretože sa generujú až po spustení aplikácie.

Napokon sme prišli na riešenie. Je potrebné manuálne prechádzať všetkými prvkami *Repeater*u a pre každý jeho *RepeaterItem* zistiť, aké komponenty obsahuje a aké majú hodnoty. Následne môžu byť tieto hodnoty uložené do databázy.

ASP.NET 2.0 poskytuje aj nástroj na správu užívateľov, vytváranie rolí a personalizáciu, tzv. Web Site Administration Tool. ASP.NET taktiež poskytuje komponenty, ktoré umožňujú vytváranie nového používateľa (*CreateUserWizard*), prihlásenie používateľa (*Login*), zistiť stav a prihlasovacie meno práve prihláseného používateľa (*LoginStatus*, *LoginName*) a podobne.

Spomenuté komponenty mali preddefinované vlastnosti, ktoré nevyhovovali nášmu systému. Problematické bolo aj vytváranie rolí a pridelovanie používateľov do pracovných skupín použitím zabudovaného nástroja. Nakoniec som sa rozhodla spraviť správu používateľov sama, bez použitia týchto web komponentov. Tak je zaručené, že bude spĺňať vlastnosti potrebné pre našu aplikáciu.

## 5.2 Architektúra

V kapitole 4.3 som sa zmienila, že prvotný záver bol vyvíjať aplikáciu v duchu 3-vrstvovej architektúry. Dodržať čistú vrstvomú architektúru však nie je jednoduché a nemôžem tvrdiť, že sa nám to podarilo. Problematické je najmä oddelenie prezenčnej a logickej vrstvy, ktoré často splynú do jednej. Tak aj v tejto aplikácii napriek snahe preniesť čo najviac volaní a rozhodovaní na logickú vrstvu, pravdepodobne nie sú jednotlivé vrstvy oddelené úplne. Problém vzniká pri už spomenutom zaradení Code behind, ktorý v niektorých prípadoch zasahuje do dvoch vrstiev. Jedine dátová vrstva je bezpečne

oddelená od ostatných vďaka *DataSetu*, ktorý poskytuje ASP.NET na tento účel. Taktiež časť logickej vrstvy, ktorá zodpovedá za prístup k *DataSetu*, je spoľahlivo oddelená. Táto vrstva obsahuje triedy, z ktorých každá pracuje s jedným *DataTablom* a *TableAdapterom*, reprezentujúcim jednu tabuľku v databáze..

## 6. Požiadavky na software a hardware

Pred stanovením hardwarových a softwarových požiadaviek je vhodné stanoviť účel inštalácie systému. Pokiaľ je cieľom len overenie funkcionality a testovanie, postačuje akékoľvek prostredie, kde sa dá aplikácia sprevádzkovať. Ak by však mala byť aplikácia uvedená do ostrej prevádzky, treba výberu prostredia venovať ďaleko viac pozornosti. Vzhľadom na to, že naša aplikácia je v štádiu vývoja, budeme sa venovať len výberu prostredia pre testovaciu prevádzku a požiadavky na prostredie pre ostrú prevádzku stanovíme až po ukončení implementácie systému.

Najjednoduchším riešením je nájsť webhosting s podporou ASP.NET 2.0, ktorý umožňuje prácu s databázou a upload súborov. Potom stačí jednoducho nakopírovať našu aplikáciu na server poskytujúci webhosting.

V prípade testovania aplikácie na lokálnom počítači je potrebné nainštalovať nasledujúce aplikácie:

- Microsoft IIS (Internet Information Services) s podporou ASP.NET
- Microsoft .NET Framework 2.0
- SQL Server 2005 (Express Edition)

Z toho vyplývajú nasledovné minimálne hardwarové požiadavky:

- 512 MB RAM
- 900 MB voľného miesta na pevnom disku
- procesor Pentium III, 800 MHz

## 7. Možnosti rozšírenia

Už pri návrhu systému sa uvažovalo o jeho vývoji vo viacerých fázach. V tej prvej sa implementovala základná funkcionálna, k plnej funkčnosti systému však ešte chýba niekoľko dôležitých krokov. Pre členov technickej komisie je dôležité mať prehľad, v akej etape sa materiál nachádza a ako dlho bude trvať. V ďalšom vývoji sa treba zamerať predovšetkým na prácu so štandardizačnými materiálmi a ich zaraďovaním do jednotlivých etáp štandardizačného procesu. Taktiež by bolo príjemné, kedy systém upozornil používateľov na blížiaci sa termín odovzdania pripomienok, dátumy hlasovní a podobne.

Niekedy je potrebné priložiť k ŠM dodatočný súbor, prílohu, pracovnú verziu či pripomienky. Aj tieto súbory by mala aplikácia evidovať a umožniť ich priradenie k danému materiálu v danej etape.

Istú potrebu úprav si vyžiada aj správa používateľov. Malo by sa jednať o starostlivejšiu kontrolu prístupu a oprávnení, najmä keď sa v rámci pracovnej skupiny budú môcť vytvárať projektové tímy. Vtedy ostatní používatelia nebudú môcť zasahovať do materiálov projektovému tímu napriek tomu, že budú v jednej pracovnej skupine. Taktiež by sa mali rozšíriť právomoci neprihláseným používateľom. Tí by mali mať prístup k verejným materiálom. V ideálnom prípade by aplikácia mohla pôsobiť ako verejná knižnica voľne dostupných štandardizačných materiálov.

Pred zverejnením aplikácie na internete a jej uvedení do ostrej prevádzky sa však treba zamerať na zvýšené bezpečnostné opatrenia. Medzi najväčšie hrozby patrí Cross Site Scripting, Session Hijack a SQL Injection.

Ďalej by sme mali myslieť na používateľov našej aplikácie. Stránky by sa v budúcnosti mohli upraviť tak, aby boli v čo najväčšom súlade so štandardom WCAG 1.0 . Tento štandard sa venuje prístupnosti webových aplikácií hendikepovaným používateľom. Pod pojmom hendikep sa však nemyslí len fyzický (napr strata sluchu, zraku, ..), ale zahŕňa aj tých

užívateľov, ktorí používajú na prehliadanie webových stránok mobilné telefóny, textové prehliadače, majú pomalé internetové spojenie a podobne.

Rovnako nemôžeme zabudnúť ani na užívateľsky príjemné prostredie, intuitívnu navigáciu a jednoduchú prácu s aplikáciou. Navrhnuť takéto prostredie nie je jednoduché, v tejto oblasti ešte je čo zdokonaľovať.

## 8. Záver

System pre evidenciu a kategorizáciu standardizačných materiálov prešiel prvou fázou vývoja. Podarilo sa implementovať základnú funkcionálnosť, ktorá bola špecifikovaná v zadaní bakalárskej práce. Predpokladáme, že teraz bude naplnený údajmi a uvedený do skúšobnej prevádzky, počas ktorej budeme opravovať odhalené chyby, zbierať pripomienky k implementovaným riešeniam a podnety pre ďalší vývoj systému. Ak sa systém osvedčí, uvažujeme o jeho dopracovaní v rámci diplomovej práce magisterského štúdia informatiky.

## Zoznam použitej literatúry

- [1] Martin Kralovič, Štandardizácia v oblasti informačnej bezpečnosti, 2008, diplomová práca
- [2] International Organisation for Standardisation, How are ISO standards developed  
[http://www.iso.org/iso/standards\\_development/processes\\_and\\_procedures/how\\_are\\_standards\\_developed.htm](http://www.iso.org/iso/standards_development/processes_and_procedures/how_are_standards_developed.htm)
- [3] ISO/IEC JTC 1 Directives, 5th EditionnVersion 3.0, 05. apríl 2007.  
<http://isotc.iso.org/livelink/livelink/fetch/2000/2122/327993/755080/1054033/2541871/JTC001-N-8557.pdf?nodeid=6319110&vernum=0>
- [4] International Organisation for Standardisation, Stage code table  
[http://www.iso.org/iso/standards\\_development/processes\\_and\\_procedures/stages\\_description/stages\\_table.htm](http://www.iso.org/iso/standards_development/processes_and_procedures/stages_description/stages_table.htm)
- [5] Slovenský ústav technickej normalizácie,  
<http://www.sutn.gov.sk/?menu=32>
- [6] World Wide Web Consortium,  
<http://www.w3.org/>
- [7] Microsoft Developer Network, 3-tier Architecture with ASP.NET 2.0,  
<http://msdn.microsoft.com/en-us/library/aa581776.aspx>
- [8] Luboslav Lacko, ASP.NET pre začiatočníkov

# Zoznam príloh

Na priloženom CD sa nachádzajú zdrojové kódy aplikácie, a vzhľadom na nestálosť internetových zdrojov aj použité webové stránky.

- [1] zdrojové kódy bakalárskej práce
- [2] International Organisation for Standardisation, How are ISO standards developed  
[http://www.iso.org/iso/standards\\_development/processes\\_and\\_procedures/how\\_are\\_standards\\_developed.htm](http://www.iso.org/iso/standards_development/processes_and_procedures/how_are_standards_developed.htm)
- [3] ISO/IEC JTC 1 Directives, 5th EditionnVersion 3.0, 05. apríl 2007.  
<http://isotc.iso.org/livelink/livelink/fetch/2000/2122/327993/755080/1054033/2541871/JTC001-N-8557.pdf?nodeid=6319110&vernum=0>
- [4] International Organisation for Standardisation, Stage code table  
[http://www.iso.org/iso/standards\\_development/processes\\_and\\_procedures/stages\\_description/stages\\_table.htm](http://www.iso.org/iso/standards_development/processes_and_procedures/stages_description/stages_table.htm)
- [5] Slovenský ústav technickej normalizácie,  
<http://www.sutn.gov.sk/?menu=32>
- [6] World Wide Web Consortium,  
<http://www.w3.org/>
- [7] Microsoft Developer Network, 3-tier Architecture with ASP.NET 2.0,  
<http://msdn.microsoft.com/en-us/library/aa581776.aspx>



# Príloha A – Špecifikácia bakalárskej práce

## A.1 Úvod

Aby bolo možné v dostatočnom rozsahu popísať účel aplikácie, je potrebné pochopiť dôvody vzniku jednotlivých požiadaviek. Požiadavky na funkčnosť tohto IS vznikli z dvoch nezávislých zámerov.

Prvým zámerom bolo vytvoriť aplikáciu, ktorá by umožňovala evidenciu informácií o veľkom počte dokumentov – štandardizačných materiálov (ďalej aj „ŠM“) vydaných rôznymi vydavateľmi, ich kategorizáciu a modelovanie vzťahov. V ideálnom prípade by aplikácia mala byť schopná v rámci informácie o ŠM uložiť aj jeden alebo viac súborov – znení ŠM.

Druhým zámerom, motivovaným skúsenosťami z pôsobenia v TK 37 SÚTN, bolo vytvorenie aplikácie na sledovanie štandardizačného procesu pre štandardy schvaľované v ISO/IEC JTC 1 (partnerskej komisii TK 37). Pracovné dokumenty JTC 1 sú členom TK 37 sprístupňované prostredníctvom pomerne neprehľadného systému, kde sú prezentované ako zoznam súborov (za posledný rok bolo týchto súborov len pre subkomisiu SK 02 takmer 700). Tento systém neumožňuje jednoduché vyhľadanie súborov súvisiacich s konkrétnym pripravovaným štandardom, a tým sťažuje sledovanie vývoja konkrétnych štandardov, resp. štandardov v určených oblastiach. Cieľom bolo teda vytvoriť systém, ktorý by za podmienky pravidelného dopĺňania informácií a súborov (zamestnancom SÚTN alebo členmi technickej komisie) umožňoval všetkým členom TK 37 jednoduché sledovanie procesu štandardizácie v oblastiach, za ktoré sú zodpovední.

Keďže systémy, ktoré by realizovali uvedené zámery, by sa do veľkej miery prelínali, bol navrhnutý systém integrujúci uvedenú funkcionálnosť. Predpokladá sa, že systém bude implementovaný v rámci záverečných prác študenta FMFI (bakalárska a diplomová práca), preto je pri návrhu požiadaviek zohľadnený aj tento faktor (viď kapitolu A.7 Možnosť vývoja vo viacerých etapách).

Aplikácia by mala umožňovať prístup k údajom prostredníctvom internetového

prehliadača (tenký klient). Vzhľadom na určenie aplikácie nie je potrebné zhotovovať samostatnú aplikáciu na prístup k údajom (tučného klienta).

Aplikácia by mala byť vyvíjaná na platforme Microsoft .NET Framework 2.0 s použitím ASP.NET 2.0, ako databázový server by mal slúžiť Microsoft SQL Server 2005. V prípade potreby je možné použiť novšie verzie, resp. rozšírenia.

## **A.2 Aplikované štandardy a kompatibilita s prehliadačmi**

Aplikácia by mala byť v súlade nasledujúcimi špecifikáciami:

- W3C XHTML 1.0 Strict
- W3C CSS 2.1

Aplikácia by taktiež mala vyhovovať požiadavkám W3C WCAG 1.0 priority 1 do takej miery, ako to umožňuje použitý framework.

Generované stránky by sa mali správne zobrazovať v najrozšírenejších webových prehliadačoch, t.j.:

- Microsoft Internet Explorer 6.0
- Windows Internet Explorer 7.0
- Mozilla Firefox 2.0

## **A.3 Štandardizačný materiál**

Pod ŠM rozumieme akýkoľvek jednotlivý publikovaný štandardizačný dokument (napr. normu), prípadne štandardizačný dokument o ktorom sa predpokladá, že sa stane publikovaným<sup>3</sup> (napr. norma v procese tvorby). Každý ŠM má nejaké základné vlastnosti, napr.:

- Označenie (číslo)
- Názov
- Dátum vydania
- Počet strán

---

<sup>3</sup> Nie je smerodajné, či je publikovaný verejne, alebo v rámci uzatvorenej skupiny používateľov.

- Abstrakt

V čase tvorby tohto dokumentu nie sú identifikované ďalšie vlastnosti ŠM, avšak architektúra aplikácie by mala umožňovať v budúcnosti doplniť alebo zmeniť vlastnosti ŠM tak, aby boli zásahy do kódu boli vykonávané iba na minimálnom počte ucelených úsekov kódu.

Každý ŠM by mal byť zaradený v jednej alebo viacerých kategóriách, pričom kategorizácia by mala byť vykonávaná podľa viacerých kritérií. Podrobnejšie informácie o kategóriách ŠM sú uvedené v kapitole A.4 Kategórie štandardizačných materiálov.

Niektoré ŠM sú s inými ŠM prepojené. Aplikácia by mala umožňovať evidenciu vzájomných vzťahov ŠM. Podrobnejšie informácie sú uvedené v kapitole A.5 Vzťahy medzi štandardizačnými materiálmi.

V závislosti na autorovi ŠM prechádzajú ŠM rôznymi procesmi tvorby a pripomienkovania. Aplikácia by mala podporovať sledovanie tohto procesu. Podrobnejšie sú požiadavky uvedené v kapitole A.6 Podpora procesu vývoja ŠM.

Aplikácia by mala umožňovať vyhľadávanie v ŠM v rozsahu evidovaných atribútov.

## **A.4 Kategórie štandardizačných materiálov**

Veľmi dôležitou funkciou, ktorú má plniť aplikácia na evidenciu ŠM je možnosť kategorizovať ŠM podľa zvolených kritérií. Príkladmi kritérií, podľa ktorých môžu byť ŠM kategorizované sú:

- Podľa vydavateľa:
  - ISO,
  - IEC,
  - ISO/IEC,
  - SÚTN,
  - BSI,
  - DIN,

- ...
- V prípade niektorých vydavateľov podľa komisie zodpovednej za vývoj normy
- Podľa záväznosti ŠM, napr.:
  - EN – európska norma – členské štáty sú povinné ŠM prevziať do 6 mesiacov od vydania,
  - aplikácia normy vyplýva z právnych predpisov SR alebo EÚ,
  - podľa normy je možná certifikácia organizácie/produktu (napr. ISO/IEC 27001),
  - súhrn odporúčaní „best practices“ (napr. ISO/IEC 17799),
- Podľa obsahového zamerania, napr.:
  - riadenie informačnej bezpečnosti,
  - kryptológia,
  - audit,
  - certifikácia systémov (Common Criteria),
  - vývoj systémov,
  - správa systémov,
  - ...,
- Podľa jazyka

System musí byť vyvinutý tak, aby umožňoval prídanie, zmenu alebo odobratie kritéria a kategórie v rámci jedného kritéria. Zmazanie nesmie spôsobiť inkonzistenciu v databáze. V prípade zmeny/vymazania používaného kritéria alebo kategórie musí používateľ na túto skutočnosť byť upozornený pred potvrdením zmazania. Upozornenie musí obsahovať počet ŠM, ktorých sa zmena/vymazanie dotkne.

## **A.5 Vzťahy medzi štandardizačnými materiálmi**

Aplikácia by mala podporovať definíciu vzťahov medzi ŠM. Vzťah ŠM je

popísaný nasledujúcimi atribútmi:

- Kategória vzťahu
- Štandardizačný materiál
- Štandardizačný materiál

Kategórie vzťahov by mali okrem mena, príp. krátkeho popisu definovať aj „orientáciu vzťahu“ so sémantikou ako v teórii grafov, t.j., vzťahy môžu byť dvoch druhov:

- „neorientované“, pri ktorých nezáleží na poradí, v ktorom sú ŠM uvedené (napr. vzťahy „súvisí s“, „koliduje s“),
- „orientované“, pri ktorých záleží na poradí, v ktorom sú ŠM uvedené (napr. vzťah „nahrádza“).

Aplikácia by mala umožňovať definovanie nových kategórií vzťahov, úpravu a zmazanie existujúcich kategórií vzťahov. Zmazanie nesmie spôsobiť inkonzistenciu v databáze, t.j. zmazanie kategórie spôsobí zmazanie všetkých vzťahov zmazanej kategórie. V prípade zmeny/vymazania používanej kategórie vzťahov musí používateľ na túto skutočnosť byť upozornený pred potvrdením zmazania. Upozornenie musí obsahovať počet vzťahov/ŠM, ktorých sa zmena/vymazanie dotkne.

## A.6 Podpora procesu vývoja ŠM

Aplikácia by mala umožňovať zadefinovať rôzne typy ŠM odlišujúce sa rôznymi procesmi ich tvorby. Proces vývoja ŠM je definovaný:

- jednotlivými etapami označenými názvom a voliteľne kódom,
- definíciami možných prechodov medzi etapami,
- definíciou aspoň jednej možnej počiatočnej etapy procesu,
- definíciou aspoň jednej možnej konečnej etapy procesu; nie všetky konečné etapy znamenajú publikovaný štandard (tzn. štandard bol odmietnutý).

Jednotlivé etapy by mali mať okrem názvu mať možnosť doplnenia poznámky, napr. o štandardnej dĺžke trvania etapy. Pri zaradení ŠM do niektorej z etáp by

malo byť možné určiť predpokladaný dátum ukončenia etapy, aby bolo možné sledovať ďalší vývoj štandardizačného procesu. V prípade potreby by malo byť možné dátum ukončenia etapy zmeniť. O každej etape v rámci konkrétneho ŠM by tiež malo byť možné doplniť poznámku o priebehu etapy, ako aj dátumy reálneho začatia a ukončenia.

Pre každú etapu je možné vložiť jeden alebo viac súborov, ktoré s ňou súvisia (napr. pracovné verzie ŠM, prílohy, súhrn pripomienok, vyjadrenia k pripomienkam). Aplikácia by preto mala umožňovať ukladanie ľubovoľných súborov. Pre každý súbor by mali byť evidované niektoré vlastnosti, ako napr.:

- Dátum vytvorenia súboru
- Názov súboru
- Autor
- Popis obsahu súboru
- ...

## **A.7 Možnosť vývoja aplikácie vo viacerých etapách**

Nakoľko sa predpokladá postupný vývoj aplikácie, významnou požiadavkou je možnosť dekomponovať jednotlivé funkcie systému do takých celkov, aby bolo možné aplikáciu využívať aj bez dostupnosti ostatných funkcií. Pri tvorbe architektúry aplikácie je však nutné zohľadniť aj funkcie, ktoré budú implementované aj v budúcnosti.

V prvej etape je potrebné realizovať všetky funkcie uvedené v tomto dokumente s výnimkou kapitoly A.6 Podpora procesu vývoja ŠM.

Následne je možné doplniť funkcie popísané v kapitole A.6 Podpora procesu vývoja ŠM, prípadne niektoré z nasledujúcich doplnkových funkcií:

- fulltextové vyhľadávanie vo vlastnostiach ŠM a obsahu externých súborov;
- N-árne vzťahy, čím by sa umožnilo vytvárať aj akési skupiny ŠM tak, že bude zachovaná konzistencia z pohľadu každého ich člena (napr. rodiny štandardov, ako sú PKCS, ISO/IEC 13335 alebo BS 25999);

- vytváranie vzťahov aj na dokumenty externé vzhľadom k ŠM (napr. právne predpisy, medzinárodné zmluvy, ...);
- e-mailová notifikácia udalostí (napr. zaslanie emailu vlastníkovi ŠM v čase predpokladaného ukončenia etapy).

## A.8 Bezpečnostné požiadavky na systém

### A.8.1 Prístupové práva používateľov

Aplikácia by mala umožňovať riadenie prístupu používateľov, na autentifikáciu je postačujúce použitie hesla.

Vzhľadom na predpokladaný rozsah využitia aplikácie (v rámci technickej komisie SÚTN) sa nepredpokladá využitie veľkým počtom používateľov. To umožňuje ponechať komplexnosť subsystému na riadenie prístupu na relatívne nízkej úrovni. Pri návrhu architektúry systému je však vhodné predpokladať, že v budúcnosti môžu byť požiadavky na riadenie prístupu rozšírené (napr. o vnáranie skupín).

O používateľoch by mali byť evidované základné informácie, ako sú:

- meno,
- organizácia,
- kontaktné údaje (e-mailová adresa, telefónne číslo).

Používatelia s rovnakými požiadavkami na prístupové oprávnenia môžu byť združení v skupine. Skupina je charakterizovaná názvom a popisom. Jeden používateľ nemusí byť zaradený do žiadnej skupiny a môže byť členom ľubovoľného počtu skupín. Skupina nesmie obsahovať inú skupinu.

Pre jednotlivé roly v systéme je možné pridelovať prístupové práva jednotlivým používateľom alebo skupinám, pokiaľ nie je v špecifikácii uvedené inak. V rámci systému by mali byť zadané nasledujúce roly:

- administrátor aplikácie (používateľ s oprávneniami na všetky činnosti, vrátane konfigurácie systému a prehliadanie auditných záznamov),
- správca používateľov (používateľ s oprávneniami na všetky činnosti súvisiace so správou používateľov, ako je ich vytváranie, zmena

atribútov, uzamknutie, ...),

- kontrolný používateľ (používateľ s oprávneniami na prehliadanie auditných záznamov)
- správca obsahu (používateľ s oprávneniami čítať, meniť a rušiť akýkoľvek ŠM, kategóriu, vzťah, ...),
- tvorca obsahu (používateľ s oprávneniami vytvoriť nový ŠM).

Každý ŠM môže mať zadaného jedného vlastníka – osobu zodpovednú za správnosť a aktuálnosť evidovaných informácií. V prípade štandardu v procese tvorby je vlastníka možné považovať aj za projektového manažéra. Vlastníkom ŠM sa pri vytvorení stáva používateľ, ktorý ŠM vytvoril. V prípade potreby môže vlastníctvo ŠM zmeniť aktuálny vlastník ŠM, administrátor aplikácie a správca obsahu.

Jednotlivé ŠM môžu taktiež mať definovaných niekoľkých členov projektového tímu (používateľov alebo skupiny používateľov). Členstvo v projektovom tíme môže určiť vlastník ŠM, administrátor aplikácie a správca obsahu.

Prístupové oprávnenia súvisiace s daným ŠM by malo byť možné nastaviť pre nasledujúce operácie:

- zobrazenie základných informácií o ŠM a vybraných informácií o etapách (dátumy začatia a predpokladaného skončenia prebiehajúcej etapy, dátumy začatia a ukončenia etapy),
- zobrazenie poznámok k etapám,
- zobrazenie súborov priložených ku konečnej etape,
- zobrazenie súborov priložených ku všetkým etapám,
- úprava základných informácií o ŠM,
- úprava informácií aktuálnej etape (dátumy, poznámky),
- vytvorenie novej etapy, úprava informácií o všetkých etapách (dátumy, poznámky),
- priloženie súboru k aktuálnej etape,
- správa súborov priložených ku všetkým etapám.



Prístupové práva na jednotlivé operácie so ŠM by malo byť možné pridelovať nasledujúcim typom používateľov:

1. Vlastník ŠM (VŠM),
2. Členovia projektového tímu (ČPT),
3. Autentifikovaný používateľ (AP),
4. Neautentifikovaný používateľ (NAP).

Každý z uvedených typov zahŕňa aj všetky typy používateľov uvedené s nižším poradovým číslom (napr. typ používateľov *Prihlásení používateľa* zahŕňa aj typy používateľov *Vlastník ŠM* a *Členovia projektového tímu*).

Prístupové práva k ŠM by mať možnosť mali upravovať iba vlastník ŠM, administrátor aplikácie a správca obsahu, a to pre každú operáciu jednotlivo. Aby bolo riadenie prístupu zjednodušené, aplikácia by mala umožňovať vytvorenie štandardných profilov pre prístup k ŠM, z ktorých bude možné jednoducho zvoliť. Ako príklad možných prístupových profilom môže poslúžiť nasledujúca tabuľka.

Operácia so ŠM	Prístupový profil		
	ŠM – vývoj 1	ŠM – vývoj 2	ŠM uzamknutý verejný
Zobrazenie základných informácií o ŠM a vybraných informácií o etapách	NAP	AP	NAP
Zobrazenie poznámok k etapám	ČPT	ČPT	ČPT
Zobrazenie súborov priložených ku konečnej etape	ČPT	ČPT	NAP
Zobrazenie súborov priložených ku všetkým etapám	ČPT	ČPT	ČPT
Úprava základných informácií o ŠM	ČPT	VŠM	VŠM
Úprava informácií aktuálnej etape (dátumy, poznámky)	ČPT	VŠM	VŠM
Vytvorenie novej etapy, úprava informácií o všetkých etapách (dátumy, poznámky),	VŠM	VŠM	VŠM
Priloženie súboru k aktuálnej etape,	ČPT	ČPT	VŠM
Správa súborov priložených ku všetkým etapám.	VŠM	VŠM	VŠM

Rozhranie aplikácie by nemalo zobrazovať ovládacie prvky, na ktorých použitie

používateľ nemá dostatočné oprávnenia. V prípade, že sa používateľ aj napriek tomu pokúsi vykonať nepovolenú operáciu (napr. zadaním URL), operácia nesmie byť vykonaná, systém by mal ponúknuť možnosť prihlásiť sa pod používateľským účtom.

### **A.8.2 Vytváranie auditných záznamov**

Aplikácia by mala mať možnosť ukladať záznamy o jej využívaní. Jedná sa o nasledujúce udalosti:

- úspešné a neúspešné prihlásenie používateľa,
- vytvorenie, zmena alebo zmazanie objektu (ŠM, kategória, používateľ, ...),
- zmena vlastníka ŠM,
- zmena prístupových práv k ŠM,
- pokus o vykonanie nepovolenej operácie,
- zmena konfigurácie prostredníctvom rozhrania aplikácie.

Pri všetkých udalostiach by mali byť zaznamenané všetky relevantné údaje potrebné pre prípadnú rekonštrukciu udalostí, minimálne však:

- Poradové číslo udalosti
- Dátum a čas
- IP klienta
- ID používateľa (ak je k dispozícii)
- Typ udalosti, popis
- ID súvisiacich objektov (ŠM, používateľ, pôvodný a nový vlastník ŠM...)

Prostredie by malo umožňovať jednoduché vyhľadávanie v auditných záznamoch, a to podľa všetkých relevantných atribútov.

Zmena alebo zmazanie auditných záznamov by nemali byť povolené ani administrátorom systému.

### **A.8.3 Bezpečnostné požiadavky webovej aplikácie**

Vzhľadom na známe hrozby pôsobiace na webové aplikácie je v systéme potrebné zabezpečiť implementáciu dodatočných bezpečnostných prvkov zaisťujúcich ochranu pred týmito hrozbami. Jedná sa najmä o ochranu pred útokmi cross-site scripting, session hijack a SQL injection.