

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky a mezioborových inženýrských studií

Studijní program: B2612 – Elektrotechnika a informatika

Studijní obor: 2612R011 – Elektronické informační a řídicí systémy

Internetový portál pro seniory

Internet portal for seniors

Bakalářská práce

Autor: **Martin Lujc**

Vedoucí BP/DP práce: Ing. Martin Vlasák

Konzultant: Ing. Zuzana Capeková

V Liberci 17. 5. 2007

Poděkování

Děkuji paní Věře Rohanové za téma mé bakalářské práce. Dále děkuji RNDr. Kláře Císařové za umožnění realizace zadání práce a spolupráci při zajištění konzultantů a souvisejících administrativních činností.

Za konzultace k mé bakalářské práci děkuji vedoucímu bakalářské práce Ing. Martinu Vlasákovi, a to především za pomoc a spolupráci při zpracování programové stránky mé práce. Děkuji i Ing. Zuzaně Capekové a Ing. Romanu Špánkovi za konzultace a pomoc při tvorbě databáze pro tuto práci. Dále děkuji i všem ostatním, kteří mě při této práci jakkoliv podporovali.

Abstrakt

Předmětem této práce je návrh internetového portálu pro seniory. Tento portál by měl řešit jejich rozsáhlou zábavní a komunikační činnost. Součástí systému je diskuse, vkládané zájmové články, dále i informace o akcích a jiných činnostech pro seniory.

Systém je členěn na dvě části - na redakční a na publikační. Obě části bylo potřeba zpracovat s ohledem na jednoduchost a systematičnost, aby mohly být používány touto věkovou skupinou. Projekt je zpracováván pro zadavatele za účelem průzkumu mezi seniory pomocí informačních technologií. Databáze pro tuto aplikaci byla navržena pomocí databázového systému SQL a samotný návrh relační databáze byl proveden pomocí volně šiřitelného softwaru DBDesigner. Celá aplikace obsluhující databázi byla napsána v jazyku PHP a doplněna GUI (Grafické uživatelské rozhraní) napsaným jazykem HTML a upraveného pomocí CSS. Systém běží na http serveru Apache. Dále bylo potřeba zajistit plynulé měnění vzhledu a to bylo zajištěno vybraným šablonovacím systémem HTML Template.

Abstract

The purpose of this bachelor work is to design internet portal for seniors. Portal should solve their broad entertaining and communication activities. Discussion, feeding interest articles, information about actions and other activities for seniors. System is divided into two parts – editorial part and publication part. Both these parts I created with respect for simplicity and systematization to be used simply by this age group. This bachelor work as a project was worked on for a customer in order to making research among seniors (with use of IT). The database for this application was designed by freeware DBDesigner. The whole application operating the database was programmed in PHP programming language with support of GUI (graphics users interface) written in HTML language and adapted with help CSS. System runs on http server Apache. Next thing need to assure were fluent changes of visual aspects. These fluent changes were ensured by system HTML Template.

Obsah

1.	Úvod.....	7
3.	Senioři a informační technologie	8
3.	Použité a zamítnuté technologie	10
3.1	Servery.....	10
3.2	Databázové systémy	12
3.3	Programovací jazyky	13
3.4	Šablonovací systémy	15
3.5	Použité programy na návrh systému.....	16
4.	Návrh databáze	17
4.1	Popis databáze v kostce a její účel a požadavky	17
4.2	Popis relací a entit	19
5.	Popis adresářové struktury portálu,popis skriptů a důležitých funkcí	29
5.1	Popis adresářové struktury.....	29
5.2	Rozdělení skriptů a popis jejich funkce.....	30
5.4	Popis důležitých funkcí pro chod portálu.....	32
6.	Popis tvorby redakčního systému	33
6.1	Proč nebyl použit nějaký stávající systém	33
6.2	Členění systému podle zadání	33
6.3	Popis jednotlivých částí systému	34
7.	Problematika vzhledů na portále.....	39
7.1	Rozložení vzhledů na portále	39
8.	Stručný popis funkce portálu.....	40
9.	Kroky nutné pro instalaci systému na doménu	41
10.	Závěr.....	42
11.	Přílohy	43
11.1	Popis funkce HTML TEMPLATE	43
11.2	Základní popis modelování v Db designeru	46
11.2	Celkový erd diagram relační databáze.....	48

1. Úvod

Tato bakalářská práce pojednává jednak o programování na internetu a všeobecných problémech vývoje aplikací, ale také především o nám cizí a často opomíjené starší generaci a jejich problémech v přístupu k informačním technologiím. Jak víme, na internetu disponujeme širokou škálou různých internetových portálů, chatů a různých stránek, které se zabývají širokým spektrem témat. Jak se vyvíjí internet i všeobecně informační technologie, vyvíjí se i přístup k nim. K jeho využívání se tím dostávají i senioři nebo spíše pro upřesnění budoucí senioři. Tato generace už se setkává s informačními technologiemi v zaměstnání a i mimo něj a bez problémů je umí používat. Tím mizí bariéra, která byla kdysi běžná, a to fenomén používání internetu mladší či produktivní generací. Takto vznikl projekt na zřízení portálu pro seniory. Hlavním problémem v této oblasti pořád zůstává to, že senioři na internetu nenacházejí co by nacházet chtěli. Chybí jim na internetu jakýsi koutek, kde by našli přesně to, co se týká jejich problematiky. Dle výzkumů se uvádí, že pokud by našli co hledají, většina z nich by takový koutek uvítala.

Mým úkolem bylo za použití stávajících technologií vytvořit přitažlivý portál pro seniory, který by zajišťoval jejich zábavu, vzdělávání a veškerou podporu pro jejich bohatší život a informovanost. Systém musel být dostatečně jednoduchý a hlavně intuitivní, což nás nakonec dovedlo k jedinému řešení - bylo potřeba vyvinout aplikaci přímo na míru potřebám. Jako na jakýkoli portál byl na ní kladen velký důraz z hlediska bezpečnosti a systematičnosti. Každý portál nám mění vzhled před očima, tudíž i tento trend musel být do práce zahrnut, což nás přivedlo i k použití šablonovacích systémů. Nakonec bylo potřeba všechny technologie odpovídajícím způsobem propojit, aby vznikl přijatelný a stabilní celek.

2. Senioři a informační technologie

Stárnutí

Stáří a stárnutí je přirozenou etapou člověka v lidské společnosti. Současný celosvětový trend stárnutí populace si žádá rozsáhlé změny v přístupu nejen v oblastech jako je např. zdravotnictví nebo penzijní politika, ale také v celkovém pojetí vnímání generace seniorů a jejich aktivnímu zapojení do společnosti.

V České republice byl v roce 2001 podíl osob ve věku 65+ (tzn. v seniorském věku) celkem 13,8%. Demografická prognóza říká, že v roce 2030 to bude přibližně 25%. (Zdroj: Rabušic, 2002).

Jak bude tato generace seniorů vypadat? Bude se od té dnešní lišit? Odpovědi lze definovat z různých pohledů. Jedním z nejzásadnějších rozdílů je jistě pohled ze strany využívání moderních komunikačních technologií, konkrétně internetu.

Již dnes je vidět značně se zvyšující podíl obyvatelstva k internetu: od roku 2000 do roku 2004 se zvýšil z 18% na 35%, přičemž největší nárůst je vidět u věkové kategorie 55+, kde je nárůst téměř čtyřnásobný. (Zdroj: GFK, 2005)

Přesto je dnešní generace seniorů u nás ve srovnání se zbytkem vyspělého světa (především USA a UK) velmi pozadu. Důvodů je několik – např. nízká dostupnost internetu (popř. PC vůbec) a psychický blok, který vyplývá z obavy učit se něčemu novému, co se pro ně objevilo již v nevhodnou dobu – v dobu, kdy už nebyli pracovně aktivní. Jde defacto o sníženou schopnost fungovat v rychle se měnící informační společnosti.

„Nové“ seniory“ lze nejlépe představit takto:

"Senioři blízké budoucnosti (kolem roku 2015) budou skupinou velmi heterogenní, kdy vedle sebe budou žít na jedné straně senioři narození v období 1930-1945, jimž bude mezi sedmdesáti až osmdesáti pěti lety, a vedle nich senioři narození v období 1950-1955, to je tzv. "mladí staří". A jelikož podle základní premisy sociologie věku platí, že díky rozdílným historickým podmínkám, jež jsou dány rozdílnými politickými, ekonomickými a sociálními strukturami, různé věkové kohorty (sociálně) stárnou různým způsobem, je zřejmé že v dynamicky se proměňujících moderních společnostech musí být zákonitě jednotlivé kohorty seniorů od sebe poměrně značně odlišné. Kohorta 1950-1955 je již dnes skupinou nebývale vzdělanou, velmi aktivní, zvyklou na individuální nezávislost, v relativně dobrém zdravotním stavu a s relativně slušnou životní úrovní. Bude navíc první generací seniorů, která bude ve

velké míře počítačově gramotná, což ve světě po roce 2015 nebude nedůležitý prvek." (Rabušic, 2002, str. 17)

Co lze z rozdílů současné a „nové“ generace seniorů vyvodit?

Relativně nízké využívání internetu současnou generací neznamena, že by pro nás senioři měli být na okraji zájmu (myšleno v aktivitách pro ně vyvíjených). Naopak. Je důležité je dostatečně motivovat k tomu, aby přešli z pasivního trávení volného času k aktivitám, které je budou určitým způsobem rozvíjet. Internet je jedním z velmi dobrých nástrojů. Pokud na něm najdou to, co je pro ně v této fázi života potřebné, co je opravdu zajímavá, co je pobaví a co jim především umožní spolu aktivně komunikovat, lze očekávat, že budou v jeho využívání velmi aktivní.

Vzhledem k tomu, že v současné době neexistuje opravdový interaktivní portál pro seniory, který by reflektoval uvedené potřeby, je cílem ho vytvořit. Zvyšující se zájem o něj v blízké budoucnosti lze volně predikovat v sumarizaci již uvedených odhadů vývoje, a to:

"S nástupem mladší generace seniorů, lze do budoucna očekávat, že internet bude sehrávat v životě starších lidí významnější roli." (Vohralíková, 2004, s. 77)

Portál a jeho záměr

Velmi podstatným faktorem životaschopnosti portálu je splnění kritických faktorů úspěchu, což představuje ty trendy a atributy portálu, které musí projekt naplnit, aby úspěchu dosáhl. Na základě provedených trendových analýz a semikvalitativního průzkumu jsou určeny tyto kritické faktory úspěchu:

- 1. Profesionální vzhled a struktura.*
- 2. Komplexnost informací.*
- 3. Dobrá orientace, rychlost.*
- 4. Interaktivita.*

Závěr

Studie proveditelnosti speciálního portálu pro seniory v oblasti socio-demografické i marketingové podporují myšlenku vytvořit speciální portál pro seniory za předpokladu nepodcenění kritických faktorů úspěšnosti.

3. Použité či zamítnuté technologie

3.1 Servery

Apache

Je webový server s otevřeným kódem pro různé platformy (Linux, BSD, Microsoft Windows). V současné době dodává prohlížečům na celém světě většinu internetových stránek. Je velice populární, což potvrzují i průzkumy, podle kterých od dubna 1996 je Apache nejpopulárnější server na internetu. V květnu 1999 běží na 57 % všech serverů a v listopadu 2005 jeho používanost dosáhla 69 % a jeho podíl podle průzkumů neustále roste.

Stručná historie:

Vývoj Apache začal v roce 1993 v NCSA (National Center for Supercomputing Applications) na Illinoiské univerzitě pod původním jménem projektu NCSA HTTPd. Vývoj pokračoval až do roku 1998, kdy se zastavil. Důvodem bylo nejprve vystoupení hlavního programátora jménem Rob McCool z projektu, čímž došlo ke zpomalení vývoje a poté k úplnému zastavení.

Důvodem znovuoobnovení byli samotní správci serverům, kteří začali dodávat patche a vývoj pokračoval. Nakonec byl kód úplně přepsán společností Apache Group, která dnes vede vývojářský tým.

Název vznikl z anglického slovního spojení „A patchy server“ (záplatovaný server). Jako indiánský symbol je ve znaku ptačí pero.

Firebird

Historie

Firebird je odvozen ze zdrojového kódu Borland InterBase 6.0. Je plně open source, nerozlišuje tudíž použití na soukromé nebo komerční účely. Technologie Firebirdu je používána již 20 let, což z ní dělá velmi zralý a stabilní produkt.

Hlavní rysy a součásti Firebirdu

- *Plná podpora uložených procedur a spouští.*
- *Plná podpora ACID transakcí.*
- *Referenční integrita.*
- *Multigenerační architektura.*
- *Velmi malé nároky.*
- *Plná podpora interního jazyka pro uložené procedury a spouště (PSQL).*
- *Podpora externích funkcí (UDF knihovny).*
- *Téměř žádná nebo žádná potřeba specializovaných administrátorů.*
- *Skoro žádná konfigurace není potřeba - jen nainstalujte a používáte!*
- *Velká komunita a mnoho míst, kde můžete zadarmo dostat dobré rady.*
- *Volitelná jednosouborová zabudovaná (embedded) verze - vynikající pro katalogy na CDROMech, jednouživatelské nebo testovací verze programů.*
- *Množství nástrojů třetích stran, obsahujících grafické administrační nástroje, nástroje pro replikaci atd.*
- *Obezřetné zápisy - rychlé obnovení, bez potřeby transakčních logů!*
- *Velké množství způsobů jak přistupovat k databázi: nativní/API, dbExpress ovladače, ODBC, OLEDB, .NET provider, JDBC nativní typ 4 ovladač, Python modul, PHP, Perl atd.*
- *Přímá podpora pro všechny hlavní operační systémy, zahrnující Windows, Linux, Solaris, MacOS.*
- *Inkrementální zálohy.*
- *Dostupné 64bit sestavení (buildy).*
- *Úplná implementace kurzorů v PSQL.*

Firebird je dodáván s plnou výbavou řádkových utilit, které umožňují vytvářet

databáze, zobrazovat jejich statistiky, spouštět SQL příkazy a skripty, zálohovat a obnovovat databáze, atd. Máte-li raději GUI (Graphical User Interface) nástroje, máte možnost si vybrat z nepřeberného množství, včetně nástrojů zdarma.

Rozhodování a zdůvodnění

Pokud jde o rozhodování bylo jasné. Firebird se hodí na aplikace klient server v podobě shromažďování dat v podnicích a nebo jak je popsáno v charakterizaci třeba na nějaká výuková cd nebo na podobné aplikace vyžadující databázi. Apache je klasický http server a hodí se na tuto aplikaci přímo na míru. Je známý, podporovaný a spolehlivý, ideální pro aplikaci na tento projekt.

3.2 Samotné databázové systémy založené na jazyku SQL

Na výběry připadali v úvahu dvě možnosti. Každá z nich byla vybrána na základě předešlých rozhodnutí. Důvod byl jasný celý výběr bylo nutné plynule navázat.

MySQL

Jedná se o databázový systém, vytvořený švédskou firmou Mysql AB. Jeho hlavními autory jsou Michael „Monty“ Widenius a David Axmark. Je k dispozici jak pod bezplatnou licenci GPL, tak pod komerční placenou licenci.

MySQL je multiplatformní databáze. Komunikace s ní probíhá – jak už název napovídá – pomocí jazyka SQL. Pro svou snadnou implementovatelnost (lze jej instalovat na Linux, MS Windows, ale i další operační systémy), výkon a především díky tomu, že se jedná o volně šiřitelný software, má vysoký podíl na v současné době používaných databázích. MySQL bylo od počátku optimalizováno především na rychlost, a to i za cenu některých zjednodušení - má jen jednoduché způsoby zálohování a až donedávna nepodporovalo pohledy, trigery, a uložené procedury. Tyto vlastnosti jsou doplňovány teprve v posledních letech, kdy začaly nejčastějším uživatelům produktu – programátorům webových stránek – již poněkud scházet. Nejaktuálnější verzí je verze 5.30 a vyvíjí se dál.

PostgreSQL

PostgreSQL je plnohodnotným relačním databázovým systémem s otevřeným zdrojovým kódem. Má za sebou více než patnáct let aktivního vývoje a má vynikající pověst pro svou spolehlivost a bezpečnost. Běží na všech rozšířených operačních systémech včetně Linuxu, UNIXů (AIX, BSD, HP-UX, SGI-IRIX, Mac OS X, Solaris, Tru64) a Windows. Stoprocentně splňuje podmínky acid, plně podporuje cizí klíče, operace JOIN, pohledy, spouště a uložené procedury. Obsahuje většinu SQL92 a SQL99 datových typů, např. INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL a TIMESTAMP. K

systemu existuje kvalitní volně dostupná dokumentace, včetně českých překladů FAQ a FAQ pro o.s. fy. Microsoft.

PostgreSQL je šířen pod BSD licenci, která je nejliberálnější ze všech open source licencí. Tato licence umožňuje neomezené používání, modifikaci a distribuci PostgreSQL. PostgreSQL můžete šířit se zdrojovými kódy nebo bez nich, zdarma nebo komerčně.

PostgreSQL umožňuje běh uložených procedur napsaných v několika programovacích jazycích, v Perlu, v Pythonu, v jazyku C nebo v speciálním PL/pgSQL, jazyku vycházejícím z PL/SQL fy. Oracle. Existují PostgreSQL varianty JDBC, ODBC, dbExpress, Open Office, PHP, .NET Perl nativních rozhraní. K PostgreSQL existuje překladač Embedded SQL pro C a C++.

Předností systému PostgreSQL je rozšiřitelnost. Systém může být bezproblémově rozšiřován o nové datové typy, funkce operátory, agregační funkce, procedurální jazyky. Díky tomu mohla vzniknout následující rozšíření: PostGIS - podpora pro geografické informační systémy, Tsearch2- podpora fulltextového vyhledávání, Slony-I - master to multiple slaves replikace.

Rozhodnutí

Po srovnání a prostudování jsem pro projekt vybral databázový systém MySQL. Důvodů bylo hned několik. Hlavním byl ten, že zažitá podpora na webových serverech vás víceméně vybízí k jeho použití. Dalším důvodem byla snadná integrace s jazykem PHP, který jsem si vybral pro tvorbu serveru viz níže a také fakt, že s ohledem na server, kde je obvykle nejvíce používána každému známá a zažitá kombinace Apache, MySQL, PHP, která dnes tvoří jakýsi nepsaný standard a je nejlépe široce podporována na celé řadě placených hostingů a nejinak je tomu i na těch oblíbenějších, které jsou poskytovány zdarma.

3.3 Programovací jazyky

V této části padla volba na dvě možnosti. Nejprve budou stručně popsány jejich charakteristiky a historie, popřípadě tvůrci, poté budou srovnány a bude odůvodněn výběr. Na výběr byly zvoleny dvě platformy pro vývoj - ASP.NET a PHP.

ASP.NET

Tato technologie vychází z technologie ASP od firmy Microsoft. Její nová verze, která byla vyvinuta z důvodu nepostačujících možností, se skrývá pod příponou .NET, která označuje rozsáhlý framework. Pro programátora jde hlavně o rozsáhlý objektový model, přes který se spojujete s databázemi, pracujete se soubory, s cookies, se sessions, grafikou atd. Technologie .NET se však nestará jenom o objekty pro programátory, ale také i o správu verzí, řízení chyb, programování v různých jazycích apod.

V rámci ASP.NET můžete standardně pracovat s programovacími jazyky, jakými je

třeba Visual Basic .NET nebo C#.NET. Tato možnost byla už u straší verze ASP, na rozdíl od technologie PHP, které vás váže k jazyku jednomu. Pozoruhodnou vlastností této technologie je, že VB.NET v ASP.NET se od VB.NET, ve kterém programujete se Visual Studiu neliší. Pracujete tak se stejným jazykem a se společným běhovým prostředím, kterým je .NET Framework. Znamená to, že pokud se naučíte jeden jazyk z prostředí Visual Studia, můžete okamžitě začít pracovat nejen na vývoji klasických okenních aplikací, ale i na webových aplikacích, aplikacích pro mobilní telefony nebo PDA, na poskytování webových služeb apod.

PHP

PHP je rekurzivní zkratkou pro PHP: Hypertext Preprocessor (kde původní význam zkratky PHP je Per-sonal Home Page). Jedná se o Open Source skriptovací technologii, jejíž základy položil Rasmus Lerdorf v roce 1995, kdy pro své potřeby sepsal několik Perlovských skrip-tů, které nazval Personal Home Page Set. Jak rostla potřeba pro nové a nové funkce, PHP/FI (jak se technologie až do verze 2.0 jmenovala) se rozrůstalo, bylo přepsáno do jazyka C, ale ještě v roce 1997 bylo jen velkým projektem jednoho člověka. Podle odhadů bylo PHP v té době nainstalováno na 50 000 doménách, což představovalo asi jedno procento všech domén Internetu.

PHP velmi podobné tomu, které známe dnes, vzniklo ke konci roku 1997 jako kompletní přepsání PHP/FI 2.0 pány Andi Gutmansem a Zeevem Suraskim. Rasmus Lerdorf se k této iniciativě přidal, v polovině roku 1998 vzniklo PHP 3.0 a vývoj PHP/FI byl definitivně ukončen. Hlavními přínosy verze 3 byla dobrá infrastruktura pro mnoho databází a různých API, rozšiřitelnost a výkonnější a konzistentnější syntaxe s podporou OOP (objektově orientované programování). Za své největší slávy bylo PHP 3.0 nainstalováno na stovkách tisíc webových serverů, což podle odhadů činilo 10% webových serverů Internetu.

PHP 3.0 bylo už poměrně dobrým nástrojem pro mnoho typových situací, nebylo ale navrženo tak, aby komplexní situace zvládalo efektivně. To bylo hlavní motivací pro PHP 4.0, jehož základem bylo nové jádro napsané Andi Gutmansem a Zeevem Suraskim. Toto jádro nese název, který byl inspirován jejich křestními jmény – Zend Engine. Kromě zvýšené výkonnosti přineslo PHP 4 několik dalších podstatných změn, jako např. HTTP sessions, podporu pro mnoho web serverů apod. PHP 4, současná nejrozšířenější verze, je nainstalována na několika milionech web serverů, odhadem na 20% doménách internetu.

Nejnovějším přírůstkem je PHP 5.0, které bylo vydáno 13. července 2004. Opět přineslo několik vylepšení, ale jedno je mimořádného významu. PHP 5 výrazně zlepšuje objektově orientovanou syntaxi. Přibyly výjimky, konstruktory, destruktory a mnoho dalších jazykových rozšíření. Mezi další podstatná rozšíření patří např. přepracovaná podpora XML, podpora COMu a technologie .NET, lepší podpora MySQL.

Srovnání a zdůvodnění

Nejprve bych chtěl naznačit to, že moje volba byla od začátku jasná a padla na skriptovací jazyk php. Nejen proto, že na vyvinutí této aplikace je zcela vhodný, ale hlavně kvůli jeho rozsáhlé podpoře a dalším nárokům a dále kvůli problémům týkajícím se

optimalizace SEO(optimalizace pro vyhledávače).Podle dostupných informací je kód generovaný asp.net automatický a s vyhledávači má problémy. Dalším faktem bylo, že platforma je podporována jen na placených serverech, což u komerčního projektu nepředstavuje problém, ale při vývoji a zkoušení to obtěžuje! Dalším faktorem byl čas. Jeho využití bylo pro projekt určující a ASP.net je mnohem vzdálenější a mě cizejší než klasické skripty v php.

3.4 Šablonovací systémy

Nejprve začneme tím co vlastně šablonovací systémy jsou. Když se začaly zavádět vedla k tomu programátory jejich touha oddělit kód od vizuálních tagů, např .oddělit php od html a css. Tento nápad vznikl z důvodu, jež je zřejmý - bylo potřeba oddělit práci HTML kodéra od práce programátora. V programátorské praxi je důvod zřejmý - ne každý programátor je dobrý grafik a opačně je pravda stejná. Ale abychom se nevyjadřovali jen o této stránce. Je také pravda, že pokud naprogramujete například nějaký portál, je pro vás mnohem jednodušší díky této technologii měnit jeho vzhled, nežli pokud jí nepoužijete. HTML kód popřípadě jiný není zatížen php a mizí tím všemi oblíbený příkaz echo a autor vzhledu, či člověk znalí pouze HTML a css či XML, může bez zásahu do kódu, či jeho luštění bez jakýchkoliv problémů vzhled změnit. Dále odpadá velice nepříjemný problém - když grafik zasahuje do kódu může program poškodit, ať vědomě, či nevědomě, php zasahuje do kódu html a opačně. Pokud oddělíme PHP a HTML problém nám mizí a nehledě na to další výhoda je zřejmá - pokud se programátor a Html kodér šikovně domluví, mohou pracovat oba současně a projekt poté pouze spojit.

Nabídka šablonovacích systémů:

V této části podotkněme, že nabídka samozřejmě nebyla celá. Zaměřili jsme se na ty, které jsme podle diskusí považovaly za nejznámější a z nich pak jeden byl zvolen a vyzkoušen na projekt.

Z nabídky byli vybrány tři. Jednalo se o Tenk - šablonovací systém vyvinutý nejznámějším českým internetovým portálem seznam, o kterém jsem zhlédl pěkné video přednášky o šablonovacím systému od seznamu na ČVUT Praha, video se dá najít na jejich stránkách. Dalším byl šablonovací systém HTML template a systém, o kterém se toho dá mnoho najít na internetu, s příznačným názvem Smarty.

Stručný popis šablonovacích systémů:

1. Teng

Teng je univerzální šablonovací nástroj. Nyní je implementováno rozhraní pro skriptovací jazyky PHP, Python a samozřejmě i pro C++, ve kterém je celý šablonovací nástroj napsán.

Teng je navrhován s maximálním ohledem na výkon, ovšem i přes to umožňuje pokročilou práci se šablonami (funkce pro práci s řetězci, podmíněné bloky, vlastní proměnné, použití slovníků apod.) Zpracování šablon pomocí Tengu zabírá minimum procesorového času, takže není problém vytvářet dynamické aplikace, které musí zvládat velkou zátěž. Teng je rychlý, díky inteligentní práci se šablonami, kdy se všechny šablony načtou do paměti v podobě bite kódu a poté se s nimi pracuje.

O způsobu práce a funkčnosti Tengu by se dalo napsat hodně, ale to je na příliš mnoho stran. Pro širší použití Tengu by bylo skvělé, kdyby nějaký významný webhoster Teng nainstaloval na své stroje a začal ho nabízet zákazníkům. Myslím, že zanedlouho by to mohla být dobrá konkurenční výhoda.

2. Smarty

SMARTY podle označení znamená chytrý, elegantní, pohotový. Tento systém má sice mnoho vlastností, které z něj tvoří jeden z nejlepších systémů šablon v současné době, nelze o něm ale říci, že se jedná o univerzálně nejlepší systém. Například rychlost není zrovna tím, v čem by SMARTY vynikal. Při výběru nejvhodnějšího systému pro jakýkoli projekt je však nutné brát v úvahu nejen rychlost, ale i funkčnost, rozšiřitelnost, komplexnost, dokumentaci, velikost vytvářeného projektu a další vlastnosti, v nichž SMARTY vyniká.

V čem se tedy SMARTY od ostatních liší? Především je to způsob, jakým jsou šablony zpracovávány. Při prvním volání (tzn. při prvním spuštění skriptu, který pro svůj výstup šablonu používá) jsou převedeny (zkompileovány) do podoby PHP skriptu, který je následně spuštěn a výsledek odeslán prohlížeči. Při dalším volání je pak spuštěna jen zkompileovaná verze šablony. Využití kompilace výrazně zrychluje odezvu serveru při opakovaném volání skriptů. Kromě toho má SMARTY také vestavěnou podporu cache paměti, což při použití PHP akceleratorů umožňuje dosáhnout velmi zajímavých výsledků.

3. HTML TEMPLATE

Tento systém se ze všech tří šablonovacích systémů líbil nejvíc i když měl pro někoho velkou nevýhodu, a to anglicky psaný manuál. Tato skutečnost víceméně, jak musím přiznat, zpočátku nedobrého angličtináře odrazovala. Jeho největší výhoda spočívá v tom, že ze všech tří obsahuje úplně nejméně náročnou logiku a co se týče implementace - pro člověka, který neví nic o programování je nejvhodnější. Tento šablonovací systém a jeho implementace jsou probrány v příloze.

3.5 Použité programy pro tvorbu:

Na tvorbu byly použity čistě Freewareové nástroje. Na prostředí simulující sever a jeho podporu výborný balík EASY PHP, zahrnující server Apache, Mysql a php v jednom. Po jeho instalaci už není nutná jakákoli konfigurace. V tomto balíčku je i oblíbený software

phpMyadmin pro správu entit databáze na serveru. Na psaní kódu jako takového byl použit editor PSpad. Pro úpravu grafiky též freewarový grafický editor GIMP, pro návrh databáze též freewarový software jménem DB designer. Tento program slouží na návrh ERD diagramů relační databáze a práce s ním bude vysvětlena v příloze.

4. Návrh databáze

Popis databáze v kostce a její účel a požadavky

Prvním krokem, kterým vlastní práce začala, bylo posezení nad tím co se od databáze očekává a jak si představujeme její jednotlivé části. Vzhledem k tomu, že se jedná o klasický model relační databáze, zamýšleli jsme se nad datovými toky probíhajícími v této databázi a rozčlenili jsme si ji na jednotlivé části dle zadání. Databázi jsme rozdělili na 8 základních částí.

1. Přístupy a registrace
2. Kategorie týkající se vkládaných článků
3. Diskuse
4. Seznamka
5. Horoskopy
6. Odkazy
7. Akce
8. Deníčky

Přístupy a registrace

Nejprve je potřeba zajistit jednotlivým osobám možnost založit si osobní účet na portálu, přidělit jim příslušné jméno a heslo, popřípadě další údaje pro pohodlné přihlášení na portál. Dále je třeba uchovat či prezentovat jejich osobní data, fotografie v případě seznamky a další náležitosti, které patří k jejich osobnímu účtu na portálu.

Kategorie týkající se vkládaných článků

Tato část má obsahovat veškeré náležitosti připadající článkům, kdo je vložil, kdy a čeho se týkají. Články mají být děleny dle dohody na nadpis, abstrakt (krátký úryvek, o čem článek je) a odstavec. Ke každému odstavci by měli jít přidělit 2 obrázky umístěné dle normativu z důvodu lepšího prezentování na serveru. Dále má být pomocí hledání přes kategorii přístupů jít přidělit libovolný počet autorů k danému textu v žádaném pořadí. Tato

část samozřejmě sděluje danému článku kam patří, do jaké sekce či podsekce.

Diskuse

Má se jednat o klasickou diskusi. Každý příspěvek obsahuje nadpis, text a zařazení. Přispívat mohou jen registrovaní uživatelé, popřípadě pokud se má jednat o příspěvek odborníka, má být odlišen od ostatních libovolným způsobem.

Seznamka

Její obsahem má být sekce, bydliště, fotografie, je-li požadována, znamení a koníček. Vkládat mohou pouze registrovaní uživatelé a inzeráty musejí jít schválit v případě, že by se uživatel chtěl pokusit vložit nepovolený obsah.

Horoskopy

Jedná se o klasické vkládané horoskopy. Obsahem má být abstrakt, přidělené znamení a text vlastního horoskopu.

Odkazy

V této části se jedná o vkládané odkazy. Mají být rozděleny do sekcí a poté každý odkaz dostane přidělenou malou ikonku, popis a link (webová adresa firmy).

Akce

Jedná se o zjištěné akce dle kalendáře organizované pro tuto cílovou skupinu a obsahují datum konání a popis akce.

Deníčky

Ve své podstatě by se mělo jednat o fotoalbum s popisky. Mělo by sloužit k pohodlnému sdílení fotek a mělo by k nim jít přidat název alba a jejich krátký popis a ke každé fotografii kratičký text.

Modelování databáze

Databázi jsem po tomto označení funkce a konzultaci se zadavatelem začal postupně modelovat. Na modelování jsem se rozhodl použít ERD diagramy jakožto rozšířeného nástroje na tvorbu relačních databází. Tento diagram definuje graficky potřebné struktury relační databáze-entity, relace, primární klíče a další. Pojmy jsou vysvětleny níže.

Celou databázi pomocí ERD diagramů jsem se rozhodl vytvářet v oblíbeném freewareovém nástroji Db Designer. Tento nástroj je velice kvalitní na to, že se jedná vlastně o freewareovou aplikaci. V tomto nástroji jsem navrhl její celkovou grafickou strukturu, včetně použitých relací a exportoval jsem v databázovém jazyku sql a kód vložil na server přes oblíbený program phpmyadmin.

Vlastní popis entit v databázi

Nyní bych si dovilil popsat jednotlivé entity a jejich vlastnosti a poté bych začal vysvětlovat jednotlivé relace a případně bych se zmínil o některých úskalích, která mě při tvorbě databáze provázela.

4.2 Popis relací a entit

Potřebné pojmy pro pochopení:

Entita

- je libovolná existující osoba, zvíře, věc či jev (obecně objekt) reálného světa. Entita musí být rozlišitelná od ostatních entit a existovat nezávisle na nich. V relační databázi označuje entita laicky řečeno tabulku v databázi.

Klíče

Primární klíč

- je pole nebo kombinace polí, jednoznačně identifikující každý záznam v databázové tabulce. Žádné pole, které je součástí primárního klíče, nesmí obsahovat hodnotu NULL. Každá tabulka může mít definovaný pouze jeden primární klíč.

Foreign key (Cizí klíč)

- je klíč v tabulce, na který se odkazuje položka z jiné tabulky.

Relace

Databázová relace se od matematické poněkud liší. Má zavedený pomocný aparát nazvaný schéma relace. Schéma relace říká, jaký je název relace, kolik má sloupců a jaké jsou jejich názvy a domény (doména určuje přípustné hodnoty v daném sloupci). V databázích je schématem relace definice struktury tabulky. Ovšem relací není jenom tabulka, ale cokoliv strukturovaného do řádků a sloupců, což znamená, že relací je i výsledek jakéhokoliv dotazu.

Kardinalita

- ta je přímo závislá a označuje mocnost relace, jednoduše řečeno kolik může mít rodič potomků. Příklad: 1:1, 1:n, M:N.

Dělení relací

1:1 - tato relace znamená, že můžeme jedné instanci entity přiřadit nanejvýše jednu instanci entity druhé.

1:n - libovolná instance jedné entity může být přiřazena k jedné či více instancím druhé entity.

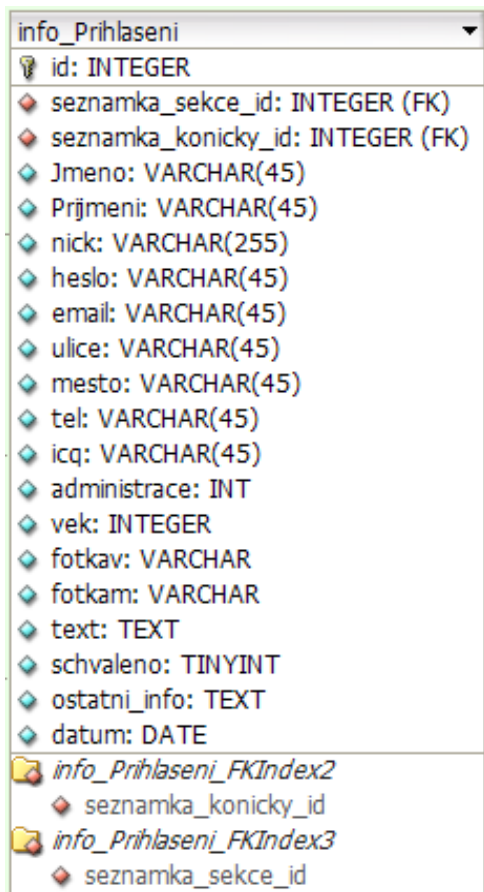
N:m - tato relace je již asi jasná libovolným počtům instancí může být přidělen libovolný počet instancí z obou entit, např. i 0.

1. Entita info_přihlášení a k ní související

Tato entita je hlavní entitou celé databáze a zahrnuje hlavní data celé databáze, a to i samotného redakčního systému. Sjednocuje do sebe základní informace o uživateli, včetně důležitých dat a je zároveň jádrem aplikace seznamka. S touto entitou byl s počátku největší problém. Musela se sjednotit s entitou záznamů seznamky. Důvodem bylo, aby nedocházelo k duplicitě dat a nebyla ohrožena stabilita databáze a také bylo možné pohodlnější spravování, potažmo lepší funkce. Pro lepší pochopení - celá databáze čerpá veškerá data o uživateli z této entity.

Relace bývají definovány cizím klíčem (FK foreign key nebo PFK primary foreign key). I když tato část měla být zařazena až dále, jak se ukázalo pro vysvětlení jednotlivých položek je nutná a je třeba jí zařadit. BD designér volí jak vidíme na obrázku první entity níže z položek: seznamka_sekce_id, seznamka_konicky_id kombinaci označení v podbodě

parenttablename_FK, jak je patrné už z označení v entitě v závorce mají (Fk).



1. seznamka_sekce_id - cizí klíč
2. seznamka_konicky_id - cizí klíč
3. obec_id - cizí klíč
4. jméno – jméno uživatele
5. příjmení- přímení uživatele
6. nick - nick uživatele
7. email - email
8. ulice – bydliště uživatele
9. tel - telefon
10. icq - icq
11. administrace - stanovení práv
12. věk
13. fotkam -miniatura pro seznamku
14. fotkav - fotka pro seznamku
15. text - vlastní text inzerátu pro seznamku
16. schváleno - udává status inzerátu v seznamce
17. datum - datum vložení do seznamky
18. ostatní info - informace všeobecné.

Popis důležitých položek pro entitu

Administrace - při návrhu bylo potřeba počítat, že uživatelé musejí být nějak rozděleni podle práv na serveru, toto zajišťuje právě tato instance. Do instance jsou vkládána číselná data od 1 – 4, které rozlišují jednotlivé uživatele.

Schváleno - jedná se o instanci, pomocí které bylo docíleno schválení či neschválení jednotlivých inzerátů do seznamky. Tato položka má charakter integer a může nabývat hodnot jedna až dvě.

Související entity

Entita obec

Obec
id: INTEGER
info_Prihlaseni_id: INTEGER (FK)
nazev: VARCHAR
Obec_FKIndex1
info_Prihlaseni_id

Tato entita zajišťuje uložení bydliště registrovaného uživatele. Je propojena s entitou info_přihlášení relací 1:1. Každý registrovaný by měl mít pouze jedno trvalé bydliště. Z toho jsem při návrhu vycházel a držel jsem se této skutečnosti. Důvod proč byla založena zvláštní entita na obec je doporučení z praxe, aby byla databáze připravena na případné vyhledávání pouze podle obce.

Entita seznamka sekce

seznamka_sekce
id: INTEGER
Nazev: VARCHAR(45)

Entita zajišťuje nastavení sekce a je spojena s entitou info_přihlaseni relací 1:N. Důvod byl následující. Původně se počítalo v souvislosti s aplikací, že půjde o možnost vložení více inzerátů do seznamky. Tato možnost byla později odmítnuta a bylo stanoveno, že půjde jen o jeden inzerát. Relaci jsem ponechal z důvodu možné změny v budoucnu.

Entita seznamka konicky

seznamka_konicky
id: INTEGER
info_Prihlaseni_id: INTEGER (FK)
nazev: VARCHAR(255)
seznamka_konicky_FKIndex1
info_Prihlaseni_id

Entita koničky vznikla z potřeby definovat a vyhledávat v seznamce podle koníčků, každý registrovaný může k sobě připojit libovolné množství koníčků. Tato entita je v relaci 1:N z jasného důvodu - jeden člověk mnoho koníčků.

Entity kraje a okresy

Význam těchto entit je jasný a nebudu je blíže specifikovat. Zajišťují pevný výběr okresu a poté automatické propojení kraje. Každá entita kraj je spojena relací 1:N s entitou okres. Automaticky se ví, jaký okres patří do jakého kraje. Bohužel bych rád naznačil, že

seznamka_Kraj
id: INTEGER
seznamka_Okres_id: INTEGER (FK)
nazev: VARCHAR(45)
seznamka_Kraj_FKIndex1
seznamka_Okres_id

došlo k malé modifikaci a místo toho, aby byla propojena entita kraje s entitou okresů, byla entita krajů propojena přímo s info přihlasení. Prozatím bude ponechána

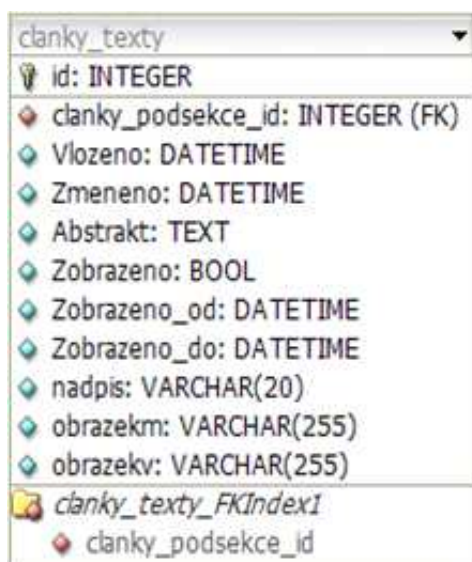
seznamka_Okres
id: INTEGER
id_kraje: INTEGER
nazev: VARCHAR(255)

databáze beze změny, protože tato změna je zvažována a je možné, že bude aplikace upravena zpět na původní model!

2. Entita články texty a k ní související

Články texty

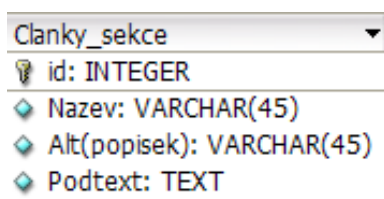
Tuto entitu lze označit opět za zhruba stěžejní u ukládání a publikování článků. Poskytuje základní informace o článku, je souvisle propojena z dalšími entitami v databázi a poskytuje jakousi kostru v náhledu na tuto část databáze. Opět z této entity probereme důležité instance a popíšeme jejich vztahy s ostatními entitami.



1. id-primární klíč (viz vysvětlení pojmů)
2. clanky_podsekce_id - foreign key (viz vysvětlení pojmů)
3. Vloženo+změněno+zobrazení (od či do) - zahrnují pouze datумы daných významů
4. abstrakt - krátký obsah článku
5. nadpis- nadpis, podle kterého probíhá hledání je povinnou položkou
6. obrázek (v a m) - miniatura + obrázek ve skutečné velikosti.
7. zobrazeno - položka označuje zveřejnění či nezveřejnění článku.

Související entity

Sekce články:



Entita zajišťuje rozvrstvení článků do jednotlivých sekcí. Její položky jsou rozvrženy vzhledem k účelu jen na název, alt (popisek) zobrazovaný po najetí myši na odkaz. Poslední položkou je podtext. Tato položka byla přidána z důvodů možných poznámek pod odkazy na sekci.

Články podsekce

clanky_podsekce
id: INTEGER
Clanky_sekce_id: INTEGER (FK)
název: VARCHAR(45)
Alt(popisek): VARCHAR(45)
Podtext: TEXT
clanky_podsekce_FKIndex1
Clanky_sekce_id

Tato entita je propojena relací 1:N s entitou clanky_sekce. Důvod pro použití této relace je jasný - jedna sekce může mít nekonečně mnoho podsekcí. Všechny položky jsou totožné s předešlou entitou, tudíž si je dovolím nerozebírat blíže. Funkce této entity je také jasná, proto bych se k ní už blíže nevyjadřoval.

Články odstavce

Tato entita je důležitou součástí databáze, pokud se týká publikování článků. Bylo nutné z hlediska funkce zajistit vkládání článků po odstavcích, aby se dal článek lidově řečeno složit z jednotlivých částí dohromady a byla zajištěna jeho co největší variabilita a pohodlná práce. Odstavce jsou propojeny s entitou články texty relací 1:N a ke každému článku je přiřazen volený počet odstavců.

clanky_odstavce
id: INTEGER
clanky_texty_id: INTEGER (FK)
Text: TEXT
Link: VARCHAR(255)
Poradi: INTEGER
nadpis: VARCHAR
cestamale: VARCHAR
cestavelke: VARCHAR
cestamale1: VARCHAR
cestavelke1: VARCHAR
zaronani: INTEGER
zarovnani1: INTEGER
clanky_odstavce_FKIndex1
clanky_texty_id

1. id - primární klíč
2. clanky_texty_id - foreign key
3. text - vlastní text odstavce
4. link - odkaz do odstavce
5. poradi - zajišťuje pořadí odstavce ve výpisu.
6. nadpis - nadpis odstavce, je-li požadován
7. cesta (male či velke 1 či 2) - cesta k obrázkům na serveru.
8. zarovnani (1 a 2) - zarovnání jednotlivých obrázků.

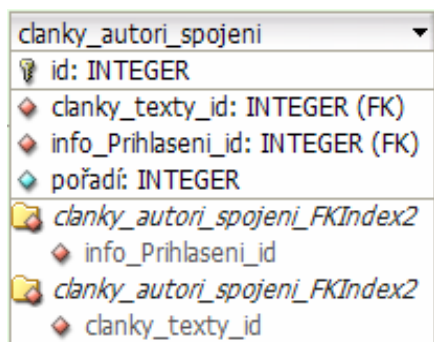
Bližší popis důležitých instancí

Cesta (malé či velké 1 či 2) - označuje cesty k jednotlivým obrázkům na serveru. Dvě jsou proto, že ke každému odstavci se dají přiřadit 2 obrázky.

Zarovnání - označuje hodnotu podle toho kam je daný obrázek zarovnán, jestli doleva či doprava, což umožňuje slušnou variabilitu. Tato varianta řešení byla navržena s ohledem na požadovaný normativ článku a snadnou obsluhu.

Články autoři

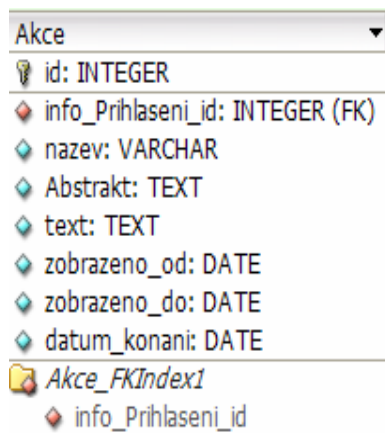
Tato entita je specifická svým obsahem. Jde o entitu spojující dvě entity, která obsahuje dva cizí klíče (foreign key) a jednu instanci pořadí. Její funkce v databázi je následující. Bylo potřeba jednotlivé články variabilně spojit v jeden celek., což je naznačeno už v předešlé části.



Tato entita přiděluje jednotlivým článkům autory, a to zcela specifickým způsobem. Bylo potřeba autory vkládat a prověřovat zároveň. Autor tudíž musí být hledán v nějakém ověřeném seznamu, z čehož vyplývá, že musí být ověřen administrátorem. Tak se zrodil nápad propojit autory a Info_přihlášení a zároveň, aby bylo vidět k čemu patří byl přidán další cizí klíč na propojení clanky_texty a autori. Protože autorů mohlo být mnoho, ne jen jeden, bylo přidáno i pořadí. V systému probíhá přiřazování automaticky, vkladatel pouze najde autora v ověřeném seznamu z tabulky info_přihlazení a přiřadí jej.

Entita Akce

Entita zajišťuje v databázi prezentování akcí. Její funkce v databázi se domnívám, že je jasná a není třeba naznačovat více.



1. *id* - primární klíč
2. *info_prihlaseni_id* - foreign key
3. *nazev* - název akce
4. *abstrakt* - krátký popis akce
5. *text* - vlastní text
6. *zobrazeno (od-do)* – datum od kdy do kdy je vyvěšena na webu.
7. *datum_konani*

Důležité entity

info_prihlaseni_id - tato instance označuje kým byla akce do databáze vložena.

Entita deníčky a s ní související

Tato entita zajišťuje jakousi hlavičku části databáze související s fotogalerií, která na portálu zajišťuje prezentaci fotografií. Každý registrovaný si u deníčků může založit libovolně mnoho fotogalerií. Aby bylo možné rozlišit komu deníček patří, byl vložen cizí klíč `info_prihlaseni_id`. Daná entita je propojena relací 1:N s entitou `info_prihlaseni`.

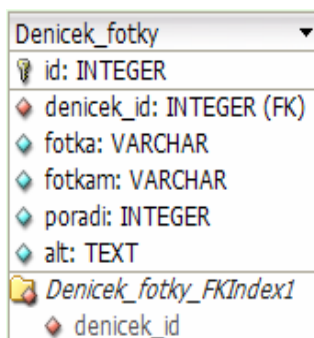


denicek	
id	INTEGER
info_Prihlaseni_id	INTEGER (FK)
nazev	VARCHAR
text	TEXT
denicek_FKIndex1	
info_Prihlaseni_id	

1. id – primární klíč
2. info_prihlaseni_id - foreign key
3. nazev - nadpis deníčku
4. text - text popisující obsah deníčku

Související entita deníček fotky

Entita je propojena z entitou `denicek` relací 1:N, protože ke každé hlavičce je možné přiřadit libovolné množství fotografií. Dalšímu popisu entity neuvádím, myslím si, že je dostatečně jasný.

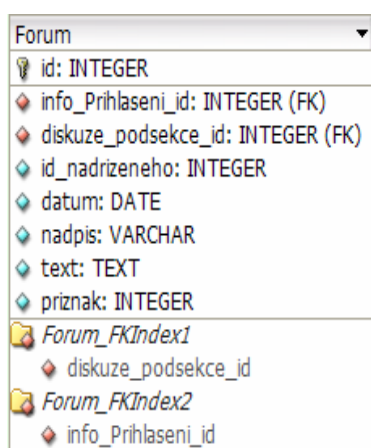


Denicek_fotky	
id	INTEGER
denicek_id	INTEGER (FK)
fotka	VARCHAR
fotkam	VARCHAR
poradi	INTEGER
alt	TEXT
Denicek_fotky_FKIndex1	
denicek_id	

1. id – primární klíč
2. denicek_id - foreign key
3. fotka(m) - miniatura a původní fotografie
4. poradi - pořadí vložené fotografie
5. alt - popisek obrázku zobrazující se po najetí myší

Entita fórum a s ní související

Tato entita je stěžejní pro diskuzi provozovanou na portálu. Jsou do ní ukládány všechny příspěvky, datum vložení, nadpis a text vlastního příspěvku a dále příznak, jehož význam vysvětlím později. Tato entita je propojena s entitou Diskuze podsekce relací 1:N z důvodu, že v jedné podsekci může být mnoho příspěvků.



Forum	
id: INTEGER	
info_Prihlaseni_id: INTEGER (FK)	
diskuze_podsekce_id: INTEGER (FK)	
id_nadrizeneho: INTEGER	
datum: DATE	
nadpis: VARCHAR	
text: TEXT	
priznak: INTEGER	
Forum_FKIndex1	
diskuze_podsekce_id	
Forum_FKIndex2	
info_Prihlaseni_id	

1. id - primární klíč
2. info_prihlaseni_id - foreign key
3. id_nadrizeneho - foreign key
4. datum – datum vložení
5. nadpis
6. text
7. priznak

Důležité instance

info_prihlaseni_id - označuje kdo příspěvek vložil

id_nadrizeneho - jelikož při vkládání příspěvků mohou uživatelé reagovat na jakýkoli příspěvek, je nutné uložit na koho reagují.

Priznak - určuje podle zadání o jaký příspěvek se jedná, jestli jde o klasický příspěvek nebo o příspěvek odborníka. Důvodem je, že diskuse se dělí na diskusi s odborníky a klasickou volnou.

Související entity

Diskuse, podsekce

Tyto dvě entity zajišťují tu samou funkci, jako entity u clanky_sekce a clanky_podsekce, proto bych se je dovolil už nepopisovat, relace mezi nimi jsou také totožné.

diskuze_sekce
id: INTEGER
nazev: VARCHAR(255)

diskuze_podsekce
id: INTEGER
diskuze_sekce_id: INTEGER (FK)
nazev: VARCHAR(255)
diskuze_podsekce_FKIndex1
diskuze_sekce_id

Inzerce + vkládané horoskopy

Tyto dvě části jsou natolik podobné, že jsem se rozhodl je shrnout do jednoho popisu i relace mezi nimi jsou totožné. Liší se pouze funkcí, což je určitě pozornému čtenáři jasné z jejich názvu a účelu.

Inzerce - vkládání prolinků do jednotlivých sekcí s logy firem.

Horoskopy - vkládané horoskopy dle zvěrokruhů.

Entity inzerce sekce + Horoskopy_znameni

Tyto dvě entity jsou totožné stavby, jen jejich účel se liší. Do názvu u horoskopů se vkládají znamení zvěrokruhu, kdežto u inzerce_sekce sekce pro inzerci.

inzerce_sekce
id: INTEGER
nazev: VARCHAR

1. id - primární klíč
2. název - jak je popsáno výše, je u obou pouze odlišný vkládaným obsahem.

Entity horoskopy_text a inzerce_prolinky

Tyto obě entity jsou propojeny k předešlým relací 1:n. Důvodem je opět, že obě výše popisované entity mohou mít mnoho entit pod sebou. U těchto entit postupně vysvětlím jednotlivé instance a poté bych popis opustil.

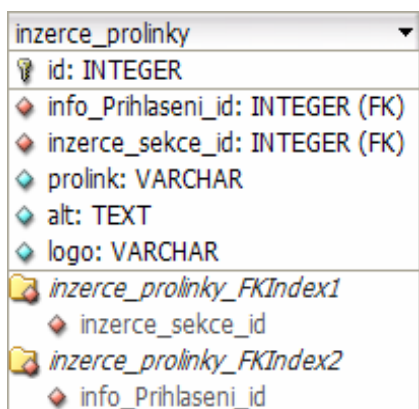
Horoskopy_text
id: INTEGER
info_Prihlaseni_id: INTEGER (FK)
Horoskopy_znameni_id: INTEGER (FK)
abstrakt: INTEGER
text: TEXT
Horoskopy_text_FKIndex1
Horoskopy_znameni_id
Horoskopy_text_FKIndex2
info_Prihlaseni_id

1. id - primární klíč
2. horoskopy_znameni_id - foreign key
3. info přihlazení id - foreign key

4. abstrakt - krátký úvod
5. text-vlastní text

Důležité instance

info_prihlaseni_id - tato instance označuje kým byl horoskop do databáze vložen.



1. id –primární klíč
2. info_prihlaseni_id -foreign key
3. prolink -adresa prolinku
4. alt - krátký popis
5. logo - cesta k logu v podobě fotografie

5. Popis adresářové struktury portálu a důležitých funkcí

Tato část by měla čtenáři přiblížit jakýsi náhled na celou stavbu portálu z hlediska adresářové struktury, zevrubně a velmi stručně popsat důležité skripty a umožnit si udělat jakýsi funkční náhled z hlediska tvůrce portálu a ne uživatele.

5.1 Popis adresářové struktury

V souvislosti s vývojem portálu vznikla potřeba vytvořit si jakousi adresářovou strukturu pro ukládání souborů a načítání šablon. Každý soubor má na portálu své specifické místo a funkci. Skripty php jsou na portálu umístěny v kořenovém adresáři a další části byli rozčleněny do jednotlivých podadresářů z hlediska jejich funkce.

Výčet adresářů:

1. CSS - v tomto adresáři jsou uloženy veškeré kaskádové styly potřebné pro úpravu portálu.
2. Ikona prolink - v souvislosti s prolinky sou zde ukládány ikonky pro zobrazení na webu.

3. Obrázky články - tento adresář obsahuje obrázky určené k vkládání do článků. Z hlediska toho, aby bylo ušetřeno zatížení serveru má i 2 podadresáře malé (miniatury obrázků), velké (velké obrázky).
4. Obrázky deníčky - jeho obsah je stejný jako obsah předešlého, pouze se jedná o fotografie z fotoalb.
5. Seznamka - obsahem také adresář totožný, jedná se o fotografie ze seznamky portálu.
6. Template - tento adresář je velice důležitý. Jsou v něm uloženy veškeré podklady pro vzhled portálu. Má jediný podadresář a tím je inc. V něm jsou uloženy šablony dle jmen, které jsou inkudované do hlavních - viz příloha s ukázkou šablon.
7. Images - tento adresář je také velice důležitý. Je v něm uložen veškerý materiál doplňující vzhled portálu, animace, obrázky atd.
8. Javaskript - zde jsou uloženy důležité funkce pro provoz portálu napsané v tomto jazyku.

Popis portálu z hlediska funkce

V této části bude popsána celá stavba portálu z hlediska programátora, ale nebude popisována stavba jednotlivých skriptů. Každá z jednotlivých aplikací i samotný redakční systém má v sobě příslušné skripty v php, jak bylo popsáno v minulých částech, které se týkaly zvolených technologií. Celý portál je napsán v tomto programovacím jazyku a popsán popisky, a tudíž nebude podrobně rozebírán, jelikož se nachází v příloze na CD a každý si jej může vyhledat, pouze zde bude vytvořena zevrubná kostra v náhledu na portál.

Celá část je strukturovaně rozdělena na 3 souvislé části, a to na skripty týkající se redakčního systému, skripty týkající se portálu z hlediska uživatele a dále na soubory obsahující důležité funkce nutné pro chod portálu, jako jsou funkce na úpravu obrázků, datum a testování vstupů pro formuláře.

5.2 Výčet skriptů týkajících se redakčního systému

1. Administrace.php - tento skript zajišťuje odeslání formuláře.
2. Administracniskript.php - zajištění ověření uživatele jména a hesla. Každému uživateli je do SESSION přidělen příznak 1, který se testuje na celém redakčním systému. Po této akci je rozhodnuto, jak již bylo popsáno v předešlé části jeho administracní právo, to je načteno do databáze a uloženo do SESSION a dále testováno na celém webu.
3. Akce2.php, - pouze obsluhuje formulářem správu a editaci akcí na portálu.
4. Centralniregistrace.php - tento skript je jeden z nejdůležitějších na portálu. Umožňuje správu jednotlivých redaktorů na webu a zajišťuje přidělení práv.
5. Clankyvkladaniautori.php - skript obsahuje kódy na přidělování autorů pro portál. Přidělování autorů je popsáno níže.
6. Clankyvkladaniodstavce2.php - tento skript zajišťuje správu každému odstavci

jednotlivých článků. Jeho uživatelská funkce je vysvětlena níže v části věnované grafickému rozhraní. Skript je úzce spjat s funkcí v souboru funkceposuv.php. Její význam je vysvětlen níže.

7. Diskuze sprava.php - skript obsahuje výpis a hledání v diskuzích, hledá se pomocí dotazů Mysql. Každý příspěvek se dá vymazat i se souvisejícími příspěvky vhodnými sql dotazy. Jeho role a rozhraní je vysvětleno v části věnované GUI.
8. Diskuzepodsekce nastaveni.php - vkládání podsekcí formulářem do databáze.
9. Diskzesekceni nastaveni.php - vkládání sekcí formulářem do databáze.
10. Horoskop nastaveni.php - vkládání horoskopů zvěrokruhu.
11. Menu.php - obsahuje hlavní grafickou stránku celého redakčního systému. Tento skript je inkudován v celém redakčním systému jako hlavní layout a vzhled.
12. Nastaveni spojeni databaze.php - vůbec nejdůležitější skript. V tomto skriptu je celé nastavení spojení na Databázi, heslo na server atd.
13. Odhlaseni.php - jeho obsah je krátký, ale velice důležitý. Obsahuje procedury na vymazání session používaných na přihlášení na portále.
14. Podsekce clanku nastaveni.php - vložení jednotlivých podsekcí k článkům.
15. Potvrzeni clanku.php - zajišťuje důležitou roli v administraci článků, aby bylo možné články potvrzovat a mazat v případě, že nevyhovují obsahem. Každý článek má v entitě články_texty položku schváleno či ne, podle toho je vyvěšen či nevyvěšen na portál.
16. Prolinky sekce.php - sekce pro vkládané prolinky.
17. Prolinky vkladani.php - tento skript zajišťuje vkládání prolinou, ukládání a úpravu obrázků.
18. Sekce clanku nastaveni.php - vkládání sekcí článku.
19. Seznamka nastaveni 1.php - vkládání krajů.
20. Seznamka nastaveni 2.php - vkládání okresu.
21. Seznamka nastaveni 3.php - vkládání koníčků.
22. Seznamka nastaveni 4.php - vkládání sekcí.
23. seznamka potvrzeni.php - administrace inzerátů.
24. Vkladani clanku.php - zajišťuje vkládání článku na portál.
25. Vlozeni horoskop.php - vložení horoskopu na portál.

Skripty zajišťující chod portálu

26. Registrace sprac.php - zpracování registrace na portálu. Činnost je stejná jako u administrace.
27. Seznamka zadani 2.php - plní šablonu se zadávacím formulářem na portál.

28. Denicek.php - plní šablonu deníčků.
29. Denicekfotky2.php - vkládá fotografie jednotlivých uživatelů.
30. Denicekhlavicka.php - vkládá jednotlivé hlavičky z texty od uživatelů.
31. Seznamka.php – spravuje celou aplikaci seznamka.
32. index.php-hlavní strana portálu plní její šablonu.
33. Odhlasiportal.php - odhlašuje ze všech stran portálu a vymazává session.

5.3 Funkce zajišťující chod portálu

34. Funkce.php - tento skript je velice důležitou součástí portálu. Obsahuje funkci na zpracování obrázků a dokáže kompletně obrázek předělat. Obrázek typu gif,jpg,ngf transformovat, respektive přepočítat na daný rozměr a upravit kvalitu.
35. Funkceposuv.php - zajišťuje změnu údajů v databázi. Byla využita například u změny pořadí fotografií v deníčku a také u odstavců v článcích.
36. Htmpltmpl.php - tento skript je důležitý pro šablonování systém a obsahuje kompletní sadu pro jeho implementaci.
37. Datumn.php - slouží pro práci s datem a jeho převod do formátu vyžadovaného portálem.
38. Datum.php - zajišťuje převod datumu ve zpětné konverzi.
39. Testyvstup.php - skript obsahuje kompletní funkce psané na testování jednotlivých vstupů. Testy probíhají pomocí regulárních výrazů.

Popis funkcí

V systému je několik funkcí.První z nich je na zpracování obrázků. Zajišťuje jejich zpracování pro celý systém a je k nahlédnutí na příloženém cd. Tato funkce transformuje automaticky kvalitu i rozměry obrázků a ukládá je na server dle dané cesty a je použita po celém systému.Uložena je v souboru funkce.php.

Další z důležitých funkcí zajišťuje ošetření vstupů. Z hlediska ušetření času je tato část opět zajištěna funkcemi, které automaticky generují hlášení o chybách. Tyto funkce se nachází v souboru testyvstup.php. Každý vstup je kontrolován regulárními výrazy pomocí php. Regulární výrazy jsou velice důležitým a skvělým nástrojem. Dále tyto funkce zajišťují i odstranění Javaskriptu a HTML a samozřejmě i php z ohrožených vstupů, aby byla zvýšena bezpečnost systému.

Dále se v systému funkce zajišťující uživatelskou stránku redakčního systému. Jejím úkolem je zajistit dotazy při mazání některých důležitých položek a dále obsahuje omezení případně ořezání důležitých textů vkládaných do databáze. Tato funkce je jako jediná napsaná v jazyku Javaskript.

Poslední funkcí v systému je funkce na převod datumu. Každý datum musí být převeden

do formátu v databázi a poté zpět na formát zadaný uživatelem nachází se v souboru datumn.php.a datum.php.

6. Redakční systém

6.1 Proč nebyl použit nějaký ze stávajících systémů.

Vzhledem k požadavku celého zadání a dále i osobní domluvě jsem se rozhodl vytvořit software na míru. Důvod byl jednoduchý. Žádný z mě známých systémů nevyhovoval požadavkům. U systémů stávajících se objevovaly následující vady - buďto absolutně nevyhovovaly požadavkům, ať funkcí či přílišnou složitostí. Vzhledem k cílové skupině uživatelů a přání zadavatele, které zahrnovalo aby byl systém systematický a jednoduchý na obsluhu. Po mém uvážení a projití jsem zjistil, že vzhledem k požadavkům i když jsou kódy systémů vždy otevřené k modifikacím, by mi ze systémů buď zbylo jenom přihlášení a správa registrovaných a zbytek bych musel dodělat na přání,nebo byl kód tak složitý, že než bych se dopracoval k výsledku stálo by to mnoho úsilí a vzhledem k účelu, ke kterému byl systém tvořen, by se práce nevyplatila. Dalším aspektem byl i ten důvod, že moderní redakční systémy obsahují obvykle wysing editor, a to i když je prvek žádaný a moderní, pro naši aplikaci je velice nepraktický. Představa seniora vytvářejícího články i s obrázky ve wysing editoru je dle mého názoru velice nepravděpodobná. I když bychom se přes tento problém přenesli, nastal by další, a to úplná volnost uživatelů z hlediska zadávání vzhledu článků, která by vedla k chaosu ve vzhledech článků na portálu, což je nepřijatelné z důvodu jeho necelistvosti. Každý portál na internetu má články vkládané na určitém normativu, či laicky řečeno šabloně, aby byla zajištěna jeho přehlednost a grafická celistvost.

6.2 Členění systému

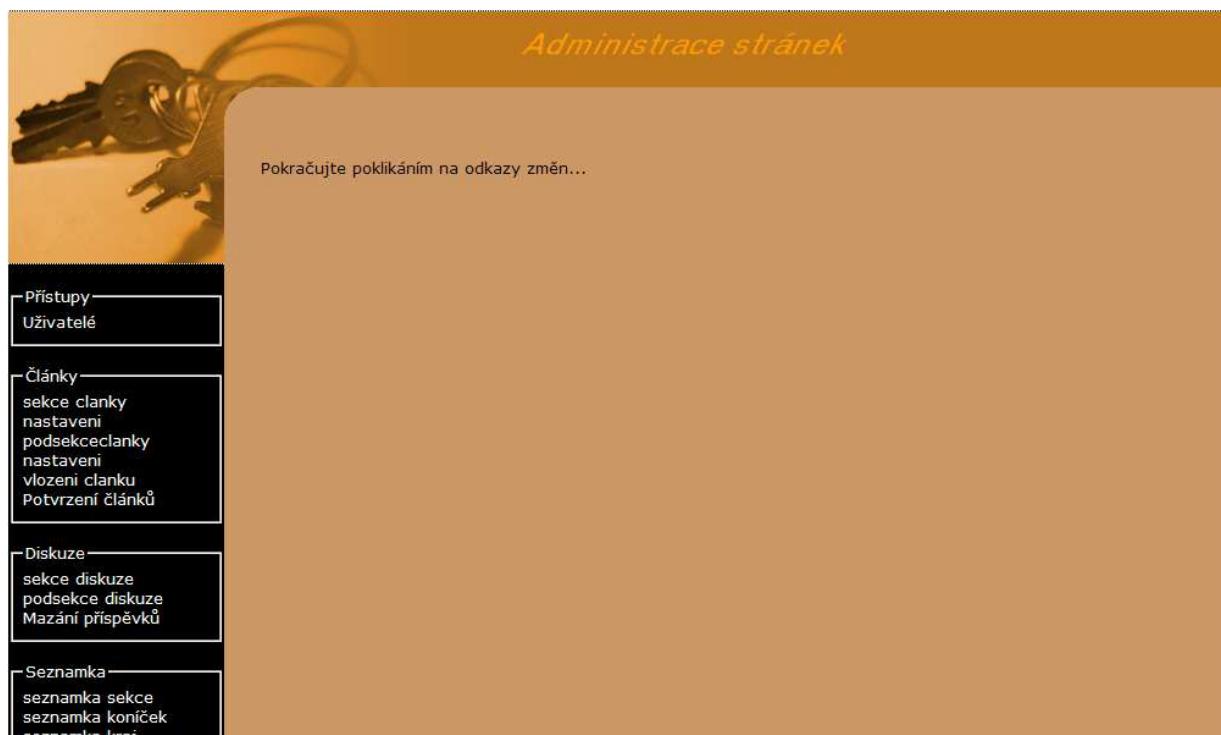
- 1.1 Přístupy
- 1.2 Články
- 1.3 Diskuze
- 1.4 Deníčky
- 1.5 Vkládané prolinky
- 1.6 Akce

Grafické prostředí (gui)

O grafickém prostředí není nutná dlouhá zmínka. Nemá totiž dle mého názoru cenu popisovat, jak byl vytvářena to z jednoduchého důvodu, že tato práce se primárně nezabývá tvorbou grafiky na internetu. Vzhledy jsou vytvořené klasickou a dnes běžnou formou kombinací HTML a Css. Ve vzhledu redakčního systému nebyly použity šablony, protože tato

funkce nebyla bezpodmínečně nutná .

Ukázka GUI pro portál senior info:



6.3 Popis jednotlivých částí systému

Přístupy

V této části byla dle zadavatelky zrealizována správa přístupu a udělení práv. Práva jsou rozdělena do tří částí - administrátor, editor a uživatel. Do redakčního systému mají přístup jen první dva subjekty výčtu. Administrátor může měnit cokoli na portálu, měnit sekce, podsekce, vkládat články a další věci a samozřejmě zadává práva ostatním a v této části zadává o nich i důležitá data. Práva jsou řešena v entitě info_přihlášení instancí administrace, je jí přidělována hodnota od 1-3 v uvažovaném pořadí. Práva editora jsou dle zadání také jasná - do systému se smí přihlásit, ale jak vyplývá z jeho funkce může pouze vkládat články a měnit je, ale pouze své články. Jiné změny mu budou systémem znemožněny.

Další funkcí, která je v této části důležitá, je vyhledávání. Po čase se ukázalo, že uživatelů bude na portálu mnoho a bylo proto nutné v nich hledat potřebné údaje.

Články

Sekce, podsekce

Tato část systému je neobsáhlejší a její realizace vyžadovala nejvíce úsilí a času. Články jsou dělené do sekcí a podsekcí. K nim se dají přidávat popřípadě alt (popisek po najetí myši) a podtext v případě potřeby. Podtextem se uvažuje krátká informace umístěná pod názvem sekce.

Vkládání hlavičky článku

Vložené články							
ssssss	12		upravit	smazat	odstavce	autoři	Náhled
dihalsf	12		upravit	smazat	odstavce	autoři	Náhled
jokokosos	10		upravit	smazat	odstavce	autoři	Náhled
Sare nora	10		upravit	smazat	odstavce	autoři	Náhled

Další částí je vlastní zadání hlavičky článku. Na výběr je podsekce, nápis, abstrakt (krátká informace o obsahu článku) a zobrazení od do. Samozřejmě je i přidání obrázku. Přidávání obrázků a jejich vkládání na server zajišťuje funkce, o které se zmiňuji v předešlé části. Poté je zde možnost hledání ve vložených článcích a jejich stručný výpis pro případnou úpravu a mazání. Dalšími důležitými položkami jsou odstavce, autoři a náhled. Náhled umožňuje podívat se na celkový vzhled

článku. Jak jsem se již zmínil v předešlé části návrhu databáze, jsou články skládány z jednotlivých částí. V tomto výpisu je samozřejmě doplněno i stránkování aby byl omezen počet výpisů článků na stránce. Tato část je aplikována na entitu clanky_texty popsanou výše.

Odstavce

Zpět na vkládání článků

babicby fdsfsd 37 1 upravit smazat

Po kliknutí na část odstavce se nám po načtení objeví nabídka s možností přidávání odstavců k článku. Odstavců se dá přidávat libovolné množství a ke každému z nich je možnost přidání dvou obrázků se zarovnáním napravo a nalevo. Obrázky jsou zpracovány automaticky a jejich zpracování je probráno v předešlé části. Dále se dá ke každému odstavci přiřadit jeho pořadí při výpisu. Každý odstavec automaticky generuje dvě tlačítka na vymazání odstavce a na jeho úpravu. Úprava umožňuje kdykoli upravit

odstavec. Mazání má specifickou funkci. Pokud se rozhodnete smazat odstavec, ostatní nad ním jsou automaticky přečíslovány (tím se rozumí úprava jejich pořadí). Pokud uživatel přehlédne nebo špatně vloží odstavec a jeden nechťně vynechá, může bez jakýchkoliv problémů mezi tyto dva odstavce vložit další a jejich pořadí se také automaticky přepočítá. Tím byla zajištěna maximální variabilita při vkládání a v případě vymazání odstavce jeho možné nahrazení. Další položky formuláře nejsou ničím specificky zajímavé, takže bych se jejich popisem nezabýval. Jediná funkce, která stojí za zmínku, je samozřejmě to, že výpis odstavců je samozřejmě také doplněn stránkováním. Tato část pracuje s entitou databáze `clanky_odstavce`. Celá funkce a jednotlivé entity jsou popsány v návrhu databáze.

Autoři

Zpět na vkládání článku

Výpis přiřazených autorů

jméno příjmení	pořadí
Martin Lujc	2
Martin Lujc	1

Tato část přiřazuje danému článku jeho autora. Každý autor musí být ověřen, z tohoto jsme při návrhu databáze i této části vycházeli. Každý autor je registrovaný a ověřený, tudíž tato část funguje následovně. Daný autor je vyhledáván podle jména, příjmení, emailu nebo icq v databázi. Poté, pokud se najde, objeví se jeho jméno v select boxu Autor. V tom si uživatel vybere z možností, které byly nalezeny a vyplní kolonku pořadí, která je povinná a autora jednoduše přiřadí k danému článku. Poté je dole umístěn výpis, který velice snadno a přehledně

poskytuje informace o přiřazených autorech a jejich pořadí. Každého autora můžeme samozřejmě z výpisu vymazat. Tato část pracuje v databázi s entitou `clanky_autori` a její podrobný popis byl uveden včetně relací v návrhu databáze.

Potvrzení článku

Tato funkce je čistě administrátorská. Důvodem zavedení je přání zadavatele přečíst si články před vyvěšením na web. Zadavatel má dále možnost se na celý článek podívat z náhledu a v případě, že není s článkem spokojen ozvat se danému autorovi nebo článek upravit sám přes předchozí zadávání nebo jej prostě smazat. V případě, že je spokojen, klikne na tlačítko potvrdit a článek vyvěsí. Tento

výpis je samozřejmě omezen stránkováním. Důvodem je možnost vyššího provozu a tudíž by bylo nepohodlné spravovat výpis bez něho. Každé potvrzení je spravováno přidáním příznaku do entity `Clanky_texty`. Schválený článek je značen číslem 1 neschválený číslem 2.

Diskuse



Tato položka zajišťuje zprávu diskusí na portálu. Její popis bude mírně zkrácen, protože již byl částečně proveden v předešlé kapitole. Bylo by zbytečné znovu popisovat záložky sekce, diskuse a podsekce diskuse. Jak je již zřetelné z názvu, je jejich funkce stejná jako u článků. Jedinou zajímavou položkou, která je v této části, je správa diskusí jako takových. Jak je zřejmé z provozu různých diskusí na internetu, každý příspěvek se musí dát v dané

diskusi zpravovat. Jinak by mohlo dojít k nekontrolovanému rozvoji diskuse na témata, jak se říká „nežádoucí“. Z tohoto důvodu zde bylo umístěno mazání a hledání příspěvků, aby byla zajištěna jejich pohodlná administrace. V této části pracuji s entitami databáze Diskuze_podsekce, diskuze_sekce a forum, jejich podrobný popis je v návrhu databáze včetně relací.

Seznamka

Seznamka, která je další položkou, má více částí, některá nich bude popsána a jiná zase vynechána, např. sekce, jejíž funkce je zřejmá, dále koníčky, zadání kraje a okresu jsou také zřejmé. Další položkou je jediná, která je opět důležitá z důvodu administrativy daného problému, a to je potvrzení inzerátů. Každý nepotvrzený inzerát se zde vypíše a administrátor ho může buď schválit nebo smazat. Pro pohodlnější práci s výpisem je vše doplněno praktickým stránkováním. V této části jsem použil entity seznamka_sekce, info_prihlaseni, seznamka_konicky a seznamka_kraj. Jejich relace a vztahy jsou opět popsány v části návrhu databáze.



Horoskopy + odkazy

Tyto dvě části byly spojeny, protože se jedná o funkčně totožné části, pouze se liší účelem. Každý prolink má na výběr sekci, do které patří a dále link (adresa na internetu), poté se dá ještě v případě zájmu připojit i malá ikonka ve formě obrázku. Pokud se týká druhé popisované části, jedná se o vkládání horoskopů. Obsahuje vybrání sekce, vložení abstraktu a dále jeho vlastní text. Každý horoskop se dá editovat a upravit či vymazat. Stejné pravidlo platí i u prolinků. V této části se pracuje s těmito 4 entitami databáze: horoskopy, horoskopy_text, horoskopy_znamení, inzerce_prolinky, inzerce_sekce. Jejich vztahy a relace jsou popsány opět v části věnované návrhu databáze.

Akce:



Vložte požadovanou akci:

*Název akce:

*Abstrakt:

*Text:

*Zobrazení od:

*Zobrazení do:

*datum:

expoziční

Beh foxe

Trhy

Lov zvířete

Hajení

vpřed

Akce jsou poslední součástí redakčního systému. Zadávatelka si přála, aby bylo možné tuto zájmovou skupinu seznámit na pro ni zajímavé akce. Jak je patrné, na tuto položku opět dosáhne pouze administrátor, neboť akce musí být ověřené a musí se k nim podat odpovídající informace. Vkládací formulář jak patrně z obrázku se dělí na Název, abstrakt, text, zobrazení (od - do), datum (jedná se o datum zveřejnění). Dále je dole doplněn výpisem vložených akcí a celá část je dále doplněna stránkováním, aby byla zajištěna pohodlnější manipulace. Samozřejmostí je také možná editace a úprava, v případě potřeby i smazání

akcí. Část databáze obsluhovaná touto částí se nazývá Akce.

7. Problematika vzhledů na portále

Vzhled byl kompletně vytvořen pomocí HTML a dodán v podobě tabulkového layoutu s menším začleněním CSS. Vše bylo implementováno v jediném souboru HTML i s CSS. Bylo třeba celý vzhled přepracovat a začlenit ho kompletně do adresářové struktury portálu. To znamenalo vše umístit na rozvržené místo dle popisu adresářové struktury (viz předešlé kapitoly). Tabulky byly odstraněny z důvodu standardu, a také z důvodů jejich problematické implementace na webu. Tabulkový layout je zastaralý a nepružný. V případě překročení možností jednotlivých buněk automaticky dochází ke znehodnocení celého grafického návrhu.

7.1 Rozložení vzhledů na portále

Celý portál má přesně definovanou adresářovou strukturu. Všechny šablony jsou uloženy v adresáři Template a srozumitelně pojmenovány podle toho, k jakým částem portálu patří. Každá šablona má mnoho vrstev a do sebe včleněných částí, které jsou uloženy v adresáři inc a opět pojmenovány velice intuitivně, popřípadě zvýrazněny popiskou v kódu dodaném v příloze na cd.

Nyní přistoupíme k vysvětlení činnosti. Podobně jako se ve skriptovacím jazyku php dosahuje skvělých výsledků příkazem include. V načítání stránek je v šablonovacím systému jeho obdoba. Také příkaz include, jeho funkce a syntaxe je vysvětlena v příloze. Důvodem proč se dané části šablon nacházejí v adresáři inc spočívá v samotném šablonovacím systému, jeho implementace má v sobě tento adresář nastaven. Pokud bychom ho změnili, dané části by byly nefunkční. Jak sem již zdůraznil v příloze a tuto skutečnost zde zdůrazním znovu, je celá šablona plněna jediným souborem php. Tak vzniká otázka, která člověku znalému programování určitě neunikla - jak pozná šablonovací systém jakou vrstvu má zviditelnit? Vše je zajištěno pomocí TMPL_IF jak je vysvětleno v příloze. Pokud má hodnotu true, je naincludována daná šablona. Tudíž je zřejmé, že se šablonovacím systémem se pracuje velice pohodlně a je velice flexibilní. Pokud se týká skriptů v jazyku php plním jednotlivé šablony webu, je jejich pojmenování totožné jako dané části portálu. Každá s těchto částí, které jsou popsány, je k nahlédnutí v příloze na cd.

Dále se ještě zmíníme o dalších skutečnostech, které souvisejí se vzhledy. Jak jsme již zdůraznili, celý vzhled je kombinací HTML +css a daných obrázků, animací a dalších souborů, které jsou uloženy v adresáři images, kaskádové styly neboli CSS jsou uloženy v adresáři CSS, jak již bylo zdůrazněno v předešlých kapitolách. Každý vzhled nebo chcete-li šablona použitá v tomto systému, načítá své styly ze dvou souborů, které se nazývají vzhled.css a , aby bylo zajištěno snadné případné předělávání či úprava. Názvy jednotlivých class jsou v souboru pojmenovány tak, aby byly intuitivní a snadné na pochopení. Všechny skripty jsou k nahlédnutí na přiloženém cd.

8. Stručný popis funkce portálu

Jak bylo zdůrazněno v předešlých částech, je portál dělen na několik částí. Hlavní strana s publikací vkládaných článků, seznamka, deníčky, diskuze a vkládané prolinky. Všechny části byly popsány z hlediska vývojáře v předešlých částech. V této části se pokusíme zdůraznit jejich uživatelskou funkci na portále. Portál má být dle zadání obsahově samostatný, tudíž jeho obsah budou uživatelé utvářet sami. A k tomu byla také jeho uživatelská část navržena. Každá z jednotlivých aplikací, mohou-li se tak nazývat, má na webu svoji funkci. Popisem funkce článků na webu se zdržovat nebudeme, jelikož k ní bylo řečeno již mnoho.

Dále zde máme seznamku. Každý uživatel registrovaný na webu si může do seznamky vložit svůj inzerát s příloženou fotografií, a to pouze jeden, jak bylo zadáno. Seznamka umožňuje vyhledávání dle sekcí, které se dají plynule vkládat přes redakční systém a dále dle krajů. Databáze byla uzpůsobena i pro případ, že zadavatel bude chtít hledat i podle okresů. A zvažuje se i podle koníčků.

Další částí portálu jsou deníčky. Jejich funkce je v podstatě zastupovat jakési album s komentáři. Samozřejmostí je, že do nich mohou vkládat jen registrovaní uživatelé. Každé album je omezeno na určitý předem daný počet, zatím se počítá zhruba s 16-20 fotografiemi.

Diskuse se dělí na volná témata, která musí být založena administrátorem a dále na odbornickou diskusi, jejíž funkcí je zejména poradenská činnost např. od lékařů a jiných povolání lidí.

Poslední částí jsou vkládané prolinky. Jedná se o vkládání důležitých odkazů na firmy a také na různé organizace zaměřené na danou skupinu.

9. Instalace portálu na doménu a kroky k tomu nutné

V této části popíšeme jak umístit portál na doménu a kde hledat nutná nastavení. Prvním krokem by měl být výběr vhodné domény, která by měla splňovat nutné technické parametry. V informacích o doméně by měla figurovat podpora serveru apache,php a mysql. Pokud je tento požadavek splněn, můžeme přikročit k dalšímu kroku. Na serveru je nutné vytvořit celou adresářovou strukturu a přenést skripty na doménu. Dalším krokem je přenesení databáze na server. Databáze je vyexportována v jazyku sql v souboru databáze.sql, který je umístěn v kořenovém adresáři. Databáze se obvykle na server vkládá pomocí programů k tomu určených přímo na doméně. Bývá to často phpmyadmin. Pokud byla databáze úspěšně vložena na server, je nutné přistoupit k dalšímu kroku, a tím je nastavení nutných položek v souboru nastavenispojenidatabaze.php.

Obsah souboru:

```
$spojeni=mysql_connect("localhost","root")  
or die ("K databazi se nelze pripojit:".mysql_error());  
mysql_select_db('centralni2',$spojeni)or die(mysql_error());
```

Nutná nastavení:

```
mysql_connect("hostitel","uzivatel","heslo")  
mysql_select_db(název databáze)-nastaví databázi jako pracovní.
```

Po pečlivém splnění těchto kroků je celý portál připraven na spuštění.

10. Závěr

Práce na tomto projektu podle mých odhadů bude asi sahat i za rámec zpracování této bakalářské práce, protože již nyní bude potřeba portál spravovat a dále jej obohacovat. Dalším důvodem je také odladění chyb. Systém je docela rozsáhlý a upřímně pochybuji o tom, že se mi podařilo všechny chyby odstranit. Řada z nich vyplyne na povrch až uvedením do provozu, s čímž samozřejmě počítám. Práce byla celkem zajímavá hlavně v tom, že bych se této problematice chtěl v budoucnu věnovat. Mezi úskalí patřilo navržení databáze, která prošla v průběhu vývoje neustálými změnami a s ní se samozřejmě měnil i celý systém. Mezi další úskalí patřila práce s obrázky. Funkce na jejich zpracování mi popravdě zabrala hodně času. Jinak s tvorbou redakčního systému problémy nebyly, až na to, že chvílemi se práce zdála rutinní a zdlouhavá. Odstraňování chyb ze skriptů občas pěkně potrápilo. Práce se stala zajímavější po započetí tvorby vzhledu portálu a použití šablon. Hlavně tím, že začalo být vidět funkci celého systému z uživatelského pohledu a práce vypadala celistvěji. I když zpočátku byly problémy se zvyknutím si na šablonovací systém jako takový, na jeho funkci a použití, a také s vyladěním chyb, aby vše fungovalo tak, jak má. Ale znatelná výhoda oproti normálnímu použití při změně vzhledu zůstala dostatečnou motivací pro použití této technologie. Mrzí mne jen, že se mi nepovedlo v této práci některé věci zdůraznit, protože nebyly předmětem práce a tudíž nemělo cenu se o nich zmiňovat a zdůrazňovat je.

Co se týče projektu jako takového, myslím, že vzhledem k tomu, že se jedná o projekt vymyšlený z pole zájmového a ne za účelem podnikání, měl by problém s financováním a vzhledem k náročnosti by asi pravděpodobně narazil na finanční problémy a vůbec nevznikl, ať již z důvodu, že by se nesehnali potřebné finance nebo by byly problémy s jeho realizací a zanikl by.

Na druhou stranu jsem se na projektu podílel nebo spíš budu podílet dál velice rád, buď jako nadšenec a člověk, co se na projektu může něco naučit nebo pro dobrou věc a pro vyplnění mezery na poli internetu jako komunikační technologie pro tuto zájmovou skupinu.

Závěrem bych si přál, aby se mě a zadavatelce společnými silami povedlo tento projekt zdárně rozběhnout a sehnalo se dost lidí, respektive nadšenců pro jeho počáteční provoz a aby cílová skupina, pro kterou je určen, si skutečně našla cestu k jeho využívání.

11. Přílohy

11.1 Popis funkce HTML TEMPLATE:

System zpočátku působí složitě, ale nakonec si člověk na tento systém zvykne. Jeho jádro tvoří docela obsáhlý soubor napsaný v php jménem htmltmpl.php, který rozděluje aplikaci a html kód. Do každého skriptu je třeba tento soubor includovat, aby zajistil prostřednictvím předdefinování funkcí naplnění šablony. Příkazy uvádím níže:

HTML TEMPLATE Použité příkazy+některé funkce:

<TMPL_VAR NAME="`PARAMETER_NAME`">

\$template->param(PARAMETER_NAME => ``VALUE")

<TMPL_LOOP NAME="`LOOP_NAME`"> </TMPL_LOOP>

Slouží na výpis pole na stránku.

<TMPL_INCLUDE NAME="`filename.tmpl`">

Jedná se o vkládání a vrstvení šablon období include v php.

Logické funkce:

<TMPL_IF NAME="`CONTROL_PARAMETER_NAME`">

<TMPL_ELSE>

</TMPL_IF>

<TMPL_IF BOOL> Výraz při pravdě**<TMPL_ELSE>** Výraz při nepravdě**</TMPL_IF>**

Php kód plní vyhodnocující položku:

\$template->param(BOOL => 1) - výraz pro vyhodnocení.

Tento výraz je opakem logiky u if:

<TMPL_UNLESS NAME="`CONTROL_PARAMETER_NAME`"> </TMPL_UNLESS>

Příklad:

<TMPL_UNLESS BOOL> Výraz při nepravdě **<TMPL_ELSE>** Výraz při pravdě
</TMPL_UNLESS>

Metody které sou nutné pro plynulí chod:

new()-voláním New() vytvoříte nový šablonový objekt

syntaxe:

```
$template = HTML::Template->new( filename => 'file.tmpl', option => 'value' );
```

Param()

param může být přivolán několika cestami:

1) Pro vrácení seznamu parametrů v šabloně :

```
@parameter_names = $self->param();
```

2) Pro vrácení hodnoty se vloží do param:

```
$hodnota = $samo -- >param('PARAM');
```

clear_params()

Vymaže všechny parametry

output()

output() vrací finální výsledek šablony.

Příklad krátkého kódu a vysvětlení funkce na seznamce

Template

```
<TMPL_IF BOOL> //podmínka za které se vypíše daný inzerát
<h2>Váš inzerát</h2>
<TMPL_LOOP vypis2> // výpis pole hodnot
<div class="inzeratseznamka">
<div class="inzeratpomocramec">
<div class="inzeratbunka">
Nick:<TMPL_VAR nick> // výpis hodnoty
</div>
<div class="inzeratvnitniramec">
<TMPL_VAR hlaska> > // výpis hodnoty
</div>
<div class="inzeratbunka">
Datum: <TMPL_VAR datum>> // výpis hodnoty
</div>
</div>
<div class="inzerattelo">
<a href="<TMPL_VAR fotkav">">" border="0"align="left"></a>
<TMPL_VAR text> > // výpis hodnoty
</div>
<div class="inzeratpomocramec">
<div class="inzeratbunka">
email:<A HREF="mailto:<TMPL_VAR email">"><TMPL_VAR email></A>
</div>
<div class="inzeratbunka">
věk:<TMPL_VAR vek> > // výpis hodnoty
</div>
<div class="inzeratbunka">
Icq:<TMPL_VAR icq> > // výpis hodnoty
</div>
<div class="inzeratvnitniramecco">
</div>
```

```

</div>
</div>
<h3>Máte právo zadat jen jeden inzerát!!<a href="seznamka.php?smacknuti=1">Zpět</a></h3>
<form action="seznamka.php?smacknuti=2" method="post" enctype="multipart/form-data">
<input type="hidden" name="idsmazat" value="<TMPL_VAR id>">
<input type="submit" name="smazat" value="smazat">
</form>
</TMPL_LOOP> //konec výpisu daného pole
</TMPL_IF> //ukončení příkazu IF

```

Naplnění Šablony jazykem php

```

$BOOL=1;
$stproc->set("BOOL", $BOOL); //naplnění podmínky if(1-true 0-false)
    $a=1;
    $resuld = mysql_query("SELECT * FROM info_prihlaseni //sql dotaz na výpis inzerátu
    where id=".$_SESSION['id'].'"
    ORDER BY info_prihlaseni.id limit 1");//
    echo "SELECT * FROM info_prihlaseni
    where id=".$_SESSION['id'].'"
    ORDER BY info_prihlaseni.id limit 1";

$Vypiss = array(); //vytvoření pole
while ($radek2=mysql_fetch_array($resuld))
{

if($radek2['schvaleno']==1){$hlaska="Inzerát čeká na schválení!";}
if($radek2['schvaleno']==2){$hlaska="Inzerát je schválen!";}
$Vypis['id']=$radek2['id'];
$Vypis['datum'] = $radek2['datum'];
$Vypis['nick'] = $radek2['nick'];
$Vypis['vek'] = $radek2['vek'];
$Vypis['fotkam'] = $radek2['fotkam'];
$Vypis['fotkav']=$radek2['fotkav'];
$Vypis['text'] = $radek2['text'];
$Vypis['email'] = $radek2['email'];
$Vypis['icq'] = $radek2['icq'];
$Vypis['hlaska'] = $hlaska;

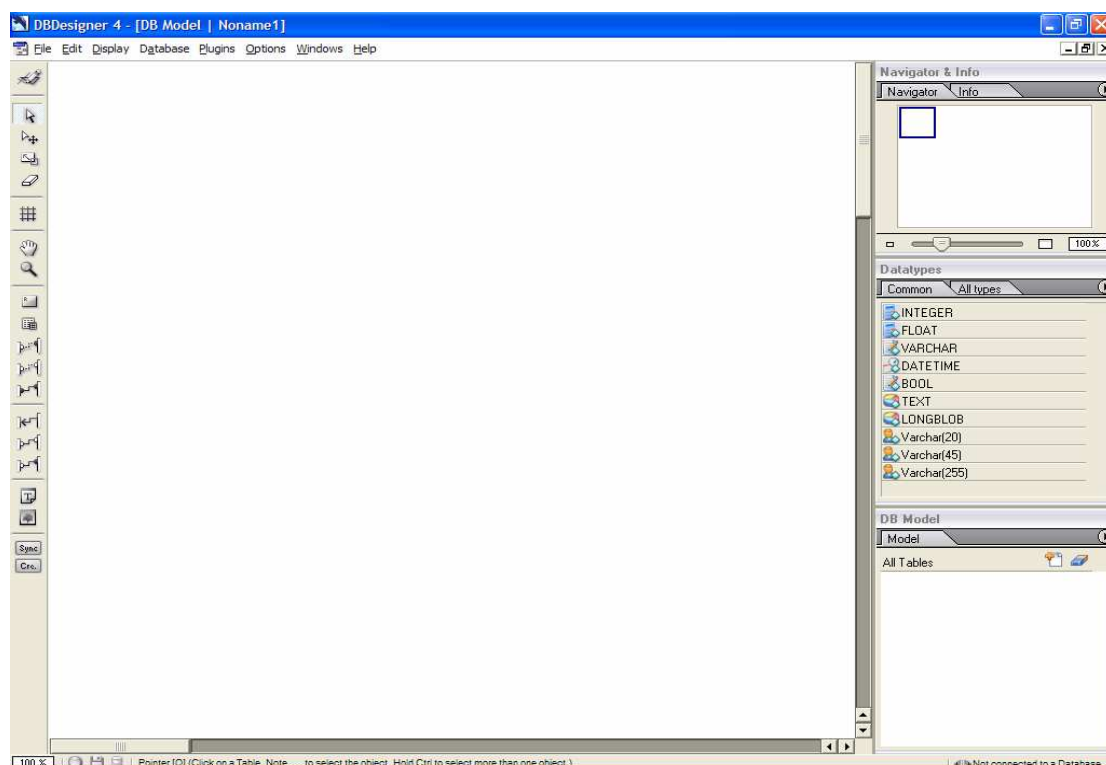
array_push($Vypiss, $Vypis); //převodění pole
$stproc->set("vypis2", $Vypiss);// odeslání proměnné na šablonu.

```

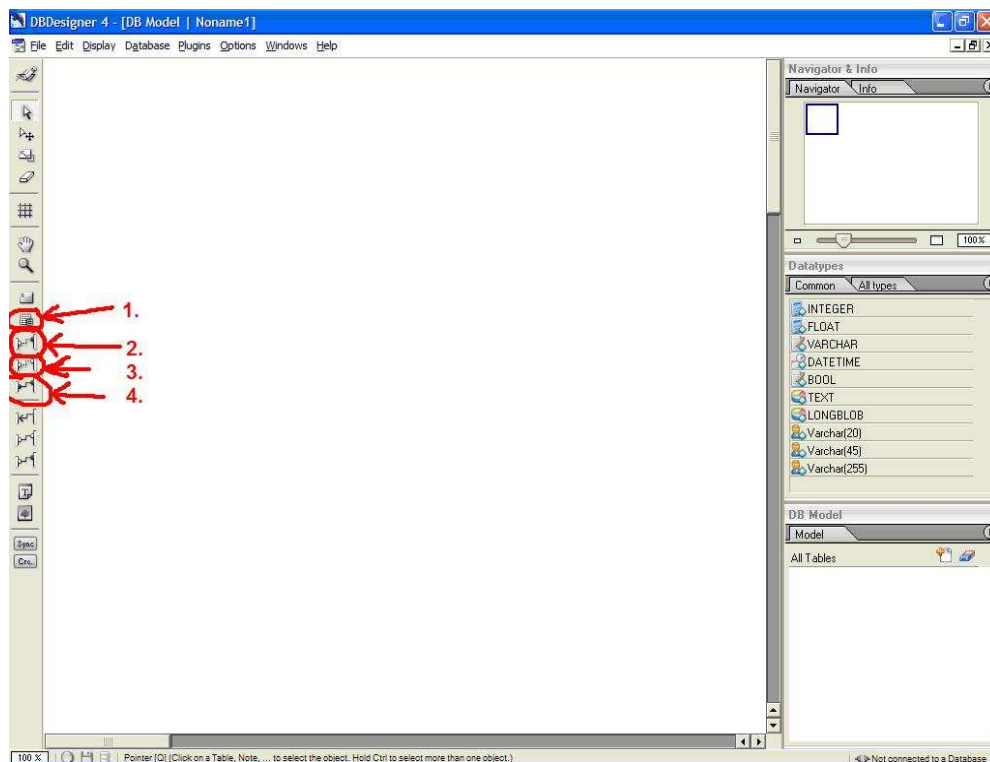
11.2 Základní popis a ukázka modelování v programu DB designer

Vývojové prostředí programu Db designer:

Je velice přívětivé a musím uznat že je na profesionální úrovni. Bez problémů se vyrovná placeným programům. A modelování databází je ideální.



Vyznačení důležitých položek




1. Vyznačuje položku na tvorbu tabulky


Pomocí této ikonky snadno vytvoříte tabulku. Dvojklikem na ni si jí můžete pojmenovat a změnit její kompletní vlastnosti, název primárního klíče a zadat všechny položky tabulky, včetně jejich datových typů (varchar, integer a další). Každá relace se vytvoří kliknutím na cílovou tabulku a zbytek program zařídí automaticky za Vás, což je snadné příjemné a rychlé, usnadňuje to práci a zvyšuje přehlednost.

2. Číslo v pořadí 2,3,4.

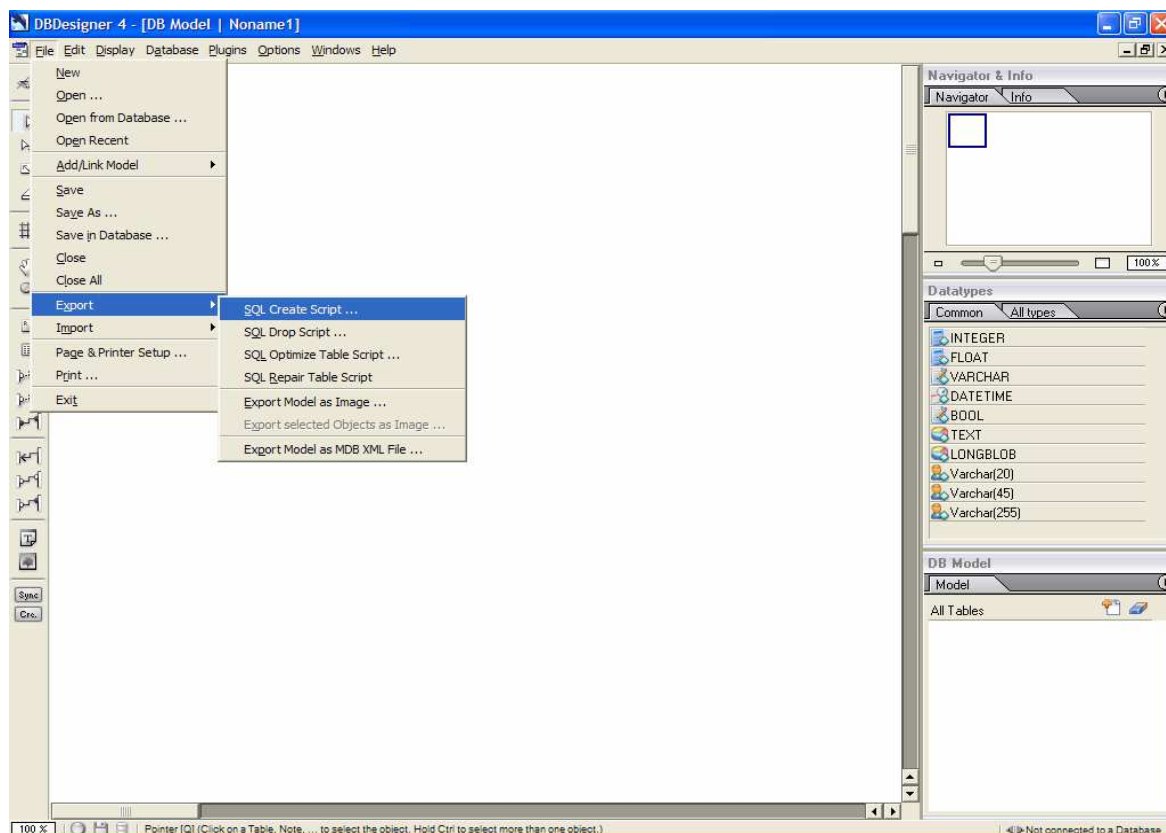
Vyznačují typy jednotlivých databázových relací v pořadí uvedeném 2 relaci 1:n, 3 relaci 1:1 a 4 relaci n:m. Co jednotlivé relace znamenají proberu později.

Jednotlivé značky odpovídající značení relací v DB designeru

 -tímto označením značí DB designer relaci 1:1

 -tímto označením značí DB designer relaci 1:n

Pokud se týká relace M:N, je v db designeru značena stejně jako relace 1:n s tím rozdílem, že entita je vytvořena automaticky a standardně umístěna mezi dvě spojované entity.



Export

Pomocí tohoto programu se dá i široce exportovat, jak je uvedeno na obrázku. Export je možný do různých formátů. Ty nejpodstatnější jsou export do mysql a také v neposlední řadě si můžete vyexportovat databázi jako obrázek, což je velice užitečné pro potřeby například tohoto textu.

