UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Informační systém základní školy

Václav Mareček

Bakalářská práce
2013

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2011/2012**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Václav Mareček**
Osobní číslo:	**I08108**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Informační systém základní školy**
Zadávající katedra:	**Katedra informačních technologií**

Zásady pro vypracování:

Cílem práce je návrh a realizace webové aplikace pro základní školu. Aplikace bude vyvíjena s důrazem na využitelnost menší základní školou, při tvorbě budou využity technologie HTML5, PHP, CSS, XML, Javascript a relační databáze.
Aplikace musí umožnit například:
- evidenci žáků a pracovníků školy
- sledování (a omlouvání) docházky
- evidenci prospěchu, kázeňských opatření
- sledování plnění domácích úkolů
- informování o školních akcích
- generování sestav pro pravidelné výkazy

Teoretická část se bude zabývat problematikou ochrany osobních dat včetně fotografií, s kterými uvedený systém bude pracovat. Dále bude diskutována problematika zabezpečení dat ve webových aplikacích.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. SCHAFER, S., M.: HTML, XHTML a CSS ? Bible pro tvorbu www stránek. 4.vyd. Grada, 2009. ISBN 978-80-247-2850-6
2. GUTMAS, A., RETHANS, D., BAKKEN, S., S.: Mistrovství v PHP 5. Computer Press, 2007. ISBN 978-80-251-1519-0
3. HARRINGTON, J., L.: SQL clearly explained. 3.vyd. Elsevier, 2010. ISBN 978-0-12-375697-8
4. HERNANDEZ, J., M., Viescas, L., J.: Myslíme v jazyku SQL: Tvorba dotazů. Grada, 2004. ISBN 978-80-247-0899-7

Vedoucí bakalářské práce: **prof. Ing. Karel Šotek, CSc.**
Katedra softwarových technologií

Datum zadání bakalářské práce: **16. prosince 2011**
Termín odevzdání bakalářské práce: **11. května 2012**

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan, Ph.D.
vedoucí katedry

V Pardubicích dne 30. března 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 10.5 2013

Václav Mareček

Poděkování

Chtěl bych na tomto místě poděkovat vedoucímu mé bakalářské práce prof. Ing. Karlu Šotkovi, CSc. hlavně za umožnění pracovat na této bakalářské práci a za jeho rady, připomínky a za čas, který mi byl ochotný věnovat. Také bych chtěl poděkovat mé rodině a přátelům za podporu v průběhu celého studia, zejména v posledním ročníku.

Anotace

V aplikační části se věnuji návrhu a implementaci vlastního informačního systému se zaměřením na komunikaci s rodiči žáků, v teoretické části se zabývám problematikou ochrany osobních údajů v informačních systémech.

Klíčová slova

Informační systém, základní škola, PHP, Oracle, WWW, ochrana osobních údajů

Title

Information system for primary school

Annotation

The application part is devoted to design and implement their IT systems with a focus on communication with parents. The theoretical part deals with the issue of privacy in information systems.

Keywords

Information system, primary school, PHP, Oracle, WWW, personal data protection

Obsah

Seznam zkratek

BFILE	binary file
BLOB	Binary large object
CLOB	Character large object
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IS	Informační systém
MD5	Message-Digest
MSSQL	Microsoft SQL Server
OOP	Objektově orientované programování
PC	Osobní počítač
PHP	Hypertext Preprocessor
PL/SQL	Procedural Language/Structured Query Language
SHA1	Secure Hash Algorithm
SQL	Structured Query Language
UIV	Ústav pro informace ve vzdělávání
WWW	World Wide Web
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XSS	Cross-site scripting

Seznam obrázků

Seznam tabulek

1 Úvod

Pro vývoj složitější a komplexnější aplikace se můžeme vydat různými směry, počínaje zvolením programovacího jazyka či databázového systému. Je potřeba zvážit možnosti, které daný jazyk nabízí, jeho obtížnost a zda je dobře dokumentován. To stejné platí pro databázový systém. Dále je dobré promyslet a vhodně zvolit vývojové prostředí, ve kterém bude aplikace vyvíjena.

V dnešní době se rychle expandující internet dostává téměř do každé domácnosti. S rostoucím přístupem k internetu přibývá mnoho služeb provozovaných přes internet prostřednictvím webových aplikací, mezi které patří elektronické bankovnictví, sociální sítě, veškerá komunikace i různé firemní informační systémy. V těchto aplikacích se ale mnohdy vyskytují citlivé údaje či firemní tajemství, a proto je vhodné myslet na jejich bezpečnost a své aplikace řádně zabezpečit.

Zájem o informační systémy stoupá z mnoha důvodů. Jedním z hlavních příčin je zjednodušení, urychlení a zefektivnění naší práce. Většina systémů obsahuje přednastavené formuláře, které stačí pouze vyplnit a data jsou řádně utříděna a uložena do databází, které je jednoduché zobrazovat, upravovat či zálohovat. Další výhodou je možnost rozdělit přístup různým uživatelům pouze k těm datům, které potřebují a přesto uchovávat data na jednom místě. V neposlední řadě mají informační systémy obrovskou výhodu, že je možné s nimi pracovat ze zaměstnání, domova i zahraničí.

Používání školních informačních systémů je na vysokých školách již běžné, ale na středních a základních školách jen výjimečné. Přestože informační systémy přináší velmi mnoho výhod, některým školám brání většinou nedostatek financí nebo neochota či strach ze změny dosavadního a zaběhlého systému, zpravidla v papírové formě, se k takové práci přeorientovat.

1.1 Cíle práce

Cílem teoretické části je problematika ochrany osobních údajů z pohledu zákona. Zaměříme se na ochranu osobních údajů jako takových, kde rozeberme, co si pod osobním údajem můžeme představit, jaké právní předpisy tuto ochranu upravují a jaké povinnosti a odpovědnosti vyplývají subjektům při provozování informačních systémů, které nakládají s osobními údaji. Dále budou popsány různé druhy útoku na webové aplikace a možnosti ochrany proti tomu.

Cílem praktické části je vytvořit skutečný provozuschopný informační systém pro konkrétní základní školu v Bernarticích u Trutnova, který bude splňovat požadavky základní školy a bude vycházet ze znalostí a nápadů z teoretické části této práce. Dále budou rozebrány všechny použité technologie včetně toho jak je použít. V bakalářské práci bude obsažena kapitola o vývojových prostředích, bude zde popsána struktura aplikace, její vzhled, rozdělení uživatelů a databázové schéma s popisem tabulek.

1.2 Struktura dokumentu

1. Úvod

   Úvodní část představuje krátké obeznámení s tématem této práce, uvádí do dané problematiky, popisuje cíle práce s kompletní strukturou dokumentu, která slouží pro lepší orientaci.

2. Ochrana osobních údajů v informačních systémech

   V této části si představíme problematiku ochrany osobních údajů. Objasníme si základní práva a povinnosti, které se při práci s osobními údaji musí dodržovat. Zmíníme se i o problematice ochrany fotografií.

3. Dostupné informační systémy na trhu

   Tato část práce se zabývá alternativními systémy, které se využívají v praxi.

4. Zabezpečení dat webových aplikací

   Pojednává o několika nejčastějších druzích útoků na webové aplikace, ukážeme si, jak taková napadení mohou vypadat a probíhat, a nakonec si popíšeme i možnosti obrany proti těmto útokům.

5. Použité technologie

   V této části se zmíníme o technologiích, které byly použity k vytvoření aplikace, o jejich vývoji a vývojových prostředích.

6. Návrh a vývoj aplikace IS

   Představení vytvořené aplikace, její architektura a rozdělení uživatelských rolí.

7. Závěr

   Shrnutí výsledků této práce a všech cílů.

2 Ochrana osobních údajů

2.1 Problém ochrany osobních údajů

Ochrana osobních údajů je zakotvena v těchto zákonech:

- Občanský zákoník- slouží k právní úpravě ochrany osobnosti (viz §11)
- Obchodní zákoník – upravuje ochranu osobních údajů právnických osob. Jedná se o ustanovení, porušení nebo ohrožení práva na ochranu obchodního tajemství, kde podnikateli přísluší právní ochrana jako při nekalí soutěži, upravená v ustanoveních § 53 až §55.
- Zákon o státní a statistické službě – upravují ochranu údajů získané od jednotlivých osob Českým statistickým úřadem či jinými státními orgány vykonávající statistickou službu.
- Zákon o péči o zdraví lidu – stanoví zdravotnickým pracovníkům, mimo jiné povinnosti, povinnost zachovávat mlčenlivost o skutečnostech, o nichž se dozvěděli v souvislosti s prováděním svého povolání. Výjimkou je, když skutečnosti sdělují se souhlasem ošetřované osoby nebo této povinnosti byli zbaveni nadřízeným orgánem v závažném státním zájmu.
- Zákon o bankách – upravuje ochranu bankovního tajemství, které se vztahuje na všechny bankovní obchody, peněžní služby, stavy na účtech a depozit. Za porušení bankovního tajemství se nepovažuje výměna údajů mezi Českou národní bankou, orgány bankovního dohledu a obdobných institucí jiných států za předpokladu, že předmětem výměny jsou informace o subjektech, které působí nebo se chystají působit na území patřičného státu.
- Autorský zákon – obsahuje úpravu institutu ochrany osobních údajů dle § 7.
- Zákon o advokacii – upravuje povinnost mlčenlivosti, která se vztahuje jak na advokáta a na advokátní koncipienty, tak i na jeho zaměstnance.
- Trestní zákoník a zákon o přestupcích – upravuje v § 230, pro trestný čin neoprávněný přístup k počítačovému systému a nosiči informací a dále v § 180 je definován trestný čin neoprávněného nakládání s osobními údaji.
- Zákon o střelných zbraních a střelivu – usměrňuje rozmezí osobních údajů na žádosti o vydání zbrojního průkazu nebo zbrojní licence a osobní údaje osoby, která v rozsahu svého pracovního zařazení zabezpečuje vykonávání služebních povinností.
- Zákon o dočasné ochraně cizinců – upravuje metodu uchovávání údajů v evidencích používaných Policií České republiky a zpravodajskými službami, obzvláště v postupu předávání osobních údajů do jiných států.
- Zákon o Vojenském zpravodajství – upravuje zabezpečení ochrany údajů v kompetenci úřadu Ministerstva obrany obsažených v evidencích před vyzrazením, zneužitím, poškozením nebo zničením.
- Zákon o Bezpečnostní informační službě – definuje povinnost zabezpečit ochranu osobních údajů, ostatních údajů a zvláště utajovaných informací, které jsou

obsaženy v evidencích, před vyzrazením, zneužitím poškozením ztrátou a odcizením.

- Zákon o Vězeňské službě a justiční stráži České republiky – upravuje vedení evidence osob ve výkonu vazby a trestu odnětí svobody. Pro zpracování osobních údajů v této evidenci, není potřeba souhlasu osoby, které se údaje týkají a vězeňská služba nemá povinnost informovat osobu o obsahu své evidence. Tyto údaje Vězeňská služba poskytuje pouze činným orgánům v trestním řízení, jako soudům, statním zastupitelstvím, správním orgánům a Rejstříku trestů, pokud je potřebují pro svou činnost. [1]

2.2 Práva a povinnosti správce

2.2.1 Povinnosti správce

Správce musí ověřovat, zda jsou osobní údaje stále pravdivé a přesné, případně provádět aktualizace. Zjistí-li, že tomu tak není, je povinen takové informace bez odkladu blokovat, opravit nebo doplnit. Není-li to možné, musí je zlikvidovat. Vhodné je zavázat subjekt údajů, aby ohlašoval správci veškeré změny nebo aby získal přístup ke zpracovaným údajům a mohl provádět změny sám.

Dále zákon stanoví, že osobní údaje můžeme uchovávat pouze po dobu, která je nezbytná k účelu jejich zpracování. Po uplynutí této doby lze údaje použít pro statistiky, archivy nebo vědecké účely, ale osobní údaje je nutné anonymizovat.

Správce nemůže sdružovat osobní údaje, které byly získány k rozdílným účelům, to znamená, že nemůže sloučit data získaná různými zadavateli. [1]

2.2.2 Oprávnění správce při práci s osobními údaji

Správce může zpravovat veškeré osobní údaje vždy pouze se souhlasem subjektů. V opačném případě, bez souhlasu, se musí řídit pravidly, která jsou stanovena zákonem, a dbát přitom na ochranu soukromého a osobního života subjektu.

Správce nebo zpracovatel nesmí osobní údaje dále zpracovávat, pokud subjekt vyjádřil nesouhlas. Tento nesouhlas je nutné vystavit písemně.

Správce může mít se zpracovatelem uzavřenou smlouvu o zpracování osobních údajů. Musí mít však písemnou formu, musí být stanoven rozsah, účel a doba, na kterou je smlouva uzavřena, a dále by měly být dány dostatečné záruky o technickém a organizačním zabezpečení. Při nesplnění těchto podmínek je taková smlouva neplatná. [1]

2.3 Ochrana práv subjektů údajů

Jestliže subjekt údajů zjistí nebo se domnívá, že správce či vybraný zpracovatel provádí zpracování jeho osobních údajů v rozporu s ochranou soukromého a osobního života nebo v rozporu se zákonem, může žádat o vysvětlení a požadovat odstranění

vzniklého stavu. Může se jednat o blokování, provedení opravy, doplnění nebo likvidaci osobních údajů. Správce je následně povinen informovat subjekt o provedených změnách.

Je-li žádost subjektu údajů shledána oprávněnou, správce nebo zpracovatel závadný stav neprodleně odstraní. Pokud je správce povinen zpracovávat osobní údaje na základě zákona nebo by tím mohla být způsobená újma třetí osobě, právo na zablokování či likvidaci nemůže subjekt požadovat. [1]

2.4 Ochrana Fotografií

Tato část je věnována problematice uveřejňování fotografií na internetu. Odpovědi nalezneme v občanském zákoníku, neboť se jedná svojí povahou o obecnou ochranu osobnosti, která je upravená v § 11 – 16 OZ. Na tuto obecnou ochranu osobnosti potom navazuje ochrana osobních údajů jako určitá speciální regulace směřující k ochraně lidského soukromí, kterou upravuje zákon č. 101/2000 Sb., o ochraně osobních údajů (ZOOÚ).

Jestliže se jedná o fotografie z hromadných akcí, kde není jasné, kdo na fotografii je, pak nebude podléhat režimu ZOOÚ. Fotografie konkrétní osoby, například nejlepší pracovník či ředitel základní školy, u nichž uvádíme jméno, anebo je možné ho jednoduše dohledat z jiných záznamů, pak uveřejňování takovému režimu bude plně podléhat. V tomto případě se jedná o citlivé údaje a musíme mít výslovný souhlas fotografované osoby. Dále subjekt musíme informovat o tom, na jaké období a za jakým účelem souhlas poskytuje. Souhlas musíme být schopni prokázat po celou dobu zpracování, proto je vhodné si k tomu účelu vytvořit písemný dokument. [2]

2.5 Zabezpečení osobních údajů

Při práci s osobními i citlivými údaji musí správce a zpracovatel přijmout opatření, aby nemohlo dojít k neoprávněnému přístupu k osobním údajům, k jejich změně, zničení či ztrátě nebo jinému neoprávněnému zpracování. S tím souvisí i to, že všichni zaměstnanci, kteří pracují s osobními daty a osobními údaji nebo s nimi přicházejí do styku, musí zachovávat mlčenlivost. Tato povinnost trvá i po skončení či po ukončení příslušné práce. Dále je správce povinen provést likvidaci všech osobních údajů, jestliže pominul účel nebo jestliže o to subjekt požádal. [1]

3 Dostupné informační systémy na trhu

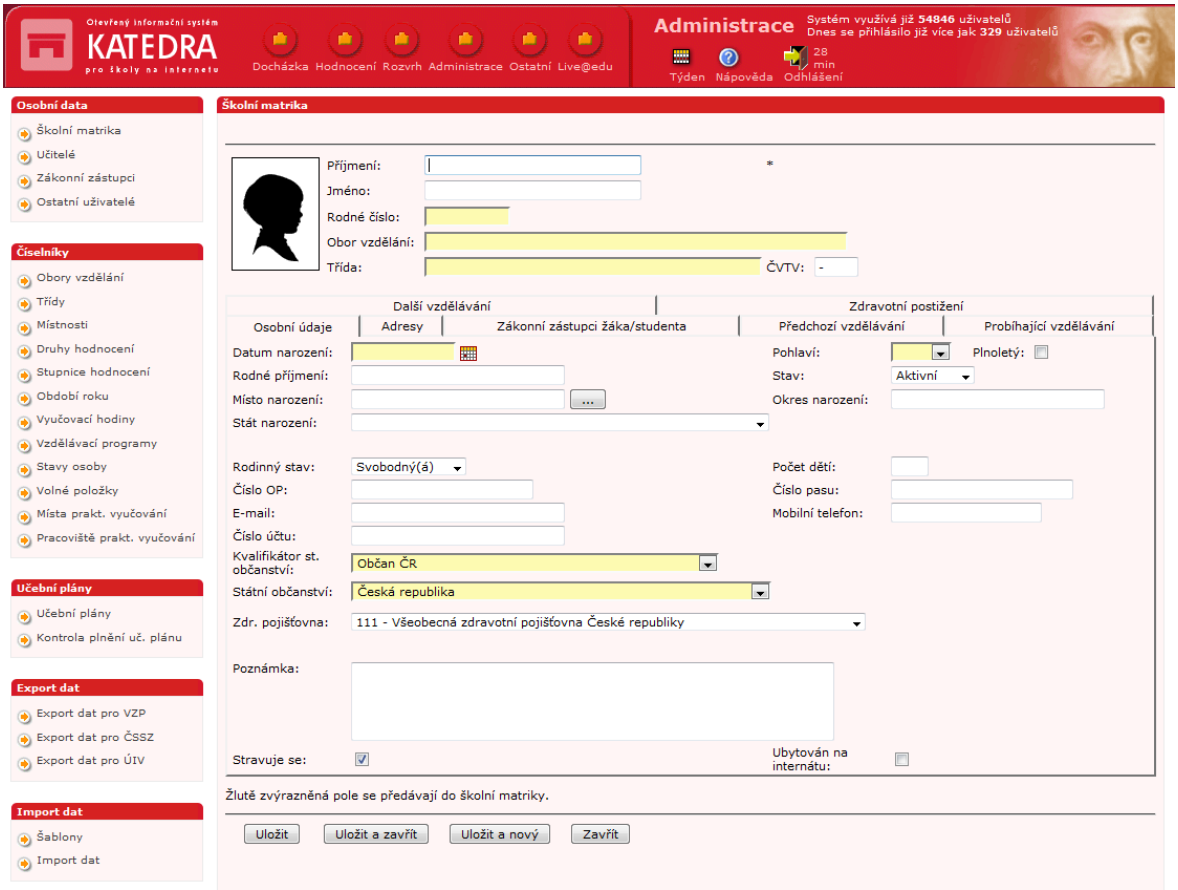
3.1 Škola OnLine

V dnešní době jedním z nejrozšířenějších informačních systému je Škola Online. Jedná se o moderní IS umožňující rychlé a efektivní zpracování školní agendy. Jde o webovou aplikaci, která je dostupná 24 hodin denně za pomocí internetu a běžného internetového prohlížeče.

Škola OnLine obsahuje několik dílčích systémů. Pro naše potřeby budou zmíněny pouze dva základní a to Katedra a Žákovská.

3.1.1 Katedra

Systém Katedra je určen pouze pro pracovníky školy, jako jsou administrátoři, ředitelé škol a učitelé. Slouží ke kompletní správě školní agendy a vedení školní matriky. Samotná aplikace se skládá z několika modulů (Obrázek 1).[20]



Obrázek 1 - Škola OnLine - Katedra

(Zdroj:Vlastní)

3.1.2 Moduly Katedry

Docházka

Tento modul slouží ke kompletní správě informací o docházce studentů do školy. Součástí je elektronická třídní kniha. Zapisovat docházku mohou pouze třídní učitelé nebo jejich zástupci. Ostatní učitelé mohou absence zapisovat pouze na konkrétních hodinách, ve kterých vyučují. Třídní kniha disponuje měsíčními, týdenními i denními mezisoučty absencí u jednotlivých studentů, včetně jejich tisknutí. [20]

Hodnocení

Modul hodnocení umožňuje zadávat učitelům do systému známky ze svých předmětů a pouze těm žákům, které učí. Hodnocení je možné zadávat třemi typy, a to pomocí procent, bodů či známek. Modul nabízí třídním učitelům nebo jejich zástupcům i uchovávání poznámek, důtek, pochval a napomenutí u jednotlivých studentů. V neposlední řadě poskytuje vysokoúrovňový pohled na data v rámci celé školy. Díky němu je schopen zobrazit průměrné známky ve všech třídách na škole, problémové a neklasifikované žáky.[20]

Výuka

Jednoduchý modul, který nabízí učitelům podporu výuky studentů na škole. Učitelé mohou pomocí tohoto modulu přiřadit k jednotlivým předmětům či hodinám různé podpůrné materiály, soubory, odkazy nebo testy pro zpestření výuky.[20]

Rozvrh

Tento modul je základním prvkem celé aplikace Katedra a umožňuje administrátorovi vytvářet a modifikovat rozvrhy žáků, tříd i učitelů. Při vytváření rozvrhu modul nabízí možnost rozdělení třídy na menší části (např. anglický a německý jazyk). Vytvořené rozvrhy je dále možné vypisovat i tisknout. Modul Rozvrh disponuje i funkcemi pro práci se suplováním. V případě změn v rozvrhu, jako je změna učitelů, třídy nebo přesun hodin, žáci hned vidí aktuálně upravený rozvrh v aplikaci Žákovská.[20]

Administrace

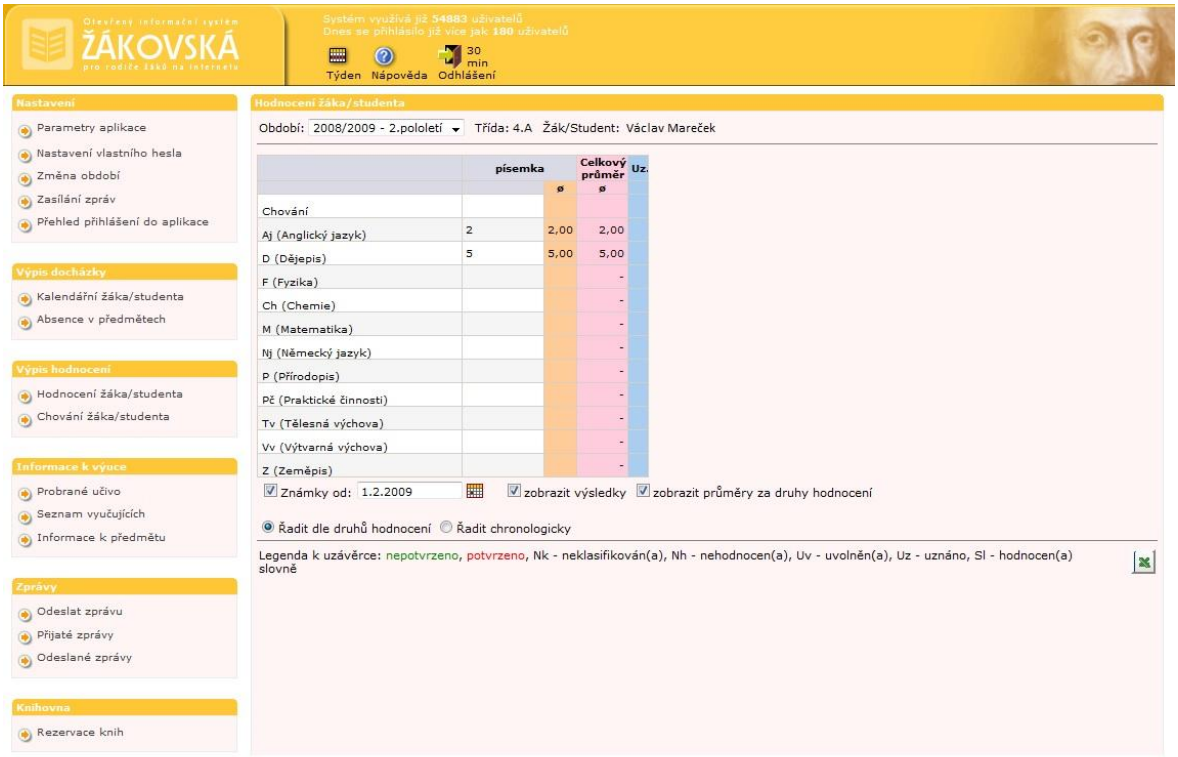
Důležitý modul, který umožňuje administrátorovi veškerou správu školy. Může spravovat informace o všech studentech, učitelích, rodičích a zaměstnancích školy uložených v systému. Dále může upravovat veškeré číselníky místností, tříd, hodnocení a vyučovacích hodin. V neposlední řadě nabízí široké možnosti ve správě přístupových práv jednotlivců i celých skupin.[20]

Ostatní

Modul Ostatní je balík složený z několika menších modulů. Jeden z menších modulů se stará o stravovací systém. Slouží k prohlížení, objednávání a odhlašování obědů. Dále nabízí rozesílání zpráv studentům a rodičům, správu školní knihovny, generování a tisk vysvědčení, evidenci veškerého školního majetku, vytváření statistik, správu přijímacích, závěrečných a maturitních zkoušek, převod dat do dalšího školního roku a mnoho dalších funkcí.[20]

3.1.3 Žákovská

Jedná se o webovou aplikaci určenou studentům školy a jejich rodičům nebo zákonným zástupcům studentů (Obrázek 2). Žákovská je přístupná pomocí internetu, pomocí běžného internetového prohlížeče a podává všem uživatelům aktuální informace denně.[22]



Obrázek 2 - Škola OnLine - Žákovská

(Zdroj: Vlastní)

3.1.4 Funkce a využití aplikace Žákovská

Aby aplikaci uživatelé mohli využívat, musejí se nejprve zaregistrovat. Každý student i zákonný zástupce obdrží jedinečný číselný identifikátor, který jasně definuje přístupová práva a pomocí tohoto identifikátoru se i uživatelé zaregistrují. Přihlášený uživatel může následně využívat všech funkcí, na které má přístupová práva.[21]

Jednou ze základních funkcí aplikace Žákovská je možnost průběžné kontroly hodnocení studenta ze všech předmětů, které studuje, a pravidelnost docházky do školy. Tyto informace mohou získat samotní studenti nebo rodiče, kteří jsou svázáni s danými studenty. [22]

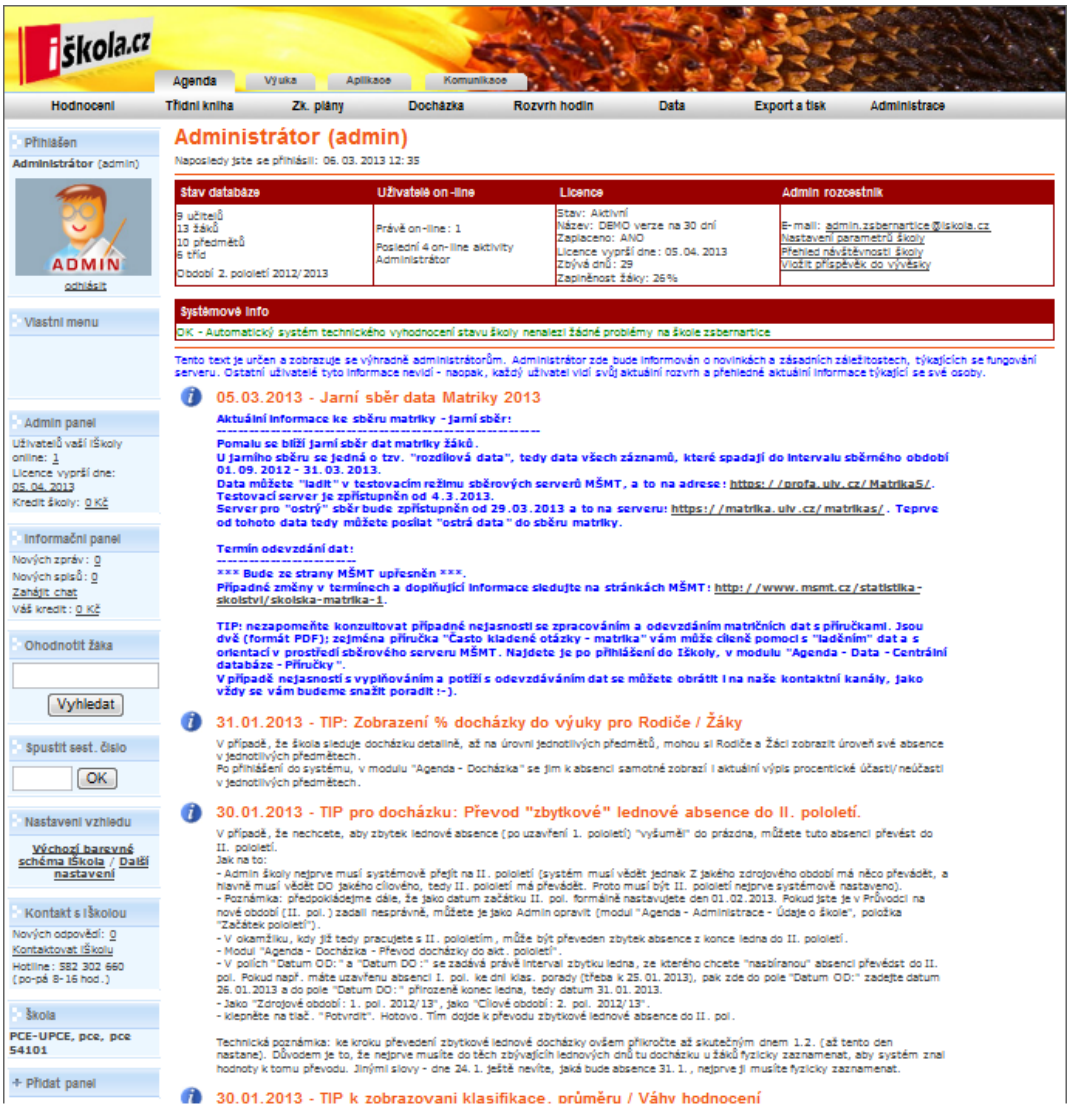
Dále aplikace nabízí zobrazení aktuálního rozvrhu, jednotlivých studentů, učeben a učitelů. V případě změn v rozvrhu, suplování nebo odpadnutí hodiny se změny uživatelům okamžitě zobrazí. [22]

Významnou funkcí je obousměrná komunikace. V případě potřeby mohou rodiče nebo žáci prostřednictvím aplikace poslat zprávu konkrétnímu učiteli nebo učitelce na škole, která se okamžitě po odeslání zobrazí příjemci v aplikaci Katedra. V opačném případě může škola nebo učitelé informovat studenty a rodiče o plánovaných školních akcích, o nepřítomnosti dítěte, kázeňská opatření nebo neomluvené hodiny. Tyto informace je možné v případě zájmu rodičů zasílat i prostřednictvím SMS.[22]

Aplikace Žákovská nabízí i mnoho menších funkcí. V případě, že škola disponuje vlastní knihovnou, mohou si žáci jednoduše zobrazit seznam dostupných knih a případně si je rezervovat. Podobně pak pracují i funkce pro školní jídelnu, kde si studenti mohou přihlašovat nebo odhlašovat obědy.[22]

3.2 iŠkola.cz

Dalším z dostupných informačních systémů na trhu je iŠkola.cz (Obrázek 3). Jedná se o webový IS a je určen pro základní i střední školy. Zaměřuje se především na školní agendu, elektronickou zprávu školní výuky a pomáhá v komunikaci škole s žáky i rodiči. iŠkola.cz nabízí velké množství funkcí, které zajistí téměř všechny potřeby škol.[21]



Obrázek 3 - Úvodní stránka IS iŠkola.cz

(Zdroj: Vlastní)

3.2.1 Moduly

Individuální funkce IS iŠkola.cz jsou rozděleny do jednotlivých modulů. Každý z nich přidává novou funkci do celého systému.

Hodnocení a poznámky

Tento modul zajišťuje evidenci známek a poznámek. Učitelé mohou zapisovat známky jednotlivým studentům ve třídě, kterou vyučují, ale pouze u těch předmětů, které učí. Modul umožňuje učitelům zadávat nejen známky 1-5 , ale i pomocné + (plus) a – (mínus), hodnocení pomocí bodů či procentuálně. Studenti i jejich rodiče mají možnost si své známky a poznámky ihned prohlížet pomocí svého uživatelského jména.[21]

Rozvrh hodin a suplování

K vytváření rozvrhů a přidávání jakýchkoliv změn slouží právě tento modul. Umožňuje všem oprávněným uživatelům sledovat aktuální rozvrhy a v případě změny učebny, učitele či odpadnutí hodiny jsou změny okamžitě promítnuty všem uživatelům.[21]

Docházka

Modul evidující docházku a absenci všech žáků školy. Tento modul nám nabízí dva režimy pro evidenci docházky a záleží na administrátorovi, který z nich nastaví. První je detailní a je provázaný s rozvrhem až na konkrétní hodiny a druhý je stručnější, který eviduje pouze počet absencí v rámci jednoho dne.[21]

Testy on-line

Jedná se o jednoduchou aplikaci, kde si každý učitel může vytvořit test pomocí prostého návrháře, který pak jednotliví žáci vyplňují pomocí PC. Otázky je možné obodovat a díky tomu se mohou testy samy vyhodnocovat a žáci vidí okamžitě výslednou známku. Otázky lze libovolně nastavovat a také vkládat i neomezeně dlouhé texty či obrázky.[21]

SMS centrum

Jedná se o unikátní službu, která umožňuje uživatelům komunikovat prostřednictvím SMS zpráv. Je možné nastavit automatické posílání SMS zpráv systémem iŠkola.cz a nechat se pravidelně informovat o suplování, hodnocení či informace z vývěsky. Bohužel tato služba je zpoplatněna nad rámec ceny za používání aplikace. [21]

Komunikace

Jedná se o interní komunikační systém školy, který umožňuje učitelům nebo vedení školy do systému přidávat nová sdělení. Mohou nastavit, komu se mají tyto informace zobrazit (učitelé, žáci, rodiče). Je možné nastavit rozeslání zprávy na uživatelské e-maily, a pokud škola využívá SMS centrum, lze rozesílat i SMS zprávy.[21]

E-mail pro každého

Jednoduchý modul, který vytváří neomezenou e-mailovou schránku pro každého uživatele systému. Díky tomu jsou všechny e-maily jednotné pro celou školu.[21]

Domácí úkoly

Tento modul umožňuje učitelům a administrátorům jednoduchou správu domácích úkolů. Vytvořené domácí úkoly lze zadat jednotlivcům nebo celé třídě a žáci musí úkol vypracovat a odevzdat. Díky propojení se známkovacím modulem, lze úkoly jednoduše ohodnotit a známky se objeví ihned v hodnocení žáka.[21]

Schránka důvěry

Jedná se o elektronickou podobu schránky důvěry, kam mohou žáci anonymně zasílat zprávy, které mohou být pro vedení školy důležité. Číst tyto zprávy mají možnost pouze oprávnění uživatelé.[21]

Zkušební plány

Užitečný modul, který umožňuje evidovat zkušební plány. Pokud učitel zadá do IS, kdo bude v určený den zkoušen, obdrží vybraný žák o tom včas informaci pomocí e-mailu či SMS a může se na zkoušení lépe připravit. [21]

Školní matrika

Rozsáhlý modul, který umožňuje uchovávat povinnou evidenci všech žáků dle příslušných právních předpisů. Modul také umožňuje export dat do souborů, které se předávají UIV.[21]

Ostatní moduly

Informační systém iŠkola.cz nabízí mnoho dalších modulů, jako závěrečné zkoušky, maturita, přijímací zkoušky a mnoho dalších, které jsou pro potřeby základní školy nevyužitelné.[21]

3.2.2 Přístupová práva

Z bezpečnostního hlediska je celý systém navržen s několikastupňovou ochranou proti úniku dat a zneužití. Právní struktura umožňuje správci virtuální iŠkoly.cz nastavit rozsáhlá přístupová práva. Díky tomu lze velmi přesně nastavit, že učitelé mohou dávat známky pouze z těch předmětů, které učí a dané třídě kterou učí. Systém se rozděluje na několik typů uživatelů.[21]

Nepřihlášený uživatel

Jedná se o jakéhokoliv návštěvníka systému. Nepřihlášený uživatel nemá v IS přístup k žádným datům uloženým v informačním systému.[21]

Admin

Jedná se o uživatele s nejvyšším oprávněním. Má právo ve vytvořené škole dělat vše. Může přidávat, upravovat i mazat předměty, učitele, známky, domácí úkoly i testy. Je schopen modifikovat uživatelská oprávnění. [21]

Učitel

Má práva pouze na správu známek předmětů, které vyučuje a pouze ve třídě kterou učí. Dále může používat systémy na domácí úkoly či testy, které muže hodnotit. Pokud je učitel v dané třídě třídním, má možnost vidět známky všech žáku ve své třídě a ze všech předmětů.[21]

Žák

Uživatel s tímto oprávněním může prohlížet a editovat své osobní údaje. Má přístup pouze ke svým domácím úkolům, testům a hodnocením.[21]

Rodič

Jedná se o uživatele s nejmenším oprávněním, který je svázán se svým dítětem (žák / student). Může pouze sledovat informace o dítěti a editovat své osobní údaje.[21]

Skupiny

iŠkola.cz umožňuje vytvářet skupiny uživatelů, které se chovají jako jakýkoliv jiný objekt. Takže je možné vytvářet skupiny pro jednotlivé kroužky, půlené hodiny a jiné. Nad jednotlivými skupinami lze nastavovat individuální oprávnění.[21]

4 Zabezpečení dat webových aplikací

4.1 Sql injection

4.1.1 Co je SQL injection?

V současné době všechny dynamické webové aplikace využívají databáze pro ukládání dat, ve kterých je uloženo mnoho citlivých informací jak o uživatelích, tak i mnoho jiných důležitých informací jako jsou kontakty a jiné.

SQL injection je technika, pomocí které je útočník schopen napadnout databázovou vrstvu programu vsunutím (odtud „injection“) kódu přes neošetřený vstup. To mu umožňuje podsunutí a vykonání vlastního pozměněného SQL dotazu. Pokud není aplikace správně ošetřena, může útočník v databázi provádět téměř cokoli, jako je zjištění všech zaregistrovaných uživatelů či smazání celé databáze. Proto by měly být aplikace dokonale otestované proti tomuto útoku. Je to jeden z nejjednodušších a nejčastějších útoků na webové aplikace. Přestože je to známá praktika, je na Internetu tolik webů spravovaných převážně nezkušenými programátory, kteří o tomto typu útoku prostě neví a tuto kritickou chybu opomíjejí. [3]

4.1.2 Ukázka napadení

Pro ukázku napadení mějme tabulku uziv, ve které budou pro zjednodušení pouze dva sloupce (jméno a heslo). Jakmile přijde uživatel na stránky, přihlásí se do systému pomocí formuláře, kde vyplní přihlašovací jméno a heslo. Díky tomu se spustí PHP skript, který pošle SQL dotaz do databáze. Ten by mohl vypadat takto:

```
SELECT * FROM uziv WHERE jmeno = '" + $zadaneJmeno + "'; "
```

Pokud tento dotaz proběhne v pořádku, z databáze se vrátí uživateli právě jeden řádek. Pokud ale uživatel zadá jako jméno třeba a'; DROP TABLE uživatele; --, bude dotaz i jeho vyhodnocení změněno.

```
SELECT * FROM uziv WHERE jmeno = 'a'; DROP TABLE uziv; --';
```

Díky tomuto dotazu útočník docílí toho, že řádně ukončí select, kde jeho výsledek je pro útočníka nezajímavý a podstrčí vlastní dotaz, který smaže celou tabulku uziv a dvě pomlčky na konci zakomentují koncový apostrof a středník.

Takto agresor získá velice jednoduše naprostou kontrolu nad celou databází a pomocí složitějších konstrukcí může získat veškeré informace uložené v databázi či celou databázi smazat. [3]

4.1.3 Zabezpečení na straně aplikace

Pro zabezpečení na straně aplikace využíváme vestavěných funkcí. Máme dvě základní, pomocí kterých můžeme zvýšit zabezpečení vytvářeného webu.

Escapování

Escapování je náhrada znaků majících v daném kontextu specifický význam na jiné odpovídající sekvence. Takže pokud budeme chtít například do řetězce ohraničeného uvozovkami zapsat uvozovky. Jelikož uvozovky mají v kontextu řetězce speciální význam a jejich prosté zapsání by bylo chápáno jako ukončení řetězce, je potřeba je zapsat jinou odpovídající posloupností. [4]

Regulární výrazy

Regulární výraz je speciální řetězec znaků, který představuje určitý vzor pro textové řetězce. Regulární výrazy se proto nejčastěji používají ke kontrole dat zadávaných ve formulářích. Pro ukázku budeme chtít ověření vstupních dat, kde čekáme pouze číselné hodnoty a použijeme funkci ereg, která má dva parametry. První je regulární výraz a druhý vstupní data. Pokud data odpovídají, tak funkce vrací true, jinak false. [5]

4.1.4 Zabezpečení na straně databáze

Abychom zabezpečili databázi je možné využít spoustu ověřených postupů. Pro přistup k databázi přes aplikaci by měl být v databázi vytvořen uživatel, který bude mít pouze ta práva, která aplikace vyžaduje. Tedy pokud potřebujeme jenom číst data, je vhodné uživateli nastavit práva jenom pro select. Díky tomuto opatření bude naše aplikace zase méně zranitelná, protože uživatel bez dostatečných práv nic nenadělá.

Pohledy

Jako další bezpečnostní prvek můžeme využít pohledy, díky kterým můžeme před útočníky skrýt pravou strukturu databáze. Další výhoda je, že u pohledů můžeme nastavit pouze čtení nebo zápis. Pokud neuvedeme nic, může pohled číst i zapisovat. [6]

Příklad pohledu:

```
CREATE VIEW login AS
SELECT jmeno, heslo
FROM uzivatele
WITH READ ONLY;
```

Procedury a funkce

Použití procedur a funkcí je jednou z nejlepších ochran, kterou máme k dispozici. Klíč je v tom, že procedura i funkce mají vstupní parametr, se kterým následně provedou určenou akci a vrátí výsledná data. Pomocí procedur a funkcí zabráníme potenciálnímu útočníkovi, aby měl přístup ke struktuře databáze, protože funkce mu to nedovolí. [7]

Ukázka funkce, která zjistí všechny známky daného studenta v jednotlivých předmětech.

```
CREATE OR REPLACE
FUNCTION SEZNAM_ZNAMEK(IDPREDMETU VARCHAR2,NICK VARCHAR2)
        RETURN VARCHAR2
IS
        seznam VARCHAR2(1000);
BEGIN
         FOR ZRETEZENY_SEZNAM IN (
              SELECT ZNAMKA FROM ZNAMKY ZN
              JOIN STUDENTI ST ON ST.ID_STUDENTA=ZN.ID_STUDENTA
              JOIN UZIVATELE UZ ON UZ.ID_UZIVATELE=ST.ID_UZIVATELE
              WHERE ID_PREDMETU=IDPREDMETU AND uz.PREZDIVKA=NICK
         )
    LOOP
      seznam := seznam || zretezeny_seznam.ZNAMKA ||'  ';
    END LOOP;
        RETURN seznam;
END SEZNAM_ZNAMEK;
```

Doporučení

Vhodné je nezapomenout vypnout výpis chybových zpráv. V průběhu tvoření a ladění aplikace je to velký pomocník, však informace poskytnuté těmito výpisy jsou velmi cenné i pro útočníka, kterému je zobrazena struktura naší databáze. Vypnutí těchto zpráv je velice jednoduché. Stačí vložit soubor .htaccess do kořenového adresáře a do souboru napsat tyto dva řádky:

```
php_flag display_errors off
php_flag display_startup_errors off
```

Dále je vhodné nezapomínat, že můžeme použít celou řadu funkcí, které nám pomohou předcházet těmto útokům. Nejjednodušší volbou je použít vhodný framework. Dnes jsou nejznámější frameworky Nette nebo Zend, které dbají na bezpečnost. Kupříkladu použití Nette frameworku disponující databázovou vrstvou Dibi, která nám kontroluje veškeré vstupy a odstraňuje riziko na sql injection.

4.2 Cross Site Scripting

Cross-site scripting (XSS) je technika narušení WWW stránek využitím bezpečnostních chyb ve skriptech, což jsou zejména neošetřené vstupy. Útočník pomocí těchto chyb v zabezpečení webové aplikace dokáže do stránek vložit svůj vlastní javascriptový kód, s jehož pomocí může poškodit vzhled stránky, její odstavení nebo může dojít dokonce k získávání citlivých údajů návštěvníků stránek. [8]

4.2.1 Ukázka napadení

XSS útok spočívá v tom, že se útočníkovi podaří do napadené stránky podstrčit vlastní HTML kód, který se při následujícím zobrazení v prohlížeči interpretuje jako HTML. Jednoduchým případem ukazujícím XSS zranitelnost může být například tento soubor:

```
<?php
      echo $_GET[ ' id ' ];
?>
```

Jestliže chceme provést injekci, zavoláme skript kupříkladu následujícím způsobem:

```
www.aaa.cz/index.php?id=<h1>Ahoj světe</h1>
```

Zadáním této URL adresy v prohlížeči, se zobrazí text, *ahoj světe*, ovšem naformátovaný jako nadpis první úrovně. Velký problém nastává, pokud útočník přepíše hodnotu id například jako:

```
www.aaa.cz/index.php?id=<script>
window.location.href="http://www.seznam.cz"; </script>
```

Pak se útočníkovi podaří přesměrování stránky na vlastní, kde může zjistit například citlivá data od uživatelů. [9]

4.2.2 Ochrana proti XSS

Spolehlivá ochrana před XSS je přes všechny uvedené složitosti překvapivě jednoduchá – ošetřit důsledně všechny výstupy z aplikace funkcí htmlentities(). Tato funkce nahradí všechny HTML citlivé znaky jejich odpovídajícími textovými entitami. Například menšítko (<) nahradí za <, většítko (>) za > apod. Díky tomu se případné HTML značky podstrčené útočníkem neinterpretují v prohlížeči ve svém významu, ale vypíší se na obrazovku přesně tak, jak je útočník zadal. Náš shora uvedený jednoduchý skript bychom tedy mohli opravit:

```
<?php
      echo htmlentities( $_GET[ ' id ' ] );
?>
```

Pokud nyní jej zavoláme jako www.aaa.cz/index.php?id=<h1>Hello world</h1>, server pošle zpátky klientovi zdrojový kód <h1>Hello world</h1> a v okně prohlížeče se vypíše prosté <h1>Hello world</h1> jako text.

4.3 Ukládání hesla

Z důvodu zadávání hesel v průběhu registrace pro budoucí přihlášení uživatelů je zapotřebí ukládat hesla tak, aby k těmto citlivým údajům neměl nikdo přístup. Pro zabezpečení hesla do nečitelné podoby slouží hashovací funkce. Existuje několik funkcí, kde nejčastěji používaná jsou sha1 a MD5. Tyto funkce se liší pouze tím, který algoritmus hashovaní používají. Vytvoří otisk (hash) zadaného řetězce a ten je poté uložen do databáze. Hashovací funkce jsou pouze jednosměrné, což znamená, že z otisku hesla nelze zpětně získat jeho původní hodnotu. Jelikož však dva různé řetězce (hesla) nemohou mít stejný otisk, můžeme při přihlašování porovnat, zda se otisk zadaného hesla shoduje s otiskem uloženého hesla. [10]

5 Použité technologie

5.1 HTML

HTML je značkovacím jazykem pro specifikaci rozvržení dokumentu a hypertextových odkazů. Přesně určuje syntaxi a rozmezí speciálních vložených příkazů, které se v prohlížeči přímo nezobrazují, ale které ovládají metodu vyobrazení obsahu dokumentu, včetně textu, obrázků a ostatních podpůrných medií. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci dokumentů na Internetu a je charakterizován množinou značek (tagů) a jejich atributů. Mezi značky se usazují úseky textu dokumentu a tím vznikají takzvané elementy. Názvy jednotlivých značek se uzavírají mezi úhlové závorky < a >. Značky[tagy] jsou obvykle párové, kde koncová značka je totožná se značkou výchozí, jen má před názvem znak lomítko.[11]

Přiklad:

```
    <b>Televize</b>
```

Tento příklad znamená, že text Televize bude zobrazen tučně, kde značka je počáteční tag elementu a značka je koncový tag. Jednotlivé elementy mohou obsahovat další, ale nesmějí se navzájem křížit.

Příklad:

```
      <b><u>Správný zápis</u></b>
      <b><u>Nesprávný zápis</b></u>
```

HTML se používá pro zobrazování informací na displeji uživatele. Soubory s html kódem mívají koncovku .htm nebo .html. Zde je ukázka kódu html: [12][13]

```
<!DOCTYPE html>
<html>
    <head>
        <title>Titulek stránky</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

    </head>
    <body>
      <h1>Nadpis 1. úrovně</h1>
    </body>
</html>
```

5.2 PHP

PHP je skriptovací programovací jazyk, který je určený zejména pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML či XHTML, což lze využít při vytváření webových aplikací.

Při využití PHP pro dynamické stránky jsou skripty prováděny na straně serveru, kde k uživateli je přenášen až výsledek jejich činnosti. Syntaxe jazyka je inspirována několika programovacími jazyky (Perl, C, Pascal a Java). PHP je nezávislý na platformě, skripty fungují bez větších modifikací na mnoha různých operačních systémech. PHP podporuje zpracování textu, grafiky, práci se soubory i přístup k většině databázových systémů (mj. MySQL, Oracle, PostgreSQL, MSSQL) nebo podporu celé řady internetových protokolů. [14] [15]

Příklad:

```
<?php
if ($retez == $cislo)
{
      echo "Jsou stejné";
}
      $promenna = "ahoj, světe!";
      echo $promenna;
?>
```

5.3 CSS

5.3.1 CSS

CSS je jazyk pro popis metody zobrazení stránek napsaných v jazycích HTML, XHTML nebo XML a byly navrženy pro oddělení datové části dokumentu od části popisující jeho vzhled. Syntaxe se skládá ze selektoru a bloku deklarací. Každý blok deklarací obsahuje vlastnost, posléze dvojtečku, hodnotu vlastnosti a ukončovací středník.

Příklad:

```
.novinkaPat{
    border-top: #D44413 1px solid;
    background-color: #FBF2EF;
    color: #D44413;
    text-align: right;
    font-size: 10pt;
    padding:  3px 15px;
}
```

5.3.2 Selektory

CSS definuje mnoho rozdílných selektorů, které zpravidla můžeme kombinovat.

- Body – platí pro všechny výskyty elementu body.
- Body p – platí pro všechny elementy p, které se nachází v elementu body, v jakékoliv hloubce.
- Body>div – platí pro všechny elementy div, které jsou přímými potomky elementu body.
- .trida – platí pro všechny elementy, které mají v HTML nastavenou třídu trida. To se provádí pomocí HTML atributu class.
- #id – platí pro všechny elementy, které mají v HTML nastavený identifikátor id. To se provádí pomocí HTML atributu id.
- sel1, sel2, sel3 – platí pro všechny selektory, protože selektory můžeme seskupovat pomocí čárek.

5.3.3 Připojení kaskádových stylů do HTML stránky

Existují 3 možné způsoby, jak aplikovat kaskádové styly v HTML dokumentu.

- Přímý inline zápis stylu pomocí atributu style, kde pravidla budou aplikována pouze na dotyčný element.
  - Příklad:

```
<p style="color: red; text-decoration: underline">Tento odstavec bude
červený a podtržený.</p>
```

- Zápis stylů do elementu style, kde se styly aplikují na celou stránku podle předepsaných selektorů.
  - Příklad:

```
<style type="text/css">
    #hlavicka{
      width: 200px;
      height: 450px;
    }
</style>
```

- Připojení externího souboru pomocí elementu link, který je nejčastější. [16]
  - Příklad:

```
<head>
<link href="styly.css" media="screen" rel="stylesheet" type="text/css">
</head>
```

5.4 Oracle Databáze

Relační databáze Oracle je založena na modelu dvourozměrných tabulek (sloupce a řádky). Ty mohou být propojeny neboli je mezi nimi vytvořena relace. Oproti hierarchickému modelu, relační nepředpokládá žádné vazby mezi tabulkami. Uživatel nemusí chápat fyzické uspořádání dat v databázi, aby je byl schopen získat. Toto pomohlo k velké oblíbenosti a rozkvětu relačních databází v 90. letech minulého století.

Aktuálně Oracle nabízí svůj databázový systém s pojmenováním Database 11g. Od své první podoby prošel tento systém obrovským množstvím vylepšení a z primitivní databáze podporující pouze standardní SQL se stal souhrnným systémem, který je používán v těch nejvýznamnějších společnostech.

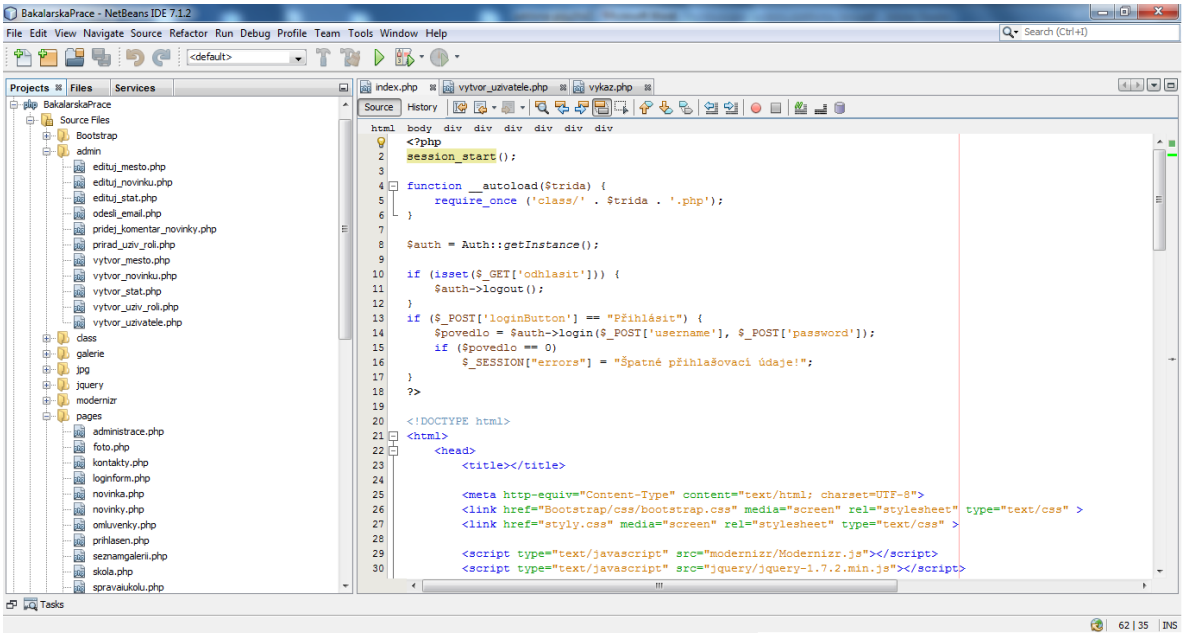
Systém je propagován v mnoha verzích rozdělených dle funkcí a určení. Pro osobní či nekomerční použití v menších organizacích je k dispozici varianta distribuovaná bezplatně a určená pro obsluhu drobných databází – Oracle Express Edition. Pro komerční a náročnější použití je nabízena jedna z pokročilejších verzí, která nabízí obsluhu obsáhlejších databází.

Individuální propagované verze se odlišují zvláště množstvím procesorů a velikostí paměti serveru. Je tedy zřejmé, že pro malou databázi stačí bezplatně distribuovaná verze Expres Edition, která je schopna běžet bez problému na jednoprocesorovém serveru s menším velikosti paměti RAM. Naopak velká a rozsáhlá databáze musí používat některou z vyšších verzí, která podporuje použití více procesorů a větší množství paměti. [17][18]

5.5 NetBeans IDE

Jedná se o profesionální vývojové prostředí (Obrázek 4), které nám poskytuje mnoho užitečných funkcí ulehčujících vývoj aplikací. Je však potřeba dobře vědět, kde je hledat a jak je správně použít. Velkou výhodou je, že NetBeans je zdarma ke stažení a je volně šiřitelný. Jednou z dalších výhod je kupříkladu výborný editor, který značně napomáhá psaní zdrojového kódu. Různé opravy kódu jsou nejen jednodušší a rychlejší, ale mnohem bezpečnější, protože se snižuje výskyt chyb, které vznikají překlepy či nepozorností. Jednotlivé části kódu jsou barevně rozlišeny, a tím se stává zdrojový kód mnohem přehlednějším.

Představme si, že máme napsanou funkci, kterou používáme na více místech. Pokud bychom chtěli změnit její název, museli bychom ji všude ručně přepisovat. Při použití jednoho z nástrojů programovacího prostředí NetBeans to můžeme udělat jednoduše. Stačí jméno změnit jen na jednom místě a NetBeans se postará o to, aby na každém místě, kde se vyskytoval stejný název, přepsal novým. [19]



Obrázek 4 - Vývojové prostředí NetBeans

(Zdroj: Vlastní)

6 Návrh a vývoj aplikace IS

6.1 Vzhled aplikace

Obrázek 5 - Layout aplikace

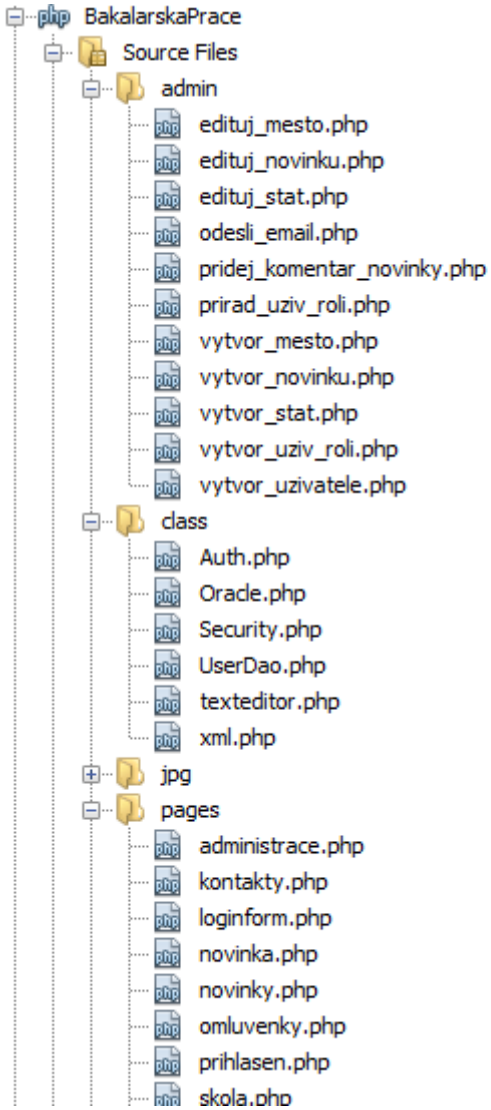
(Zdroj: Vlastní)

Z předchozího obrázku (Obrázek 5) je vidět vzhled aplikace, tedy její úvodní obrazovku. V horní části aplikace je umístěno logo, pod ním je hlavní menu a nyní se nacházíme na první záložce (Novinky), která je úvodní. Dále je možné si vybrat z jiných záložek (Škola, Foto, Události, Zaměstnanci, Kontakty). Pod tímto menu se nachází dvě části, kde v levé části je samotný obsah již spojený s možností z menu, v našem případě informace o novinkách, které nás informují o novém dění ve škole, a je možné po přihlášení tyto jednotlivé novinky i komentovat. V pravé části se nachází formulář pro přihlášení do systému. Zde se následně po přihlášení místo formuláře zobrazí takzvané panely dle uživatelské role, které nabízejí příslušné služby. Stránka je zakončena patičkou, kde je podepsán autor.

6.2 Adresářová struktura

Projekt je logicky rozdělen do následující adresářové struktury (Obrázek 6), kde jednotlivé adresáře shromažďují podobnou funkcionalitu z důvodu usnadnění orientace programátora. V kořenovém adresáři je hlavní soubor index.php a soubor styly.css, který obsahuje kaskádové styly a formátuje vzhled celých stránek.

- Admin – v tomto adresáři jsou soubory*.php, které zajišťují veškerou administraci, ke které má přístup pouze uživatel s rolí Administrátor. Do této sekce patří správa adres, novinek, uživatelských rolí, uživatelů a komentářů.
- Class – v této složce je ukryta veškerá funkcionalita zajištující komunikaci s databází, do které patří zjišťování přihlášení uživatele či kontrola uživatelských rolí.
- Jpg – tato složka je určena pro všechny obrázky, které se vyskytují na webu, nikoliv však obrázky z galerií.
- Galerie – obsahuje veškeré obrázky z galerií.
- Pages – hlavní složka, která zajišťuje funkcionalitu a zobrazení všech stránek a formulářů.
- Teach - v tomto adresáři jsou soubory*.php, které zajišťují veškerou administraci, ke které má přístup uživatel s rolí Ucitel. Do této sekce patří správa úkolů, událostí a známek.



Obrázek 6 - Adresářová struktura

(Zdroj: Vlastní)

6.3 Možnosti aplikace

6.3.1 Školní akce

Obrázek 7 - Události školy

(Zdroj: Vlastní)

Z předchozího obrázku (Obrázek 7) je dobře vidět, že získání informací o školních akcích je možné i bez přihlášení pomocí záložky *Události* již v hlavním menu. Po stisku tlačítka jsou zobrazeny v levé části aplikace všechny plánované i uskutečněné školní akce. Je možné u každé události vyčíst její název, obsah a popis dané události, kdy bude nebo byla zahájena, kdy bude či byla ukončena, a kdo danou událost vytvořil.

6.3.2 Generování sestav pro pravidelné výkazy

Obrázek 8 - Generování sestav

(Zdroj: Vlastní)

Aby bylo možné generovat výkazy, musí se uživatel nejprve přihlásit do systému a mít přiřazenou roli *Učitel* nebo *Administrátor*. V pravém sloupci z panelu pro učitele zvolit možnost *Generovat výkaz* a následně v levé části aplikace vybrat ze seznamu studentů. Jakmile je vybrán student, systém automaticky vygeneruje tabulku pouze s těmi předměty, u kterých má daný student zapsané známky, jak je možné vidět na předchozím obrázku (Obrázek 8). Následně systém ke každému předmětu vypočítá známku, na základě průměru známek a jejich důležitosti a výsledek doplní do tabulky. Uživatel má ještě možnost přihlédnout například k chování či snaživosti studenta a známku upravit. Pokud je uživatel se známkami již spokojen, použije tlačítko sloužící k přípravě pro tisk, kde se mu zobrazí čistě výsledná tabulka, kterou je možné vytisknout (Obrázek 9).

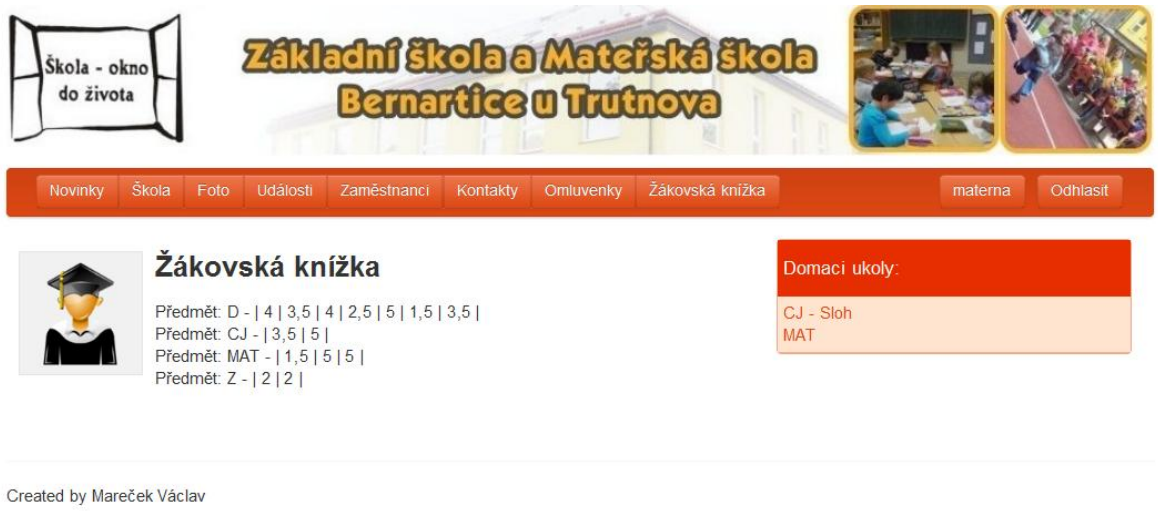
Student:Mareček Václav

Předměty	MAT	Z	D	CJ	TV	Fyz	VL	CH	PR
Známky	4+	2	3-	4	2	1	1	2+	3

Obrázek 9 - Výkaz pro tisk

(Zdroj: Vlastní)

6.3.3 Evidence prospěchu



Obrázek 10 - Evidence prospěchu

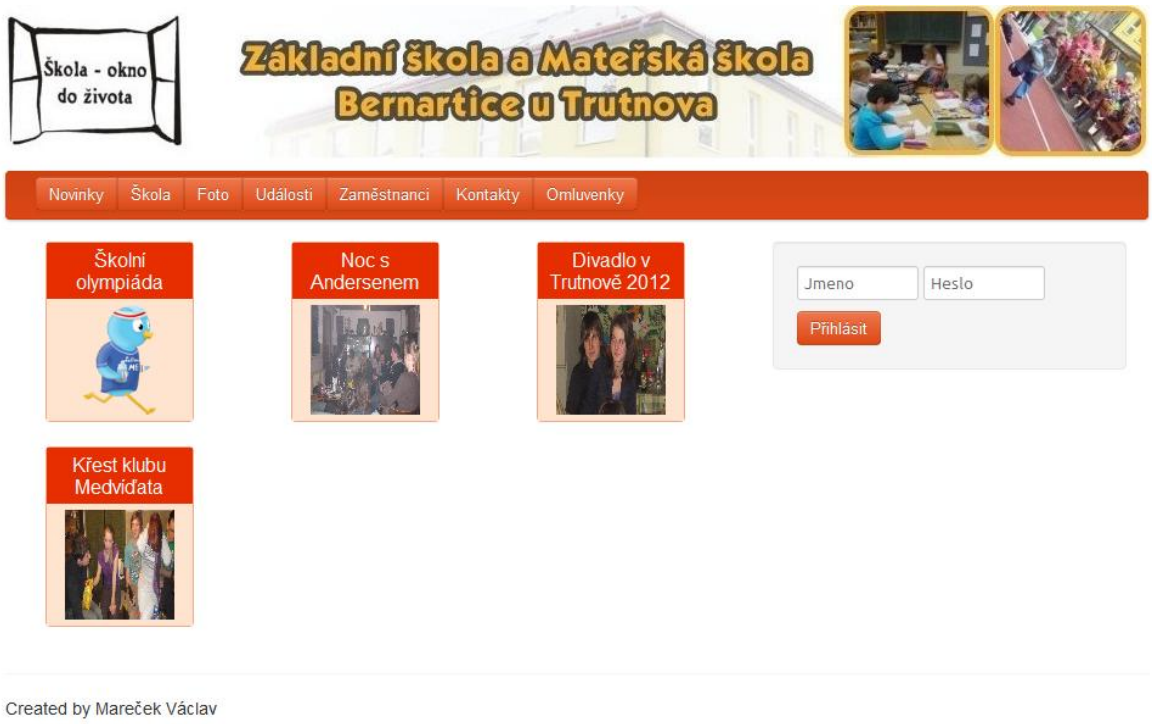
(Zdroj: Vlastní)

Každý student může sledovat vlastní prospěch i přes aplikaci (Obrázek 10). Jakmile se přihlásí na svůj uživatelský účet, hlavní menu se rozšíří o položku *Žákovská knížka*. Po vybrání této možnosti, se v levé části zobrazí veškeré známky, které má student zapsané v systému, včetně toho, ke kterému předmětu patří. Každý zástupce studenta má také přístup k těmto údajům. Stačí se přihlásit do systému, z pravé nabídky vybrat studenta a v levé části se zobrazí jeho známky s předměty i jeho nesplněné úkoly (Obrázek 11).

Obrázek 11 - Evidence pro zástupce

(Zdroj: Vlastní)

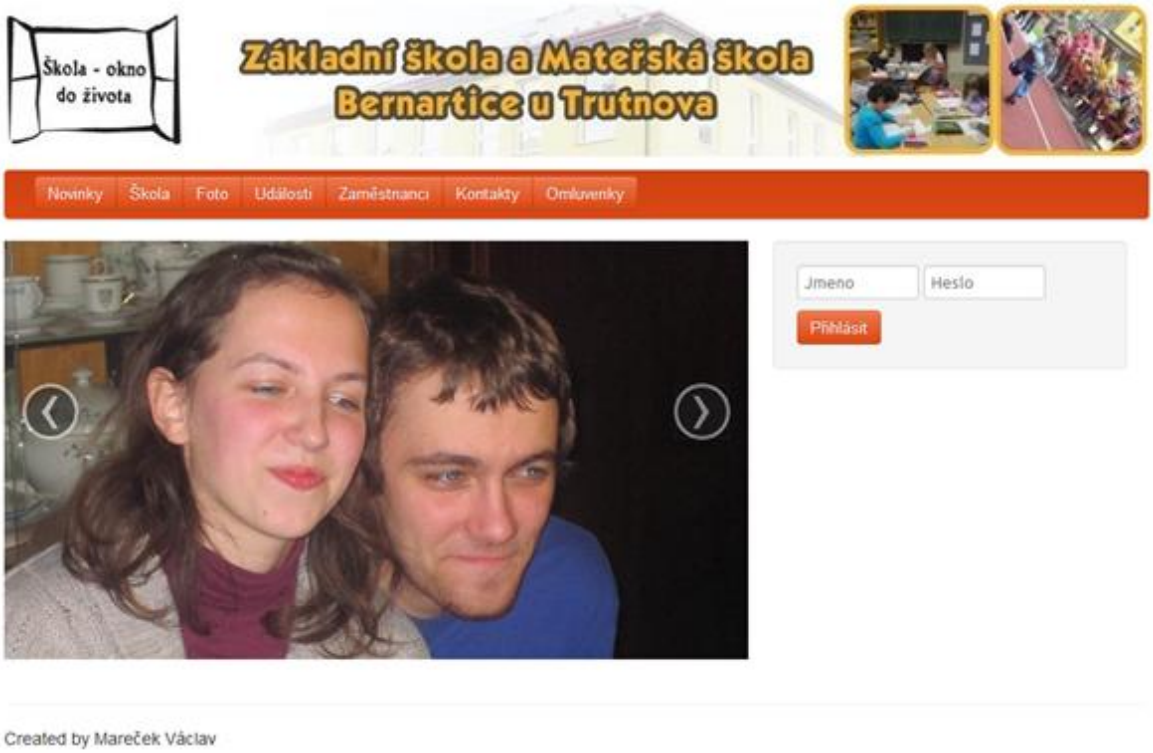
6.3.4 Prohlížení fotografií



Obrázek 12 - Seznam skupin obrázků

(Zdroj: Vlastní)

Prohlížení fotografií je možné již z veřejné části, tudíž není potřeba se do systému přihlašovat. Stačí zvolit položku *Foto* z hlavního menu. Následně se v levé části aplikace zobrazí seznam všech skupin, kde se ke každé skupině vytvoří *okénka*, která se přehledně rozmístí po celé pravé části aplikace. V horní části tohoto *okénka* se zobrazí názvy jednotlivých skupin a pod ním se náhodně přiřadí jedna fotografie z dané galerie, čímž se zajistí lepší přehlednost a představa o skupině obrázků (Obrázek 12). Po vybrání jedné skupiny se zobrazí intuitivní prohlížeč obrázků, díky kterému je prohlížení velice příjemné (Obrázek 13).

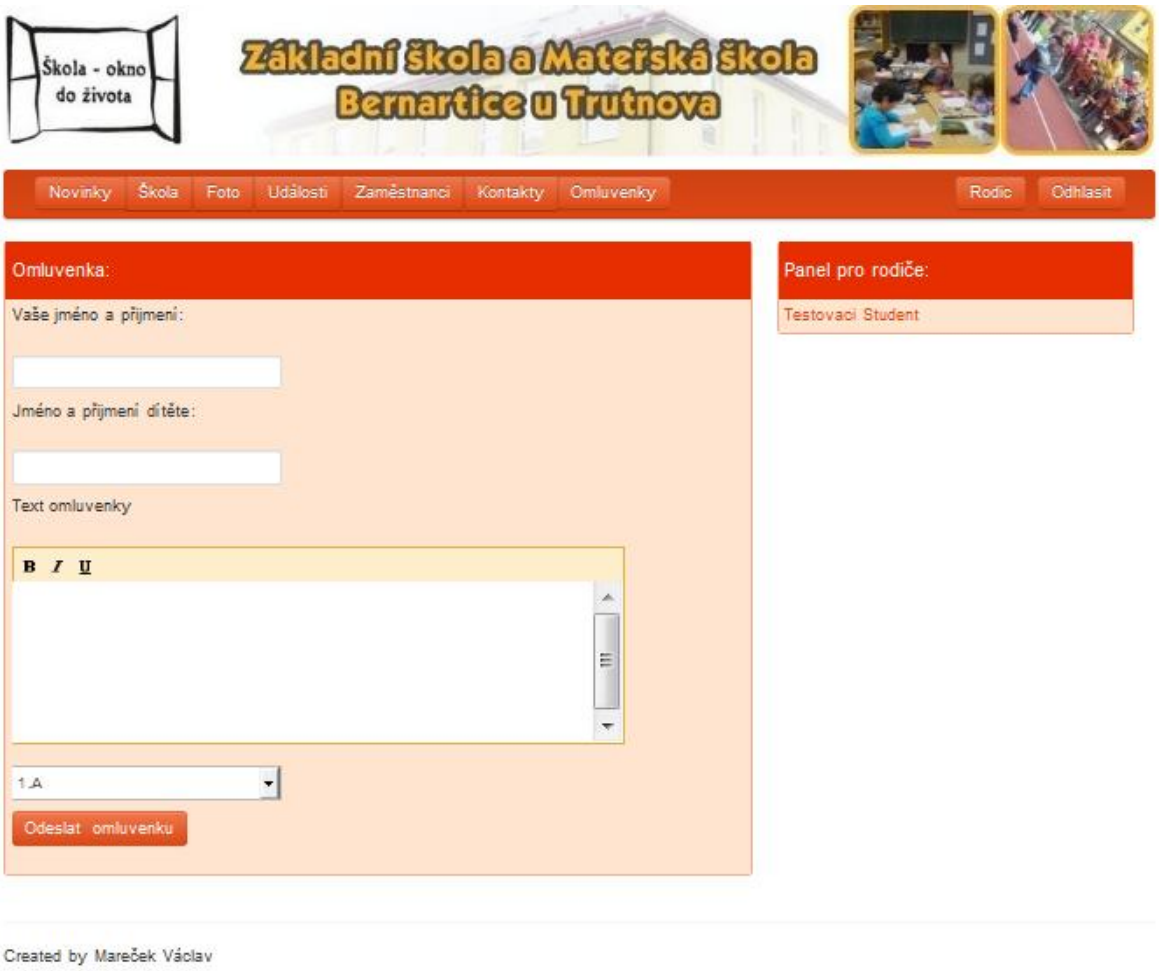


Obrázek 13 - Prohlížeč obrázků

(Zdroj: Vlastní)

Používání tohoto prohlížeče je velice jednoduché. Pomocí pravé šipky se posouváme na další fotografie v galerii a pomocí levé šipky se vracíme o fotografii zpátky. Nestandardní výhodou tohoto prohlížeče je ta vlastnost, že pokud přesuneme ukazatel myši pryč z levé či pravé šipky, spustí se automatické prohlížení. Pokud se zobrazí poslední fotografie, začíná se procházet galerie znovu.

6.3.5 Omlouvání docházky



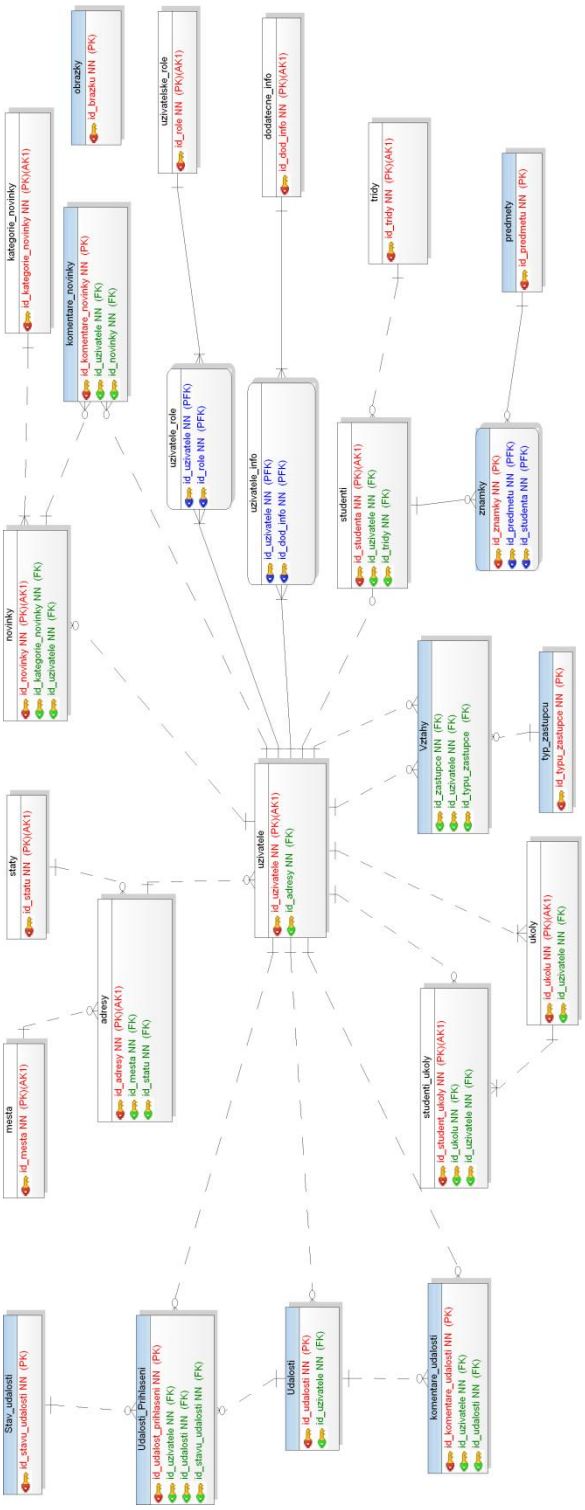
Obrázek 14 - Omlouvání docházky

(Zdroj: Vlastní)

Pro omlouvání docházky je zapotřebí, aby se zástupce studenta přihlásil do systému pomocí přihlašovacího formuláře v levé části aplikace. Pokud je přihlášený uživatel v roli *Rodič*, rozšíří se hlavní menu o další položku *Omluvenky*. Po vybrání této možnosti z menu se zobrazí v pravé části aplikace příslušný formulář (Obrázek 14), kde je zapotřebí vyplnit jméno a příjmení zástupce studenta, jméno a příjmení studenta, jeho třídu a následně důvod a doba jeho absence. Jakmile je vše potřebné vyplněné, pomocí tlačítka *Odeslat omluvenku* se odešle e-mail se všemi vyplněnými informacemi na e-mailovou adresu administrátora aplikace, či jinou nastavenou adresu.

6.4 Databázové schéma

Pro rozsáhlost fyzického databázového modelu jsem byl nucen použít pouze zobrazení primárních a cizích klíčů (Obrázek 15).



Obrázek 15 - Databázové schéma

(Zdroj: Vlastní)

6.5 Uživatelské role

Všichni uživatelé využívající systém mají přesně stanoveno, do kterých částí mají přístup. Pro rozlišení, jaký uživatel má která práva, slouží uživatelské role, které jsou každému registrovanému uživateli automaticky přiřazena.

Nepřihlášený uživatel

Pokud není uživatel přihlášen, má práva pouze do veřejné sekce, kde si může prohlížet základní informace o škole, kontakty, základní výpis zaměstnanců školy nebo fotografie. Dále může prohlížet novinky a události, které jsou určené i pro veřejnost, ale komentovat je nemohou.

Student

Uživatel s rolí student má oproti nepřihlášenému uživateli navíc přístup do žákovské knížky, kde nalezne všechny své známky jednotlivých předmětů. Dále má přístup do studentské sekce, kde může zjistit zadané úkoly, datum jejich odevzdání. Navíc mají možnost komentovat události a novinky.

Učitel

Uživatelé s rolí učitel mají přístup do učitelské sekce, kde mohou připravovat písemné práce, vytvářet domácí úkoly a zadávat je jednotlivým studentům. Dále mohou vytvářet události, které jsou viditelné i ve veřejné části. Mají práva ke správě známek, kde mohou jednotlivým studentům přidávat známky do žákovských knížek. Také mohou komentovat události i novinky.

Rodič

Všichni uživatelé s rolí rodič mají přístup do rodičovské sekce, kde vidí všechny své děti, u kterých si mohou zjistit jak veškeré známky jednotlivých předmětů, tak zda mají splněné všechny domácí úkoly. V neposlední řadě mají přístup k elektronické omluvence, kde mohou omlouvat nepřítomnost svých dětí. Také mohou komentovat novinky i události.

Administrátor

Uživatelé s rolí administrátor, mají nejvyšší práva v systému. Mají přístup jak do učitelské sekce, tak do administrátorské sekce, kde mohou přidávat nové uživatele, přidávat uživatelům nové role, spravovat adresy a vytvářet, editovat či komentovat novinky i všechny události.

7 Závěr

Cílem bakalářské práce bylo seznámit čtenáře se základní problematikou ochrany osobních údajů. Na základě získaných informací a jejich následným zpracováním je předložen seznam zákonů, které se této problematiky týkají, vysvětlena práva a povinnosti, která vznikají uživateli. V dalších částech jsou čtenáři poučeni, jak své údaje zabezpečit a co dělat v případě zneužití těchto údajů. Dále jsou čtenáři informováni o různých druzích útoků, se kterými je možné se v dnešní době setkat a je nutné tedy s nimi počítat. Nastínil jsem, jak je možné ze zákona se proti těmto útokům bránit a zabezpečit citlivá data při vývoji svých aplikací. Závěrem teoretické práce jsou čtenáři seznámeni s jinými systémy dostupnými na trhu.

Druhým cílem bylo vytvořit internetovou aplikaci, která usnadní komunikaci mezi rodiči dětí a základní školou a rozšířit tak stávající nabídku na trhu. Myslím, že zavedení této aplikace bude přínosem pro školu. Míra usnadnění práce bude více zhodnotitelná až při delším ověřování v praxi. Aplikace byla vyvíjena pro internetový prohlížeč Opera a následně byla optimalizována pro internetový prohlížeč Google Chrome. Při vývoji jsem kladl důraz na bezpečnost dat, avšak všechny bezpečnostní chyby nebyly opraveny. Pokud skončí testování v reálném provozu informačního systému úspěšně, v této práci bych rád pokračoval a rozšířil již hotový IS o další funkce a opravil některé části, které nejsou příliš uživatelsky přívětivé.

Jedním z cílů bylo implementovat do aplikace zapisování docházky, které by bylo možné, ale po dlouhém zvažování jsem od tohoto záměru ustoupil. Třídní učitelé by museli znovu přepisovat z třídní knihy docházku, což by bylo velice nepohodlné. Při úplném odstranění papírové formy by muselo být do každé učebny přidáno zařízení (počítač či tablet), na kterém by se zapisovala docházka. Jelikož tato zařízení jsou finančně náročná a škola na to nemá prostředky, tak jsem od tohoto cíle, po konzultaci se školou, upustil.

Literatura

1. Zákon č. 101/2000 Sb., o ochraně osobních údajů (účinné znění). *uoou.* [Online] Úřad pro ochranu osobních údajů, Tesco SW, a. s., 2012. [Citace: 28. července 2012.] http://www.uoou.cz/uoou.aspx?menu=4&submenu=5&loc=20.

2. **Sláma, David.** Fotografie: autorské právo a právo na ochranu osobnosti. *digiarena.e15.* [Online] 12. Prosinec 2010. [Citace: 31. červenec 2012.] http://digiarena.e15.cz/fotografie-autorske-pravo-a-pravo-na-ochranu-osobnosti_3.

3. wikipedia. *SQL injection.* [Online] 17. květen 2012. [Citace: 17. červenec 2012.]

4. **Grudl, David.** Escapování - definitivní příručka. *phpFashion.* [Online] 2012. [Citace: 18. červenec 2012.] http://phpfashion.com/escapovani-definitivni-prirucka.

5. **Pecka, Miroslav.** Regulární výrazy: Regexp není zaklínadlo... *regularnivyrazy.* [Online] 2008. [Citace: 18. červenec 2012.] http://www.regularnivyrazy.info/.

6. **LONEY, Kevin a BRYLA, Bob.** *Mistrovství v Oracle Databases 10g.* Brno : Coputer press, 2006. 80-251-1277-2.

7. **Scoot, Urman.** *Oracle : programovani v PL/SQL.* Brno : Computer press, 2007. ISBN 978-80-251-1519-0.

8. Cross-site scripting. *wikipedia.* [Online] 11. června 2012. [Citace: 18. červenec 2012.] http://cs.wikipedia.org/wiki/Cross-site_scripting.

9. **Tichý, Jan.** Cross-site scripting. *phpguru.* [Online] 22. únor 2008. [Citace: 18. červenec 2012.] http://www.phpguru.cz/clanky/cross-site-scripting.

10. **Fiala, Martin.** hash. *abclinuxu.* [Online] 21. srpen 2004. [Citace: 18. červenec 2012.] http://www.abclinuxu.cz/slovnik/hash.

11. **Kosek, Jiří.** Historie a vývoj HTML. *htmlguru.* [Online] 1997. [Citace: 11. červenec 2012.] http://htmlguru.cz/uvod-historie.html.

12. **Janovský, Dušan.** Verze HTML. *jakpsatweb.* [Online] 7. duben 2012. [Citace: 11. cervenec 2012.] http://www.jakpsatweb.cz/html/verze-html.html.

13. **Kennedy, Chuck Musciano & Bill.** *HTML a XHTML.* Praha : Computer Press, 2000. 80-7226-407-9.

14. PHP. *wikipedia.* [Online] 8. červen 2012. [Citace: 12. červenec 2012.] http://cs.wikipedia.org/wiki/PHP.

15. **Group, The PHP.** Historie PHP. *php.tonnikala.* [Online] 2. únor 2007. [Citace: 12. červenec 2012.] http://php.tonnikala.org/manual/cs/history.php.

16. Kaskádové styly. *wikipedia.* [Online] 11. červenec 2012. [Citace: 12. červenec 2012.] http://cs.wikipedia.org/wiki/Kaskádové_styly.

17. *Oracle.* [Online] Oracle Application Development Fra-mework, 2010. [Citace: 27. červenec 2012.] http://www.oracle.com/index.html.

18. **Lacko, Luboslav.** *ORACLE Správa, programování a použití databázového systému.* Brno : Computer press, 2003. 80-7226-699-3.

19. **Briškár, Matej.** Netbeans IDE - popis prostredia - II. *linuxsoft.* [Online] 20. září 2010. [Citace: 2. srpen 2012.] http://www.linuxsoft.cz/article.php?id_article=1761.

20. **a.s., ŠKOLA ONLINE.** aplikace.skolaonline. *Katedra - uživatelská příručka.* [Online] 2010. [Citace: 15. duben 2013.] https://aplikace.skolaonline.cz/dokumentace/KS/katedra/web/index.html.

21. **s.r.o., Computer Media.** iskola. *iškola.* [Online] 2005-2013. [Citace: 15. duben 2013.] http://www.iskola.cz/texty/coumi.php.

22. **a.s., ŠKOLA ONLINE.** aplikace.skolaonline. *Žákovská - uživatelská příručka.* [Online] 2010. [Citace: 15. duben 2013.] https://aplikace.skolaonline.cz/dokumentace/KS/zakovska/web/index.html.

Příloha A – Zdrojový kód souboru Security.php

```
class Security {


      public function prevodTextu($text, $maxLength = 2000, $textarea =
        false) {
        $result = htmlspecialchars($text, ENT_QUOTES);
        if ($textarea == true) {
            $result = $this->ObnovaTextArea($result);
        }
        if (strlen($result) > $maxLength) {
            return false;
        }
        return $result;
    }
public function ObnovaTextArea($txt, $maxLength = 2000) {

        $txt = @ereg_replace("&lt;p&gt;", "<p>", $txt);
        $txt = @ereg_replace("&lt;/p&gt;", "</p>", $txt);
        $txt = @ereg_replace("&lt;strong&gt;", "<strong>", $txt);
        $txt = @ereg_replace("&lt;/strong&gt;", "</strong>", $txt);
        $txt = @ereg_replace("&lt;em&gt;&lt;", "<em>", $txt);
        $txt = @ereg_replace("&lt;span style=&quot;text-decoration:
                  underline;&quot;&gt;", "<span style'text-decoration:
                  underline;'>", $txt);
        $txt = @ereg_replace("&lt;span style=&quot;text-decoration:
                  line-through;&quot;&gt;", "<span style'text-decoration:
                  line-through;'>", $txt);
        $txt = @ereg_replace("&lt;/span&gt;", "</span>", $txt);
        if (strlen($txt) > $maxLength) {
            return false;
        }
        return $txt;
    }

public function osetreniCisel($cislo, $maxHodnota, $kladna = true,
       $minHodnota = 0, $maxLength = 13, $minLength = 0) {
        $cislo = @ereg_replace("[^0-9|\.|\,]", "", $cislo);
        $cislo = @ereg_replace("\.", ",", $cislo);

        if ($cislo > $maxHodnota) {
            return false;
        }
        if ($kladna == true) {
            if ($cislo < 0) {
                return false;
            }
        }
        if ($cislo < $minHodnota) {
            return false;
        }
        if (strlen($cislo) > $maxLength || strlen($cislo) < $minLength) {
            return false;
        }
        return $cislo;
    }

}
```

Příloha B – CD

Na přiloženém CD naleznete kompletní vyvinutou aplikaci a také tento text v elektronické podobě.

Příloha C – Databázové tabulky

Zde se budu věnovat popisu jednotlivých tabulek a jejich atributů.

Tabulka 1 – uživatele

Tabulka uživatele obsahuje základní informace o uživatelích, kontakty a heslo, které je šifrováno pomocí vestavěné funkce crypt.

Tabulka 1 - Uživatelé

Název	Vlastnost	Popis
Id_uzivatele	Number(PK)	PK, Id uživatele
Id_adresy	Number(FK)	FK, Id adresy
Prezdivka	NVarchar2(50)	Přezdívka uživatele (nick)
Heslo	NVarchar2(100)	Heslo v šifrované formě (hash)
Jmeno	NVarchar2(50)	Křesní jméno
Prijmeni	NVarchar2(50)	Příjmení
Email	NVarchar2(40)	E-mail
Tel	NVarchar2(30)	Telefon
Titul_pred	NVarchar2(30)	Tituly před jménem
Titul_za	NVarchar2(30)	Tituly za jménem

Tabulka 2 – mesta

Tabulka mesta uchovává základní informace o městech.

Tabulka 2 - Města

Název	Vlastnost	Popis
Id_mesta	Number(PK)	PK, Id města
Mesto	NVarchar(50)	Název města
Psc	Number	Poštovní směrovací číslo

Tabulka 3 – staty

Tabulka staty uchovává základní informace o státech.

Tabulka 3 - Státy

Název	Vlastnost	Popis
Id_statu	Number(PK)	PK, Id státu
Stat	NVarchar(50)	Název státu
Zkratka	NVarchar(20)	Zkratka státu

Tabulka 4 – adresy

Tabulka adresy uchovává všechny adresy použité v systému.

Tabulka 4 - Adresy

Název	Vlastnost	Popis
Id_adresy	Number(PK)	PK, Id adresy
Id_mesta	Number (FK)	FK, Id města
Id_statu	Number (FK)	FK, Id státu
Ulice	Nvarchar2(50)	Ulice + číslo popisné

Tabulka 5 – novinky

Tabulka novinky uchovává přidané novinky do systému.

Tabulka 5 - Novinky

Název	Vlastnost	Popis
Id_novinky	Number (PK)	PK, Id novinky
Id_kategorie_novinky	Number (FK)	FK, Id kategorie novinky
Id_uzivatele	Number (FK)	FK, Id uživatele
Datum	Date	Datum vytvoření
Text	Clob	Text (hodnota) novinky
Viditelnost	Nvarchar2(20)	Zda bude viditelná veřejnosti
Nadpis	Nvarchar2(200)	Nadpis

Tabulka 6 – kategorie_novinky

Tabulka kategorie_novinky rozděluje novinky do různých kategorií.

Tabulka 6 - Kategorie novinky

Název	Vlastnost	Popis
Id_kategorie_novinky	Number (PK)	PK, Id kategorie
Nazev	Nvarchar2(50)	Název kategorie

Tabulka 7 – komentare_novinky

Tabulka komentare_novinky uchovává všechny komentáře k novinkám.

Tabulka 7 - Komentáře novinky

Název	Vlastnost	Popis
Id_komentare_novinky	Number (PK)	PK, Id komentářů novinek
Id_uzivatele	Number (FK)	FK, Id uživatele
Id_novinky	Number (FK)	FK, Id novinek
Komentar	Clob	Obsah komentáře
Datum	Date	Datum vložení

Tabulka 8 – uzivatelske_role

Tabulka uzivatelske_role uchovává různé uživatelské role, pomocí kterých se v aplikaci přiřazují práva.

Tabulka 8 - Uživatelské role

Název	Vlastnost	Popis
Id_role	Number (PK)	PK, Id uživatelské role
Nazev	NVarchar2(50)	Název role

Tabulka 9 – uzivatele_role

Tabulka uzivatele_role uchovává informace, které role má uživatel.

Tabulka 9 - Uživatelé role

Název	Vlastnost	Popis
Id_uzivatele	Number (FK)	PK, Id uživatele
Id_role	Number (FK)	FK, Id uživatelské role

Tabulka 10 – dodatecne_info

Tabulka dodatecne_info uchovává všechny nestandardní informace o uživatelích jako například pochvaly, poznámky, alergie či informace o náplních zaměstnání.

Tabulka 10 - Dodatečné informace

Název	Vlastnost	Popis
Id_dod_info	Number (PK)	PK, Id dodatečné informace
nazev	NVarchar2(500)	Název informace
Hodnota_infa	NVarchar2(500)	Hodnota informace
Typ_viditelnosti	NVarchar2(20)	Komu bude informace viditelná

Tabulka 11 – uzivatele_info

Tabulka uzivatele_info přiřazuje uživatelům dodatečné informace.

Tabulka 11 - Uživatelé informace

Název	Vlastnost	Popis
Id_uzivatele	Number (FK)	FK, Id uživatele
Id_dod_info	Number (FK)	FK, Id dodatečné informace

Tabulka 12 – studenti

Tabulka studenti uchovává informace o tom, kteří uživatelé jsou studenti.

Tabulka 12 - Studenti

Název	Vlastnost	Popis
Id_studenta	Number (PK)	PK, id studenta
Id_uzivatele	Number (FK)	FK, id uživatele
Id_tridy	Number (FK)	FK, id třídy
Rok_nastupu	Number	Rok nástupu studenta
Stav_studia	NVarchar2(30)	Stav studia studenta

Tabulka 13 – tridy

Tabulka tridy uchovává seznam všech tříd.

Tabulka 13 - Třídy

Název	Vlastnost	Popis
Id_tridy	Number (PK)	PK, Id třídy
Nazev	NVarchar2(30)	Název třídy

Tabulka 14 – znamky

Tabulka znamky uchovává známky všech studentů u jednotlivých předmětů.

Tabulka 14 - Známky

Název	Vlastnost	Popis
Id_znamky	Number (PK)	PK, Id známky
Id_predmetu	Number (FK)	FK, Id předmětu
Id_studenta	Number (FK)	FK, Id studenta
Znamka	Float	Známka
Datum	Date	Datum vložení známky
Dulezitost	Number	Důležitost známky
Popis	NVarchar2(500)	Popis udělení známky

Tabulka 15 – predmety

Tabulka predmety ukládá seznam všech předmětů a jejich zkratek.

Tabulka 15 - Předměty

Název	Vlastnost	Popis
Id_predmetu	Number (PK)	PK, Id předmětu
Nazev_predmetu	NVarchar2(100)	Název předmětu
Zkratka	NVarchar2(20)	Zkratka předmětu
Popis	Clob	Popis předmětu

Tabulka 16 – vztahy

Tabulka vztahy uchovává, kteří uživatelé zastupují jednotlivé studenty.

Tabulka 16 - Vztahy

Název	Vlastnost	Popis
Id_zastupce	Number (FK)	FK, Id zástupce
Id_uzivatele	Number (FK)	PK, Id dítěte
Id_typu_zastupce	Number (FK)	FK, typ zástupce
Priorita	Number	Priorita zástupce

Tabulka 17 – typ_zastupcu

Tabulka typ_zastupcu definuje typy zástupců.

Tabulka 17 - Typy zástupců

Název	Vlastnost	Popis
Id_typu_zastupce	Number (PK)	PK, id typu zástupce
Nazev	NVarchar2(50)	Název zástupce

Tabulka 18 – ukoly

Tabulka ukoly uchovává seznam všech úkolů.

Tabulka 18 - Úkoly

Název	Vlastnost	Popis
Id_ukolu	Number (PK)	PK, Id úkolu
Id_uzivatele	Number (FK)	FK, Id uživatele
Zadani	Clob	Zadání úkolu
Datum_zadani	Date	Datum zadání úkolu
Datum_konce	Date	Datum odevzdání úkolu
Nazev	NVarchar2(100)	Název úkolu

Tabulka 19 – studenti_ukoly

Tabulka studenti_ukoly ukládá seznam všech studentů, kteří mají přiřazen nějaký úkol.

Tabulka 19 - Úkoly studentů

Název	Vlastnost	Popis
Id_student_ukoly	Number (PK)	PK, Id student úkoly
Id_ukolu	Number (FK)	FK, Id úkolu
Id_uzivatele	Number (FK)	FK, Id uživatele
Stav	NVarchar2(50)	Stav úkolu
Datum	Date	Datum zadání úkolu

Tabulka 20 – udalosti

Tabulka udalosti uchovává všechny události.

Tabulka 20 - Události

Název	Vlastnost	Popis
Id_udalosti	Number (PK)	PK, Id udalosti
Id_uzivatele	Number (FK)	FK, Id uživatele
Nazev	NVarchar2(100)	Název události
Popis	Clob	Obsah události
Datum_zahajeni	Date	Datum zahájení události
Datum ukonceni	Date	Datum ukončení události

Tabulka 21 – komentare_udalosti

Tabulka komentare_udalosti uchovává všechny komentáře týkající se událostí.

Tabulka 21 - Komentáře událostí

Název	Vlastnost	Popis
Id_komentare_udalosti	Number (PK)	PK, Id komentáře událostí
Id_uzivatele	Number (FK)	FK, Id uživatele založení
Id_udalosti	Number (FK)	FK, Id události
Komentar	Clob	Text komentáře
Datum	Date	Datum vytvoření komentáře

Tabulka 22 – udalosti_prihlaseni

Tabulka udalosti_prihlaseni, uchovává seznam všech uživatelů, kteří jsou přihlášeni na jakoukoliv událost.

Tabulka 22 - Přihlášení na událostech

Název	Vlastnost	Popis
Id_udalosti_prihlaseni	Number (PK)	FK, Id události přihlášení
Id_uzivatele	Number (FK)	FK, Id přihlášeného uživatele
Id_udalosti	Number (FK)	FK, Id události
Id_Stavu	Number (FK)	FK, Id stavu přihlášení

Tabulka 23 – stav_udalosti

Tabulka – stav_udalosti definuje, kterých stavů mohou události nabývat.

Tabulka 23 - Stavy událostí

Název	Vlastnost	Popis
Id_stavu_udalosti	Number (PK)	PK, Id stavu události
Stav	NVarchar2(30)	Stav události