

Yii Framework

Yii Framework

Zadání bakalářské práce

Student: **Martin Chapčák**
Studijní program: B2647 Informační a komunikační technologie
Studijní obor: 2612R025 Informatika a výpočetní technika
Téma: **Yii Framework**
Yii Framework

Zásady pro vypracování:

Tvorba aplikací je v dnešní době zaměřena především na využití nejrůznějších frameworků. Jedním z nich je Yii Framework, vysoce výkonný PHP-framework pro vývoj aplikací podporujících Web 2.0. Cílem práce je popsat Yii Framework, navrhnout a naimplementovat ukázkové aplikace.

1. Seznamte se s Yii Frameworkem.
2. Vhodným způsobem popište Yii Framework.
3. Navrhněte a naimplementujte vhodné demonstrační aplikace v Yii Frameworku.
4. Aplikace otestujte.

Seznam doporučené odborné literatury:

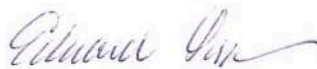
Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Petr Madecki**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



vedoucí katedry



prof. RNDr. Václav Snášel, CSc.

děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2012


.....

Rád bych poděkoval panu Ing. Petru Madeckemu za jeho rady a připomínky, které mi velice pomohly při tvorbě práce.

Abstrakt

Cílem této bakalářské práce je popsat PHP framework Yii a demonstrovat použití na ukázkové aplikaci. V teoretické části jsou popsány důvody, proč používat systém pro správu obsahu a proč používat framework Yii. Dále není opomenuta rozsáhlá architektura Yii. Praktická aplikace má podobu CMS systému, který je popsán v poslední části.

Klíčová slova: framework Yii, PHP, AJAX, MVC, tutoriál, bakalářská práce

Abstract

The aim of this Bachelor's work is a description of PHP framework Yii and to demonstrate the usage on the sample application. In the theoretical part of the work there are defined the reasons why to use the system of the "content administration" and why to use framework Yii. Again the extensive architecture Yii is not left out of consideration. The practical application has a shape of CMS system that is described in the last part.

Keywords: framework Yii, PHP, AJAX, MVC, tutorial, bachelor thesis

Seznam použitých zkratek a symbolů

AJAX	– Asynchronous JavaScript And XML
API	– Application Programming Interface
CMS	– Content Management System
CRUD	– Creat, Read, Update, Delete
CSS	– Cascading Style Sheets
HTML	– Hyper Text Markup Language
HTML5	– Hyper Text Markup Language version 5
IT	– Information Technology
PDO	– PHP data objects
PHP	– Hypertext Preprocessor
SEO	– Search Engine Optimization
SQL	– Structured Query Language
URL	– Uniform Resource Locator
WWW	– World Wide Web
WYSIWYG	– What You See Is What You Get

Obsah

1	Úvod	3
2	Internetové technologie	4
3	Vývoj webových aplikací	5
3.1	CMS	5
3.2	PHP	5
3.3	Framework	6
3.4	Framework Yii	7
4	Popis Yii	9
4.1	MVC	10
4.2	Databáze SQLite	12
4.3	Vývojové prostředí	14
4.4	Generátor kódu	14
5	Vývoj aplikace v Yii	16
5.1	Modules	16
5.2	Components	17
5.3	Widgety	18
5.4	Behavior	19
5.5	AJAX	20
5.6	Filter	22
5.7	Autentizace	23
5.8	Autorizace	24
5.9	Multijazyčnost	27

5.10 Správa adres	30
5.11 Témata	31
5.12 Cash	31
5.13 Logování	33
6 Implementace CMS	35
6.1 Module	35
6.2 User	35
6.3 Article	36
6.4 Photo	36
6.5 Případy užití	36
7 Závěr	40
8 Reference	44
Přílohy	45
A Příloha na CD/DVD	46

1 Úvod

Každý, kdo se někdy pokoušel tvořit internetové stránky, se potýkal s několika problémy. Jedna z těch nejzásadnějších otázek je pomocí jakých technologií vytvářet stránky a jak jejich obsah spravovat. Nejrozšířenější je v současné době tvorba dynamických stránek, skriptovaných v php. K vývoji většinou vývojář využívá framework, který poskytuje sadu užitečných funkcí, a práci tak ulehčuje. Jako téma bakalářské práce jsem si vybral framework Yii, který je řazen mezi ty nejvýkonnější. V kapitole 2 a 3 jsou popsány některé rozšířené technologie a důvody, proč využívat frameworky, hlavně pak Yii. Kapitola 4 se věnuje frameworku konkrétněji. Popisuje instalaci Yii a využití základních nástrojů. Kapitola 5 se zabývá jednotlivými komponenty a rámcově popisuje implementaci běžných součástí systémů. V poslední části 6 je popsán CMS systém, který byl vytvořen a jež používá komponenty popsané v kapitole 5.

2 Internetové technologie

Internet je bezesporu fenoménem začátku 21. století. Mnohé věci z běžného života se oběvují i na internetu, jiné se tam kompletně přesouvají. Obchody uzavřené prostřednictvím internetu, nevyjímaje nákup zboží denní spotřeby, dosahují takových objemů, že má tento virtuální svět citelný dopad na celosvětovou ekonomiku a přímo tak ovlivňuje miliardy lidí na planetě. Ačkoliv internet poskytuje nespočet různých služeb pro různá odvětví, nejpoužívanější je bezesporu HTTP protokol, konkrétně pak WWW stránky. V současné době je nejrozšířenější webový server Apache, který je nasazen na operačním systému Linux. Moderní internetová stránka využívá řadu nejnovějších technologií, které ji výrazně ovlivňují funkčně i vzhledově. Dynamické stránky, jejichž dynamický obsah se převede do statického HTML na serveru, dnes prakticky úplně nahradily své archaické čistě statické předky. V posledních letech, dochází k masivnímu vzestupu AJAX, HTML5 a CSS3 technologií, které dynamické stránky nezavrhuje, ale podstatně rozšiřují jejich možnosti. Nejrozšířenější jazyk pro tvorbu dynamických stránek je PHP, následuje ASP a výrazně méně používaná JAVA. Pro moderní internetové stránky se v posledních letech vžilo označení web 2.0. Výraz nepojednává o technických aspektech, ale o celkovém chápání webových stránek, jako o prostředí, ve kterém můžou obyčejní uživatelé obsah nejenom sdílet, ale i vytvářet. Nejlepším příkladem jsou sociální sítě, kde jejich vlastníky vytvořil aplikaci, do které vkládají příspěvky sami návštěvníci, kteří nejenom pasivně čtou, ale i sami vytvářejí obsah stránky. Ačkoliv je obsah, vytvářený pouze uživateli, na klasických korporátních webech absolutně nemyslitelný, moderní technologie umožňuje přiblížit se konceptu web 2.0 přidáním například diskuze pod jednotlivé články a také využívání, dnes velmi oblíbených, widgetů sociálních sítí, které jsou zároveň hodnotným marketingovým nástrojem. Výsledkem snahy, o skloubení rozšířeného PHP s požadavky týkajícími se směru web 2.0 s rychlým, bezpečným a efektivním vývojem webových aplikací, je množství php frameworků, které toto všechno splňují a zároveň vytváří vlastní standart pro vývojáře, stejně jako jeden z nejlepších frameworků Yii.

3 Vývoj webových aplikací

3.1 CMS

Možnost spravovat obsah internetových stránek je naprosto klíčová. Jejich majitel musí být schopen rychle a jednoduše měnit obsah, což umožňuje pouze kvalitní grafické rozhraní administrace srozumitelné i lidem z jiných odvětví, než IT. Existuje mnoho Open Source řešení, stejně tak jako nespočet komerčních produktů renomovaných softwarových společností. Každý systém by měl mít možnost editovat většinu textového obsahu, stejně tak jako práva uživatelů systém spravující. Mezi další důležité funkce patří možnost nahrání dodatečných modulů, správa kategorií, souborů, bannerů, jazykových mutací a případně i fotogalerie.[1]

3.2 PHP

PHP je skriptovací programovací jazyk a používá se především pro programování dynamických webových aplikací. PHP kód, v textovém tvaru, je zpracován serverem a ke klientovi je přenášén výstup skriptu. Syntaxe vychází převážně z jazyků Perl, C (C++) a Java. PHP je možné nasadit na různých operačních systémech. Umožňuje jednoduchou práci s textem, grafikou, soubory a databázemi. Dokáže pracovat s mnoha internetovými protokoly, například HTTP, SMTP, IMAP nebo POP3. Dnes je nejrozšířenější programovací (skriptovací) jazyk pro dynamické internetové stránky PHP. Vyniká jednoduchostí a mnoha funkcemi. Jeho rozšířenosti také napovídá využití v rámci LAMP serveru, což je velice častá implementace webového serveru, který obsahuje Apache, MySQL a PHP běžící na operačním systému Linux.[4]

3.2.1 Významné projekty implementované v PHP

- Software MediaWiki – použit při implementaci Wikipedie.
- phpBB – komplexní webové fórum.

- WordPress – publikační systém pro provoz blogů.
- phpMyAdmin – nejrozšířenější systém pro správu databáze MySQL.
- Yii Framework – framework pro tvorbu webových aplikací v PHP 5.

3.2.2 Výhody PHP

- Zaměřený na webové stránky.
- Velké množství funkcí, již v základní knihovně.
- Podpora několika databází.
- Strmá křivka učení.
- Rozšířenost - PHP nabízí prakticky každý poskytovatel web hostingů.
- Velké množství OpenSource projektů.
- Licence odvozená od BSD.

3.2.3 Nevýhody PHP

- Různé pořadí parametrů funkcí.
- Nejednoznačné pojmenování funkcí, například `strpos()`, `strpos()`, `str_replace()`, `str_pad()`
- Nízká podpora kódování Unicode.
- Neobsahuje debugovací nástroje.

3.3 Framework

Framework pro PHP je sada funkcí a tříd, které jednoduše řeší většinu požadavků na webové stránky a urychlují vývoj systému. Mezi další kladné vlastnosti se řadí fakt, že framework většinou funguje na principech architektury MVC a nutí tak programátora

k čistějšímu psaní kódu. Framework stojí za prakticky každou větší internetovou prezentací. Mezi nevýhody patří zejména nepříliš velký výkon. Další nepříjemnosti může způsobit dlouhá doba potřebná k naučení a někdy i obtížnější úprava řešení oproti původnímu návrhu.

3.4 Framework Yii

Yii je mladý a výkonný Open Source PHP framework určený pro vývoj webových aplikací. Yii, stejně jako ostatní moderní frameworky, je objektově orientovaný. Každá komponenta je snadno rozšiřitelná a není pevně vázaná na ostatní komponenty. Oproti jiným frameworkům však vyniká vysokou znovupoužitelností, efektivností, přehlednou dokumentací a podporou web 2.0 technologií, čemuž mimo jiné nasvědčuje použití jQuery jako výchozí JavaScriptové knihovny. Jako slabé místo se považuje málo komentovaný kód. V říjnu 2008 představil Qiang Xue , který dříve pracoval na frameworku Prado, nový framework Yii, který upoutal pozornost vývojářů, protože výkonem překonal všechny PHP frameworky a zároveň obsahoval velké množství nejrůznějších funkcí. Oficiální verze 1 byla vydána v prosinci téhož roku.[2] Yii se inspiroval projekty:

- Prado
- Ruby on Rails
- Symphony
- Joomla

Softwarové komponenty frameworku jsou poskytovány na základě BSD licence. V současné době se připravuje dlouho očekávaná verze 2.0.

3.4.1 Vlastnosti

- Využívá architektury MVC

- Do databáze přistupuje pomocí DAO a Active Record
- Obsahuje jQuery knihovnu pro práci s JavaScriptem
- Jednoduchá validace formulářových dat
- Autentizace a autorizace
- Internacionalizace
- Automatické generování WDSL pro webové služby
- Cache
- Zpracování chyb

4 Popis Yii

4.0.2 Instalace

Instalace Yii je velice jednoduchá. Jako první se stáhne instalační balíček a do adresáře webového serveru se extrahují zdrojové kódy frameworku. V adresáři requirements je script, který ověří, zda webový server obsahuje vše nezbytné. Soubor index.php slouží jako bootstrap soubor, který při každém požadavku vytváří, konfiguruje a zavádí aplikaci ve formě objektů. Adresář protected obsahuje zdrojové kódy aplikace a také provozní složky.

4.0.3 Adresářová struktura

- runtime – dočasné soubory vytvořené během práce aplikace
- extensions – rozšíření frameworku od třetích stran
- modules – každý modul je ve vlastní podsložce
- config – konfigurační soubory
- controllers – třídy kontrolérů
- views – soubory pohledů
- models – třídy modelů

4.0.4 Komponenty aplikace

Framework obsahuje množství nejrůznějších komponent, přičemž je každá komponenta rozšiřitelná. Mezi základní komponenty, jež využívá většina aplikací patří:

- CAssetManager – správa souborů
- CAuthManager – poskytuje autorizaci, implementuje správu přístupu založenou na rolích

-
- CCache – poskytuje cachování
 - CClientScript – umožňuje dynamicky přilinkovat JavaScript k renderované stránce
 - CDbConnection – reprezentuje připojení k databázi
 - CErrorHandler – zpracovává chyby
 - CHttpRequest – reprezentuje HTTP dotaz a pracuje se soubory cookie
 - CUrlManager – zpracovává URL adresu
 - CSecurityManager – poskytuje bezpečnostní prvky
 - CWebUser – reprezentuje aktuálního uživatele

4.0.5 Dokumentace

Na oficiálních stránkách Yii (<http://www.yiiframework.com/>) se nachází rozsáhlá dokumentace, která se zabývá nejenom popisem jednotlivých tříd, ale i ukázkami řešení častých úloh. Při řešení problémů lze jako velký zdroj informací využít taktéž oficiální fórum (<http://www.yiiframework.com/forum/>).[2]

4.1 MVC

Yii, stejně jako většina ostatních webových frameworků implementuje návrhový vzor MVC, který ve své podstatě odděluje business logiku od zobrazených dat.

- Model - Reprezentuje práci s daty, spolu s validačními pravidly. Většinou komunikuje s databází.
- View - Zobrazuje uživateli data, která jsou součástí layoutu, text, formuláře, odkazy.
- Controller - Zprostředkovává komunikaci mezi View a Model. Zpracovává požadavky uživatele.

Ještě, než se ke zpracování dotazu dostane MVC, požadavek je předán hlavní komponentě zvané Application, která funguje jako front-controller. Typický požadavek z pohledu zpracování vypadá takto:

- Uživatel pošle požadavek na konkrétní stránku.
- Soubor index.php, v kořenovém adresáři webového serveru, vytvoří instanci Application, která tento požadavek přebere a zpracuje pomocí urlManageru.
- Application zavolá patřičný Controller a konkrétní metodu, která představuje požadovanou akci a předá jí zbylé parametry.
- Metoda zpracuje požadavek, pro práci s daty volá Model a výsledek obdrží patřičný View.
- View zobrazí data, která předal Model, nejčastěji pomocí widgetu.
- View volá patřičný layout, který ho obalí další HTML strukturou a výsledek je poslán uživateli.

Application je součástí frameworku, proto není nutná implementace, avšak je vhodné nastavit některé její vlastnosti. V nastavení můžeme taktéž přidat komponenty (nejčastěji CurlManager) a také je konfigurovat tak, aby vyhovovaly naší aplikaci. Nastavení se provádí v souboru `/protected/config/main.php`. Vzhledem k tomu, že implementuje návrhový vzor Singleton je všude přístupná přes `Yii::App()`. Model, View a Controller je potřeba implementovat pomocí rozšíření tříd pro ně určených.[9]

4.1.1 Model

Potomek třídy Cmodel, který reprezentuje data. Data jsou reprezentovány nejčastěji jako jeden řádek v tabulce, model pak implementuje CActiveRecord, potomka Cmodel. Atributy reprezentují jednotlivé sloupce řádku. Třída musí obsahovat metodu rules, která obsahuje pravidla pro validaci dat. Pro specifické formuláře, které jsou speciálním případem Cmodel implementujeme CformModel.

4.1.2 View

Jedná se o soubor, který vedle HTML kódu obsahuje také php script pro zobrazení dat, které předa Controller. Je umístěn ve složce se stejným názvem jako má controller, který ho volá. Každý view je při renderování zasazen do layoutu, který obsahuje části stránky jako hlavička, menu, patička. Nejčastější zpracování dat je pomocí volání widgetu, který data přehledně zobrazí a minimalizuje tak PHP kód obsažený v každém view.

4.1.3 Controller

Potomek třídy Ccontroller, jež je instancován třídou Application, která volá jeho metody v závislosti na požadované akci. Každá akce musí mít v názvu metody action, tedy akce view odkazuje na metodu actionView(). Pro snadnější práci s GET parametry je možné tyto proměnné specifikovat jako parametry metody. Controller ArticleController a metoda actionView se volá přes URL domena.cz/article/view.

4.2 Databáze SQLite

4.2.1 Výkon

SQLite je malý relační databázový systém. Na rozdíl od ostatních databází, které běží jako server, je tato databáze reprezentována malou C knihovnou, která se přilinkuje k aplikaci. Není tedy nutná stálá režie pro běžícího démona serveru, prostředky jsou alokovány pouze, když se databáze aktivně využívá. Běžící databáze by měla zvýšit zaplnění operační paměti jen o 25kB. Databáze je uložena v souboru .dbm a může být sdílena multiplatformně mezi počítači. Můžeme jí využít v programech napsaných v jazycích C, C++, Java, Delphi, PHP, Perl, Python. Rychlost databáze je extrémní a téměř vždy překoná už tak velice rychlou databázi MySQL. Implementuje téměř celý standart SQL92, vydaný v roce 1992 jako rozšíření specifikace SQL z roku 1989.[13]

4.2.2 Využití v praxi

Mezi významné společnosti, které využívají tuto databázi patří Adobe v aplikaci Adobe Photoshop Lightroom a Adobe Reader, Airbus v letadlech typu A350 XWB, Apple ve většině svých aplikací, Firefox ve svém prohlížeči a emailovém klientu, Google v operačním systému pro mobilní telefony Android.

4.2.3 Omezení

Jako možnou nevýhodu, nebo vlastnost, lze brát fakt, že databáze je "typeless", což znamená, že do pole typu int můžeme vložit text. Tato na první pohled zásadní chyba má v důsledku jen malé dopady, neboť se validace dat a ošetření vkládaných hodnot většinou provádí již na úrovni doménové logiky v aplikaci. Jediným případem, kdy databáze kontroluje datový typ je vkládání do pole typu int deklarovaném jako PRIMARY KEY. SQLite nemá ambice stát se robustní korporátní databází, naopak se profiluje k využití v malých aplikacích, kde je instalace klasické serverové databáze nesmyslná a ukládání dat přímo do textového souboru značně neefektivní. [12]

4.2.4 Rozšířenost

Podpora SQLite databáze u web hostingových společností je stejně rozšířená, jako podpora MySQL a stává se tak nepsaným standardem, čemuž podstatně napomáhá jednoduchá instalace a nízké vytížení systémových zdrojů. Mezi hostingsy, které nabízejí SQLite databázi patří gigaserver.cz, wedos.cz, ebola.cz, hukot.cz, bezobav.cz.

4.2.5 Nasazení

Z výše uvedených informací vyplývá, že databáze je vhodná pro nasazení na malých internetových projektech. Jelikož je databáze tvořena skutečně jedním souborem, je možné ji extrémně jednoduše nejenom zálohovat, ale i sdílet a posílat napříč vývojáři a projekty. Yii používá k přístupu k databázi knihovnu PDO, která je v PHP dostupná od verze 5.1.

Tato knihovna umožňuje pracovat s různými databázemi při využití jedné sady funkcí. Používání databáze SQLite deklarujeme v `/protected/config/main.php` v poli `components`.

```
'db'=>array(
    'connectionString'=>'sqlite :/ wwwroot/blog/protected/data/blog.db',
    'tablePrefix'=>'tbl_',
),
```

Výpis 1: Připojení k SQLite

Pro využití jiné databáze, například MySQL, stačí nastavit:

```
'db'=>array(
    'connectionString' => 'mysql:host=localhost;dbname=blog',
    'emulatePrepare' => true,
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
    'tablePrefix' => 'tbl_',
),
```

Výpis 2: Připojení k MySQL

4.3 Vývojové prostředí

Jedno z nejlepších vývojových prostředí pro PHP je Zend studio, avšak vzhledem k jeho ceně je mnohdy lepší se u menších projektů přiklonit k Eclipse nebo NetBeans, které jsou zdarma. Poslední jmenovaný NetBeans poskytuje vše potřebné a proto je vhodnou volbou.

4.4 Generátor kódu

4.4.1 Yii

Yii poskytuje, dnes téměř nutný, generátor kódu. Pro vytvoření nové webové aplikace yiiCMS je potřeba využít generátoru yiic a v příkazové řádce zadat

```
% YiiRoot/framework/yiic webapp WebRoot/yiicms.
```

Generátor vygeneruje základní adresářovou strukturu i s jednoduchou aplikací, jež slouží jako odrazový můstek mnoha jiných projektů.

4.4.2 Gii

Pro generování vlastního kódu, spojeného s CRUD operacemi je nutné využít Gii, jež je pokročilejší nástroj pro generování kódu s přehledným grafickým rozhraním. Využití tohoto generátoru není ve vygenerované aplikaci nastaveno, proto je nutné ji zavést v application configuration v adresáři `/protected/config/main.php` přidáním:

```
'modules'=>array(
    'gii' =>array(
        'class'=>'system.gii.GiiModule',
        'password'=>'heslo',
    ),
),
```

Výpis 3: Zavedení Gii

Poté je generátor dostupný na adrese `http://domena.cz/index.php?r=gii`. Jako první je nutné vygenerovat model pro každou tabulku v databázi. Po kliknutí na Model generator je možné vložit název tabulky v databázi, a vytvořit tak model pro danou tabulku. Tento krok je nutné opakovat pro všechny ostatní tabulky. Nový model tabulky `yii.article` je uložen v `/protected/model/article.php` Stránka Crud Generator slouží pro vytvoření controller a view pro daný model. Stačí zadat pouze název modelu, vytvořeném v Model generator. Controller pro práci s article lze najít v `/protected/controller/articleController.php`. Soubory umístěné v `/protected/view/article/` jsou `index.php`, `view.php`, `create.php`, `update.php`, `admin.php`, `_view.php`, `_form.php`, `_search.php`. Pokud je volán controller article a action index, jsou data předána do widgetu v `index.php` a každý záznam je zvlášť renderován pomocí `_view.php`. Pokud přijde dotaz na action update, controller renderuje `update.php`, který získává formulář pro data z `_form.php`. Je až zarážející, v jak krátkém čase lze bez programování, jen za pomoci generátoru kódu vytvořit aplikaci, která korektně pracuje s námi vytvořenou databází. Aplikace v této podobě pravděpodobně nespĺňuje všechny požadované funkční požadavky, avšak je dobrým základem, čímž odbourává zdlouhavou a monotónní implementaci CRUD. Zkušený programátor je schopen tento funkční skelet, pro malou aplikaci o velikosti do 10 tabulek, naimplementovat za 10 minut a rapidně tak urychlit vývoj malých dynamických projektů.

5 Vývoj aplikace v Yii

Přestože generátor kódu odvede určitou práci za vývojáře, je nutné velkou část kódu výrazně předělat. Z původní vygenerované aplikace většinou zbyde jen pár řádek kódu, které slouží spíše jako skelet, do kterého se musí implementovat požadovaná funkčnost. Rozsáhlejší webová aplikace mnohdy požaduje pokročilejší funkčnost a proto je většina funkcionalit zprostředkována skrz moduly, které často vyžadují vlastní rozšíření. Mezi rozsáhlejší aplikace zajisté spadá i CMS systém, jež je v této bakalářské práci implementován pomocí frameworku Yii. V této kapitole je zjednodušeně popsáno mnoho běžných problémů, související s jednotlivými komponenty, jež se objevují při vývoji CMS.

5.1 Modules

Samostatná komponenta, která obsahuje *model*, *view*, *controller*. Do velké míry se podobá kořenové aplikaci v `/protected/`, avšak nemůže být samostatně nasazena, pouze jako součást jiné aplikace a to zkopírováním do `/protected/modules/`. Modul se používá většinou při implementaci velkých aplikací nebo samostatných funkčních celků. Typickým příkladem je implementace CMS, který byl vyvinut jako modul. V jiné aplikaci je možné využít modul CMS, pokud bude nahrán do adresáře `/protected/modules/`. Přístup k modulu je pomocí URL `domena.cz/cms/controller/action`. Nejlepší volbou pro tvorbu modulu je použití *module generator* v sadě nástrojů Gii. Pokud je modul vytvořen, je potřeba ho zpřístupnit aplikaci. Toto nastavení, stejně jako většina ostatních se provádí v `/protected/config/main.php`, kde se vloží řádek `'modules'=>array('cms')`.

Modulu je možné předat libovolný počet parametrů. Předání parametru, který specifikuje seznam využívaných jazykových verzí je možné provést pomocí pole prvků.

```
'modules'=>array(
    'cms'=>array(
        'languages'=>array('cs','en'),
    ),
),
```

Výpis 4: Předání parametru modulu

Toto řešení je jednoduché a přehledně specifikuje nastavení pro každý modul. K seznamu jazyků je možné přistupovat odkudkoliv pomocí `Yii::app()->getModule('cms')->languages`. Pokud je zapotřebí mít určité data globálně, což je konkrétně případ jazykových verzí, implementace globální komponenty je efektivnější. Každý vygenerovaný modul obsahuje standardní controller default. Změna přednastaveného controlleru na ArticleController se provádí v `/protected/modules/cms/cmsModule.php` přidáním atributu `public $defaultController = 'article'`.

`CmsModule.php` je inicializační třída modulu, která je potomkem třídy `CwebModule`. Obsahuje veřejnou metodu `init()`, která je spuštěna ještě před voláním konkrétního controlleru. V controlleru se layout nastavuje inicializací atributu

`public $layout='/layouts/column2'`. Toto nastavení bude využívat soubor `/protected/modules/cms/views/layouts/column2.php`. Pokud má controller renderovat view podle layoutu z `/protected/views/layouts/column2.php`, je nutné změnit hodnotu layout na `'//layouts/column2'`.

5.2 Components

Potomek třídy `Ccomponent` nebo `CApplicationComponent`. Nachází se v adresáři `/protected/components/`. Je využíván v případech, kdy je potřeba využívat nějakou specifickou, vývojářem definovanou, funkcionalitu opakovaně nebo na několika místech aplikace. Pokud je potřeba sdílet jazykové nastavení napříč moduly a komponenty, je vhodné využít vlastní komponentu `lang`, která bude obsahovat atribut `public $languages`. Zavedení komponenty se provádí v application configuration

```
'components'=>array(
    'lang'=>array(
        'class'=>'ext.components.Lang',
        'languages' => array('cs','en') ,
    ),
),
```

Výpis 5: Přidání vlastní komponenty

Pokud je komponenta součástí modulu CMS, její zavedení je identické, avšak cesta ke třídě se změní z `ext.components.Lang` na `cms.components.Lang`. Takto nastavená komponenta

je globální a tedy dostupná v celé aplikaci. Vytvořit komponentu pouze v rámci modulu CMS je možné a provádí se její inicializací v rámci inicializace modulu a ne komponent.

Komponenta umožňuje automatické využití getteru a setteru. V takovém případě stačí implementovat metodu `getLanguages()` a `setLanguages($value)`. Vrácení atributu `languages` probíhá naprosto identicky, jako v případě bez implementovaného getteru a setter, a to voláním `Yii::app()->lang->languages`.

Widgety a Componenty od jiných vývojářů nebo ty, které mají předpoklad použití i v jiných aplikacích zpravidla umístíme do `/protected/extensions/`, vlastní pak do `/protected/components/`

5.3 Widgety

Potomek `CWidget`, který je součástí `view`, jehož úkolem je zobrazit určitý prvek či funkcionalitu uživateli. Velkou mírou napomáhá k znovupoužitelnosti a transformaci kódu, protože jeden `view` rozděluje na několik nezávislých prvků. Typickým příkladem může být například menu v podobě widgetu. Při volání předáme v parametru všechny odkazy které má menu obsahovat a widget se postará o jejich formátování. Toto menu pak můžeme použít na jiných místech nebo i jiných aplikacích a vždy stačí pouze předat seznam odkazů v parametru. Mezi další typické widgety patří zobrazení dat z databáze, kalendář a formulář pro změnu jazyka. Widget se do `view` vloží pomocí `<?php $this->widget('ext.components.menuWidget'); ?>` Jako widget může sloužit popup okno.

```
class PopUp extends CWidget
{
    public $id='popup';
    public $text='PopUp..message';
    public function run()
    {
        $this->render('PopUp',array('id'=>$this->id,'text'=>$this->text));
    }
}
```

Výpis 6: Vlastní widget

Třída `PopUp` je uložena v `/protected/components/`. Ze zdrojového kódu je patrné, že renderuje `view` `PopUp.php`, který je uložen v `/protected/components/views/`

a pracuje s proměnnými *\$id* a *\$text*. View `PopUp.php` může například obsahovat `<div id='<?php echo $id;?>' style='opacity: 0.5;position: absolute;'><?php echo $text;?></div>`

Tento widget je možné použít ve všech **view** celé aplikace velice jednoduše a to vložením řádku `$this->widget('ext.components.PopUp', array('id'=>'popup','text'=>'nedostatečné oprávnění'))`. Framework poskytuje vlastní sadu widgetů, které jsou dostupné přes `zii.widgets.*`. Widget pro renderování menu se volá pomocí `$this->widget('zii.widgets.CMenu', array('items'=>$array))`. Widget, pro výpis dat v přehledné tabulce, renderuje každý řádek podle `_view.php`, což je klasický **view**.

```
$this->widget('zii.widgets.CListView', array(
    'dataProvider'=>$dataProvider,
    'itemView'=>'_view',
));
```

Výpis 7: Widget ve view

5.4 Behavior

Rozšiřuje funkčnost objektu a to přidáním dalších metod. Jedná se tedy téměř o ekvivalent k dědičnosti, ale poskytuje několik benefitů navíc. Například `model article` potřebuje funkcionalitu pro formátování data přidání příspěvku. První možnost je implementovat metodu přímo do těla třídy. Toto řešení má několik nevýhod. Stejně metody není možné použít v ostatních modelech a pokud by se při změně struktury databázové tabulky generoval nový `model` přes `Gii`, stará třída bude přepsána a implementace ztracena. Rozšíření rodičovské třídy zavede funkcionalitu pro všechny ostatní modely, i ty které s datem vůbec nepracují a vnese do celého kódu zmatek. O něco lepší řešení je vytvoření nové třídy `FormatDate`, ze které bude dědit `model article`.

Class article extends CActiveRecord, FormatDate

Vzhledem k zachování čistoty kódu může být přidání třídy `FormatDate` mírně matoucí, celé řešení má však lehké omezení v tom, že funkcionalita `FormatDate` nemůže být přidána dynamicky a musí být děděná z každé třídy, jež ji má obsahovat. Nejeftektivnější je vytvořit třídu `FormatDate` dědící z `Cbehavior` nebo v případě `behavior` modelu z `CactiveRecordBehavior`. Třidu `FormatDate` obsahující potřebné metody můžeme připojit přímo

v kódu `$component->attachBehavior('artDateFormatBeh',new FormatDateBehavior)` v application configuration je možné přilinkovat behavior ke každé komponentě v `components`

```
'behaviors'=>array(
    'art_dateformat_beh'=>array(
        'class'=>'ext.components.FormatDateBehavior.',
    ),
),
```

Výpis 8: Přidání behavior v main.php

Pokud je třída, ke které chceme behavior přidat, potomkem jedné ze tříd `CController`, `CFormModel` nebo `CActiveRecord`, lze připojit behavior přepsáním metody `behaviors()`.

```
public function behaviors()
{
    return array(
        'art_dateformat_beh'=>array(
            'class'=>'ext.components.FormatDateBehavior',
        ),
    );
}
```

Výpis 9: Specifikace behavior

5.5 AJAX

5.5.1 Úvod

AJAX přináší benefit v tom, že aktualizuje data na stránce bez nutnosti znovu načtení stránky. Využívá se u moderních stránek hlavně kvůli možnosti poskytnutí příjemnějšího uživatelského rozhraní, ale i kvůli snížení přenášených dat mezi serverem a klientem. Dnes se již, alespoň v malé míře, využívá ve většině stránek a v některých tak výrazně, že rozdíl mezi webovou a desktop aplikací je jen nepatrný. Velké využití AJAX technologie je hnacím motorem pro web 2.0, stejně jako HTML5.[14]

5.5.2 AJAX v Yii

Yii poskytuje mnoho nástrojů pro práci s AJAX, které jsou většinou úzce spojeny s JavaScript knihovnou jQuery. Nejčastější použití je při mazání záznamu. Práci s HTML a formuláři usnadňuje ve view statická třída CHtml.[14]

```

CHtml::ajaxLink(
    'smazat',
    array(
        'photo/delete',
        'id' => $data->id
    ),
    array(
        'type' => 'POST',
        'complete' => 'function () {$.fn.yiiListView.update(" listt ");$("#myDiv").removeClass("loading");}'
    ),
    array(
        'confirm' => 'Are_you_sure_you_want_to_delete_this_item?'
    )
);

```

Výpis 10: Generování AJAX odkazu pro smazání záznamu

Po kliknutí na odkaz "smazat" je vyvoláno potvrzovací okno, které se ptá, zda chceme záznam smazat. Po jeho potvrzení je proveden AJAX dotaz typu POST na controller photo, action delete a předá parametr id. Server tento požadavek vyhodnotí a smaže prvek z databáze. Poté stránka obdrží informaci o úspěšném odeslání dotazu a aktualizuje data v ListView opět pomocí AJAX. Mezi další časté příklady využití AJAX patří aktualizace určitých údajů. V Yii je tento úkon velice jednoduchý a vývojáři zabere maximálně pár minut. View zobrazující data obsahuje:

```

<div id="ajaxdata">
    <?php $this->renderPartial('_index',array('text'=$text)); ?>
</div>
<?php
    echo CHtml::ajaxButton(
        "Update",
        CController::createUrl('controller/UpdateAjax'),
        array(
            'update' => '#ajaxdata'
        )
    );
?>

```

Výpis 11: AJAX tlačítko pro obnovení výpisu

Voláním `renderPartial('_index')` se renderuje pouze soubor `_index.php`, voláním `render('_index')` se vyrenderuje celý layout, jehož součástí je i `_index.php`.

```
public function actionUpdateAjax()
{
    $data['text'] = 'data_uploaded_by_ajax';
    $this->renderPartial('_index', $data, false, true);
}
```

Výpis 12: Action podporující AJAX

Po kliknutí na tlačítko update se pošle dotaz na action `UpdateAjax` a server vrátí HTML výsledek, který je vložen do divu s id `ajaxdata`. AJAX požadavek lze navázat na standartní události formulářového prvku. Aplikaci, která po vybrání položky z dropdown listu odešle automaticky data na server musí obsahovat:

```
<?php
echo $form->dropDownList(
    $data,
    'permission',
    Permission::getPermissionLabel(),
    array(
        'onchange' => CHtml::ajax(
            array(
                'type' => 'POST',
                'url' => CController::createUrl('user/updateajax'),
                'update' => '#ajaxdata',
            )
        )
    )
);
?>
```

Výpis 13: Dropdownlist využívající AJAX

5.6 Filter

Část kódu, který je vykonáván před, nebo po volání konkrétní action v controlleru. Nejčastější využití filtrů je nepochybně z důvodu autorizace, kdy systém vyhodnotí, zda má daný uživatel ke konkrétní action přístup, či nikoliv. Jako další možné použití se nabízí logování přístupů k jednotlivým actions nebo měření výkonosti. Každý controller může mít několik filtrů a každý filtr může být přiřazen ke všem nebo pouze k některým actions. Toto nastavení se provádí v metodě `filters()` controlleru.

```

public function filters ()
{
    return array(
        'AccessControl',
        'postOnly,+_edit,_create',
        array(
            'application.filters.PerformanceFilter,-_edit,_create',
        ),
    );
}

```

Výpis 14: Přidání filtru do controlleru

Filtr `accessControl`, který je tvořen metodou `FilterAccessControl` je přiřazen ke každé action. Filtr `PostOnly` je volán pouze v případě action `edit` a `create`. Externí filtr `PerformanceFilter`, který je uložen v `/protected/filters/PerformanceFilter.php` se aplikuje pouze v případě, že není volána action `edit` nebo `create`. Externí filtr musí být potomek třídy `Cfilter`. Obsahuje metodu `preFilter`, která se provádí před voláním action v controlleru a metodu `postFilter`, která se aplikuje po vykonání volané action. Této vlastnosti lze využít například při měření doby potřebné pro vykonání action. V metodě `preFilter` se získá `timestamp()`, který se v metodě `postFilter` odečte od aktuálního `timestamp()`, čímž se získá počet sekund, po které byla vykonávaná action.

5.7 Autentizace

Ověření totožnosti uživatele se provádí nejčastěji pomocí přihlašovacího jména a hesla. V Yii je již připravena implementace interface `IwebUser`, která je dostupná pomocí `Yii::app()->user` a reprezentuje uživatele aktuálně pracujícího s aplikací. Aby bylo možné tuto komponentu využívat, `application configuration` musí obsahovat v poli `components` `'user'=>array('loginUrl'=>array('site/login'))`. Přihlášení uživatele nejčastěji zprostředkovává formulář dědící z `CformModel`. Ten po validaci vstupních dat vytvoří autentizační instanci a v případě úspěšného ověření uživatele provede jeho přihlášení do systému.

```

$identity=new UserIdentity($username,$password);
if ( $identity->authenticate()
    Yii::app()->user->login($identity);

```

Výpis 15: Autentizace uživatele

O celý proces autentizace se stará převážně potomek třídy `CuserIdentity`, který obsahuje metodu `authenticate`.

```
public function authenticate()
{
    $user=User::model();
    $user = $user->find('LOWER(login)=?',array(strtolower($this->username)));
    if ($user===null)
        $this->errorCode=self::ERROR_USERNAME_INVALID;
    else if (!$user->validatePassword($this->password))
        $this->errorCode=self::ERROR_PASSWORD_INVALID;
    else
    {
        $this->_id=$user->id;
        $this->username=$user->username;
        $this->errorCode=self::ERROR_NONE;
    }
    return $this->errorCode===self::ERROR_NONE;
}
```

Výpis 16: Ověření hesla pro přihlášení

Metoda vyhledá uživatele podle uživatelského jména a poté zavolá validaci hesla, což vložené heslo převede na hash a porovná s hodnotou z databáze.

Kvůli zvýšení bezpečnosti, v případě úniku dat z databáze, se k heslu přidává takzvaná sůl. Sůl je řetězec znaků, pro každého uživatele unikátní, který ztěžuje slovníkový útok a výrazně zvyšuje bezpečnost hashovacích algoritmů.

5.8 Autorizace

Autorizace zajišťuje oprávnění/přístup k určitým částem systému nebo funkcím. Před samotnou autorizací je většinou vyžadována i autentizace, avšak ta není vždy nutnou podmínkou.

5.8.1 Standartní

Standardní autorizace uživatele probíhá pomocí zabudovaného filtru `accessControl`. Filtr se konfiguruje velice jednoduše a to přepsáním metody `accessRules()` v controlleru.

```
public function accessRules()
{
```

```

return array(
    array(
        'deny',
        'actions'=>array('create', 'edit'),
        'users'=>array('?'),
    ),
    array(
        'allow',
        'actions'=>array('delete'),
        'roles'=>array('admin'),
    ),
    array(
        'deny',
        'actions'=>array('delete'),
        'users'=>array('*'),
    ),
);
}

```

Výpis 17: Oprávnění pro controller

Pravidla se vykonávají ve sledu za sebou a pokud je splněno jedno z pravidel, další se již neaplikuje. Action create a edit nemohou být vykonány nepřihlášeným uživatelem. Action delete může volat pouze uživatel s username admin a pokud není přihlášen, pak je akce zakázána pro všechny uživatele přihlášené i nepřihlášené. V poli users lze specifikovat přihlášeného uživatele hodnotou @. Tyto pravidla předá metoda *filterAccessRules()* filtru CaccessControlFilter, který data zpracuje a provede autorizaci.

5.8.2 Pokročilá

Autorizační údaje u dynamického systému, hlavně pak CMS, nemusí být vždy nastavována v metodě *accessRules()*. Pokud jsou data o autorizaci uložena v databázi, je vhodná implementace vlastního filtru. Ve speciálních případech může být nastavené oprávnění nejenom pro controller a action, ale i v závislosti na předaném parametru. Pokud má uživatel přístup pouze ke svým článkům, je důležité, aby bylo kontrolováno, kdo je vlastníkem článku s volaným id.

```

public function filterUserAccess($filterChain)
{
    $othercontent = $this->othersContent();
    $permission = Permission::model()->getPermission(Yii::App()->user->id,$this->getId(),
        $othercontent);
}

```

```

$rule = array(
    'r'=>array('index','view','admin'),
    'w'=>array('index','view','admin','create','update','delete'),
    'x'=>array('x'),
);
$this->array_merge_unique($rule['r'], $this->action['r']);
$this->array_merge_unique($rule['w'], $this->action['w']);
$filter =new CmsAccessControlFilter;
$filter ->set($rule[$permission]);
$filter ->filter( $filterChain );
}

```

Výpis 18: Oprávnění které jsou uloženy v databázi

Na příkladu filtru `UserAccess` je patrné, že jako parametr pro vyhledání oprávnění je id přihlášeného uživatele, název `controlleru` vrácený metodou `getId()` a údaj, zda je uživatel vlastník obsahu v proměnné `$othercontent`. Dále je specifikováno, které actions jsou dostupné při oprávnění pro čtení (r) nebo zápis (w). `$this->action` obsahuje akce specifikované konkrétním `controllerem`. Objektu `CmsAccessControlFilter` se poté předá pouze seznam actions, ke kterým je uživatel oprávněn přistupovat. Filtr tento seznam vyhodnotí a přístup povolí nebo zakáže.

5.8.3 Predikční

U takto složitých pravidel přístupu, založených i na informaci, kdo je vlastníkem přistupovaného prvku je vhodné sdělit uživateli, zda má oprávnění pro přístup ke stránce ještě předem, než na ni skutečně vstoupí. Přístup na jinou stránku je vždy spojen se změnou URL adresy, která se generuje pomocí `$this->createUrl('cms/')`. Řešením je získání informace, zda má uživatel přístup k požadované URL, při každém volání `createUrl()`. Stačí tedy přepsání metody `createUrl()` `controlleru`.

```

public function createUrl($route,$params=array(),$ampersand='&')
{
    if (isset($params['id']))
        $othercontent = $this->othersContent($params['id']);
    else
    {
        $othercontent = $this->othersContent();
        $routeInArray = explode("/", $route);
        $controller = array_shift($routeInArray);
        $action = array_shift($routeInArray);
    }
}

```

```
$permission = Permission::model()->getPermissionForAction(Yii::App()->user->id,
    $contoller,$othercontent,$action);
if (!$permission && !is_null($permission))
{
    $route = '#permission';
    $route .= rand();
    return $route;
}
return parent::createUrl($route,$params,$ampersand);
}
```

Výpis 19: Uprava URL parseru

Tato změna zajistí, že pokud uživatel nemá přístup ke stránce, funkce vrátí aktuální URL. Po kliknutí na odkaz nedojde k přesměrování, ale JavaScript, který je součástí widgetu, zobrazí autorizační varování. Změna metody *createUrl()* tak zajistí, že jakýkoliv odkaz bude automaticky procházet autorizací. Pokud není autorizace vyžadována, stačí využít zabudovanou metodu *Yii::app()->createUrl()* namísto *\$this->createUrl()*.

5.9 Multijazyčnost

Pokud jsou internetové stránky u kterých se předpokládá, že část návštěvníku bude ze zahraničí, neměla by se opomínat možnost změnit český jazyk, na jazyk uživateli srozumitelnější, zpravidla angličtinu, francouzštinu nebo španělštinu, Multijazyčnost je dnes na webových stránkách velice častá a proto Yii poskytuje jednoduché nástroje pro tvorbu jazykových mutací. Nejdůležitější je nastavení zdrojového jazyka a jazyka překladu, který je často nastavován dynamicky pomocí cookies nebo GET metody. Nastavení je vhodné provést v *init()* metodě controlleru.

```
Yii :: app()->sourceLanguage = 'cs';
Yii :: app()->setLanguage($_GET['language']);
```

Výpis 20: Nastavení jazyka

5.9.1 Statický překlad

V adresáři */protected/messages/* je nutné vytvoření složky s názvem jazyka pro každý jazyk, do kterého je vyžadován překlad. Pokud bude nastavený jazyk *setLanguage('en')*,

musí existovat složka `/protected/messages/en/`. V této složce se vytvoří soubory, které budou obsahovat překlady jednotlivých částí aplikace. Soubor `usermenu.php` obsahuje:

```
return array(
    'Home'=>'Home',
    'O_projektu'=>'About',
    'Administrace'=>'Admin',
    'Přihlasit'=>'Login',
    'Odhlásit_{user}'=>'Logout_{user}',
);
```

Výpis 21: Překlad menu

Po zavolání překlada `Yii::t('usermenu','Odhlásit (user)', array('user'=>'administrator'))` funkce vrátí `"Logout (administrator)"`. Pokud bude jazyk překlada nastaven na stejný jazyk, jako zdrojový, pak se překlad neprovede a funkce vrátí `"Odhlásit (administrator)"`. Pokud je potřeba přeložení velkých bloků textu, je vhodné celý `view` přeložit staticky a zkopírovat do složky s názvem jazyka. Soubor `/protected/view/default/index.php` se přeložený zkopíruje do `/protected/view/default/en/index.php`. Při volání `$this->render('view')` v `controlleru` se automaticky provede volba nahrání vhodného `view` souboru.

5.9.2 Dynamický překlad

Pokud je použit pro zobrazení dat z databáze `widget`, získává popisky k datům z metody `attributeLabels()` příslušného modelu. Tyto popisky jsou však statické a neumožňují překlad. Řešení je přepsat metodu `attributeLabels()` tak, aby automaticky načítala popisky ze souborů.

```
public function attributeLabels ()
{
    $labels = $this->getAttributes();
    foreach ($labels as $name=>$l){
        $arr[$name] =Yii::t('model', get_called_class () . '.' . $name);
    }
    return $arr;
}
```

Výpis 22: Automatický překlad popisků

Toto řešení má jeden nedostatek a to ten, že pokud není nutný překlad (jazyk je shodný se zdrojovým), zobrazí se popisek i s názvem třídy. Naštěstí lze tuto situaci lehce vyřešit. Je nutné zajistit, aby došlo k překladu vždy, nastavením `Yii::app()->sourceLanguage = '00'`; 00 není název žádného jazyku, k překladu tedy dojde vždy a pokud bude jazyk nastaven na cs, pak bude hledat soubory s překladem v `/protected/messages/cs/`. V `/protected/messages/en/` je potřeba vytvořit soubor `model.php` obsahující:

```
return array(
    'Article .label' => 'Label',
    'Article .body' => 'Body',
    'Article .createtime' => 'Create..time',
)
```

Výpis 23: Překlad popisků do angličtiny

Stejný soubor pro české popisky v `/protected/messages/cs/`:

```
return array(
    'Article .label' => 'Nadpis',
    'Article .body' => 'Telo',
    'Article .createtime' => 'Datum..vytvoreni',
)
```

Výpis 24: Překlad popisků do češtiny

`Model.php` by měl obsahovat popisky pro všechny `modely`. Tento krok umožňuje automatické generování správných popisků, převážně kvůli využití widgetů. Při změně jazyka je nutné taky upravení formátování čísel a datumů. Ani tady Yii nezklamal a poskytuje jednoduché rozhraní s přednastaveným formátováním

```
Yii::app()->dateFormatter->formatDateTime(time(), 'short') [15]
```

5.9.3 Výběr jazyka

Výběr jazyka je vhodné uskutečnit přes widget, který vygeneruje odkaz a pomocí metody POST nebo GET předá zvolený jazyk controlleru. Aby jazyk zůstal stále nastaven, existuje několik řešení. Jednoduché řešení je vytvoření cookies, která bude obsahovat zvolený jazyk. Pokud robot vyhledávače navštíví stránku, jelikož neumí pracovat s cookies, nebude schopný přejít na jinou jazykovou mutaci a obsah nezaindexuje. Druhé a mnohem lepší řešení počívá v zahrnutí jazykové verze do URL adresy.

5.10 Správa adres

Správa URL adres aplikace pomocí `Urlmanager`. Zavádí se přidáním `'urlManager'=>array()` do `components` v `application configuration`, kde je také možné specifikovat pravidla pro parsování URL. Pro specifické nastavení URL se používají speciální formátovací výrazy v poli `rules`. Základní zobrazení článku, pomocí adresy `domena.cz/article/view/id/5`, není většinou žádoucí. Častá úprava spočívá ve změně vyhledávání článku v databázi. Místo hledání podle `id` se do systému zavede hledání podle jednoznačné URL přiřazené článku, nejčastěji totožné s nadpisem. K článku poté směřuje adresa `domena.cz/article/view/url/clanek-o-yii`, která však obsahuje několik pro uživatele, zbytečných údajů. Zobrazení článku pomocí pěkné adresy `domena.cz/article/clanek-o-yii/` se provede pomocí regulárního výrazu `'article/<url:w+>'`, který odkazuje na `controller article`, `action view` zároveň s `GET` parametrem `url`, obsahující řetězec `clanek-o-yii`.

```
'urlManager'=>array(
    'urlFormat'=>'path',
    'rules'=>array(
        'article/<url:w+>' => 'article/view',
    ),
),
```

Výpis 25: Pravidla pro parsování url

Pokud má být v URL zahrnut také aktuální jazyk, nutné rozšíření `CurlManageru` se provede přidáním `'class'=>'ext.components.CmsUrlRule'` a implementací třídy `CmsUrlRule`.

```
'urlManager'=>array(
    'class'=>'cms.components.CmsUrlRule',
    'urlFormat'=>'path',
    'rules'=>array(
        '<language:[a-z]{2}>/article/<url:w+>' => '<language>/article/view',
    ),
),
```

Výpis 26: Vlastní pravidla pro parsování url

Třída `CmsUrlRule` implementuje metody `createUrl()` a `parseUrl()`. Při vytváření adresy se přidává aktuální jazyk na začátek adresy, při parsování se tento řetězec odstraňuje a nastavuje aktuální jazyk, čímž je uchováno aktuální jazykové nastavení.

```
public function createUrl($route,$params=array(),$ampersand='&')
{
    $route = Yii::app()->getLanguage().'.'.$route;
    return parent::createUrl($route, $params, $ampersand);
}
public function parseUrl($request)
{
    $a = parent::parseUrl($request);
    Yii::app()->setLanguage(substr($a,2));
    return substr($a,3);
}
```

Výpis 27: Url parser využívající aktuální jazyk

Anglická verze stránek je přístupná přes `domena.cz/en/controller/action`, zatímco česká přes `domena.cz/cs/controller/action`.

5.11 Témata

Témata (skiny) se využívají pro rychlou a jednoduchou změnu vzhledu aplikace. Témata jsou ve samostatných adresářích v `/themes/`. Každé téma obsahuje ve složce `views` adresář `layouts`, jež je totožný s `/protected/views/layouts` a taktéž obsahuje soubory `main.php`, `column1.php`, `column2.php`, které tvoří základní layout aplikace. Další potřebné soubory, mezi které patří CSS nastavení, obrázky a JavaScript soubory jsou uloženy ve složkách adresáře `téma`. Pro přístup k CSS souboru

`/themes/mytheme/css/style.css` z `/mytheme/views/layout/main.php` využijeme `Yii::app()->theme->baseUrl.'/css/style.css'`. Vytvořené téma se aktivuje přidáním `'theme'=>'mytheme'` do `application configuration`. Pokud je v modulu vyžadováno jiné téma, než které je nastaveno v `application configuration`, nastavení se provede jednoduše pomocí `Yii::app()->theme = 'mytheme2'`. Práce s tématy je velice jednoduchá a rychlá. Díky jejich struktuře jsou téměř nezávislá na aplikaci a je možné je snadno sdílet mezi vývojáři.

5.12 Cash

Dramatickému zvýšení výkonu aplikace nejvíce napomáhá vhodně zvolené cashování. Yii poskytuje několik nástrojů, které umožňují efektivní a přesto jednoduché cashování

na několika úrovních. Dynamicky získaná data se uchovávají maximálně možnou dobu v cashi tak, aby obsah cashe byl pokud možno vždy aktuální a odpovídal dynamickým datům. Aplikace využívá cash namísto zdlouhavého generování dynamických dat. Data se cashují na těchto úrovních:

5.12.1 Datová

Cashuje obsah proměnných. Při každém cashování je možnost nastavit závislost (dependency) na nějakém objektu, při jehož změně se provede nové cashování. Tato závislost může být na souboru, databázi, GET požadavku nebo volané funkci.

\$dependency = new CDbCacheDependency('SELECT MAX(update_time) FROM article'). Pokud kód zpracovává velké množství dat, která jsou následně uložena v proměnné, je vhodné tyto data cashovat a generovat nové pouze, když jsou změněna. Nastavit maximální dobu uložení v cashi je nutné zejména, pokud není nastavena závislost.

```
$value=Yii::app()->cache->get($id);  
if ($value===false)  
{  
    Yii::app()->cache->set($id,$value, 600, $dependency);  
}
```

Výpis 28: Cashování proměnných

Velice výhodné je taktéž uchování výsledku databázového dotazu.

```
$posts = Post::model()->cache(600, $dependency)->findAll();
```

5.12.2 Fragmentová

Cashuje pouze část vygenerované stránky. Vhodné zejména pro cashování widgetů.

```
$this->beginCache($id,$dependency); $this->endCache();
```

5.12.3 Stránková

Cashuje celou vygenerovanou stránku. Jedná se o speciální případ fragmentového cashování. Nastavuje se jako filtr COutputCache a pokud je stránka v cashi nedojde k vykonání action daného controlleru.

```
array(  
    'COutputCache',  
    'duration'=>100,  
),
```

Výpis 29: Cashování rendrované stránky

5.13 Logování

Slouží pro zpracovávání informací o běhu aplikace. Zprávy dělíme na log a trace. Log jsou brány v potaz pouze, pokud je aplikace ve vývojovém módu (*YII_DEBUG = true*). Pokud je aplikace nasazena do provozu, uplatňují se trace zprávy.

```
Yii::log($message, $level, $category); Yii::trace($message, $category);
```

5.13.1 Level

- trace: stejný level využívá trace()
- info: tento log poskytuje základní informace.
- Profile: pro měření výkonu.
- warning: varující zprávy.
- error: slouží pro závažné chyby.

5.13.2 Kategorie

Kategorie specifikuje třídu, která zprávu generuje. Pro log z controlleru nastavíme kategorii na `system.web.CController`

5.13.3 Zpracování

Zprávy je možné odchytnout pomocí třídy `CLogRouter`, inicializované jako komponenta v application configuration. Specifikací routes v konfiguraci lze určit jakým způsobem budou které zprávy zpracovány. Zprávy je možné uložit do souboru, odeslat emailem, zobrazit na konci renderované stránky, nebo uložit do databáze.

```
'components'=>array(
  'log'=>array(
    'class'=>'CLogRouter',
    'routes'=>array(
      array(
        'class'=>'CFileLogRoute',
        'levels'=>'trace,info',
        'categories'=>'system.*',
      ),
      array(
        'class'=>'CEmailLogRoute',
        'levels'=>'error,warning',
        'emails'=>'admin@example.com',
      ),
    ),
  ),
),
```

Výpis 30: Zpracování logu

6 Implementace CMS

V současné době se celý CMS nachází na <http://yiiguru.chapcak.cz>. CMS využívá většinu komponent frameworku a mnoho z těchto komponent je rozšířeno, shodně s popisem v části 5. Systém je navržen modulově, s maximálním ohledem na rozšiřitelnost a efektivnost. Mezi jeho další velké benefity patří možnost určit všem uživatelům oprávnění přístupu k modulům a multijazyčnost aplikace. Mezi implementované moduly patří.

6.1 Module

Modul, který umožňuje správu ostatních modulů. Jádro celého systému, bez kterého není možné provádět jakoukoliv operaci. Nový modul je možné aktivovat pouze, pokud je jeho controller uložen v `/protected/controllers/` a modul není aktivní. Tato funkcionality využívá funkci procházející obsah složky na disku, ve spojení s databázovým SQL dotazem.

Po přidání nového modulu mají k němu všichni uživatelé přístup. Oprávnění pro přístup lze změnit v nastavení uživatele. Instalované moduly se automaticky přidávají do horního menu administrace a jsou tak jednoduše přístupné uživatelům.

6.2 User

Správa uživatelů a jejich oprávnění pro jednotlivé moduly. Tento prvek, stejně jako předchozí, je v systému klíčový, neboť umožňuje přihlášení a práci uživatelům. Poskytuje efektivní náhled detailů pomocí AJAX okna. Editace uživatelského oprávnění je také dynamická. Oprávnění lze nastavovat pro všechny instalované moduly a je rozděleno do několika úrovní:

- zakázán
- číst pouze vlastní

- číst vše
- měnit pouze vlastní
- měnit vlastní, číst vše
- měnit vše

6.3 Article

Vytváření a správa článků. Články je možné vytvářet v libovolných jazykových mutacích. Pohodlnou editaci obsahu poskytuje WYSIWYG editor TinyMCE. Do článku je možné vkládat obrázky z modulu Photo pomocí TinyMCE API. Každý článek může obsahovat Meta tagy (title, keywords, description) důležité pro SEO. Modul také obsahuje podrobný výpis článků s možností vyhledávání pomocí několika parametrů.

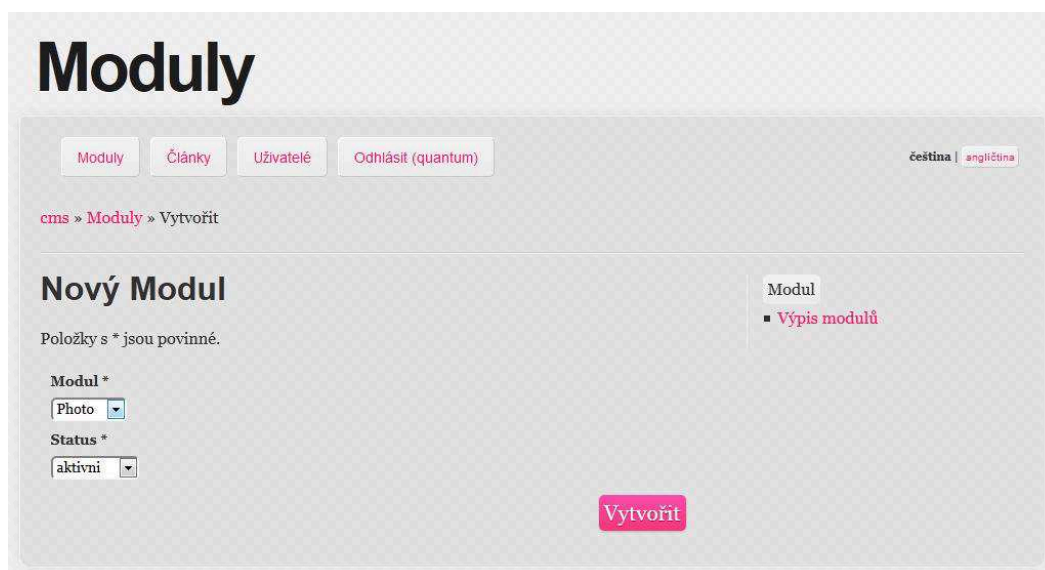
6.4 Photo

Modul určený pro nahrávání fotografií. Pro nahrání fotografie na server implementuje komponentu třetí strany pomocí widgetu. Nahrané fotografie zpracuje a uloží v originální a zmenšené velikosti na základě vlastní komponenty. Poskytuje jednoduché AJAX mazání fotografií. Ostatní moduly mohou využívat AJAX zobrazení náhledů fotografií a pomocí JavaScriptu odchytil událost při kliknutí na obrázek.

6.5 Případy užití

6.5.1 Aktivace modulu

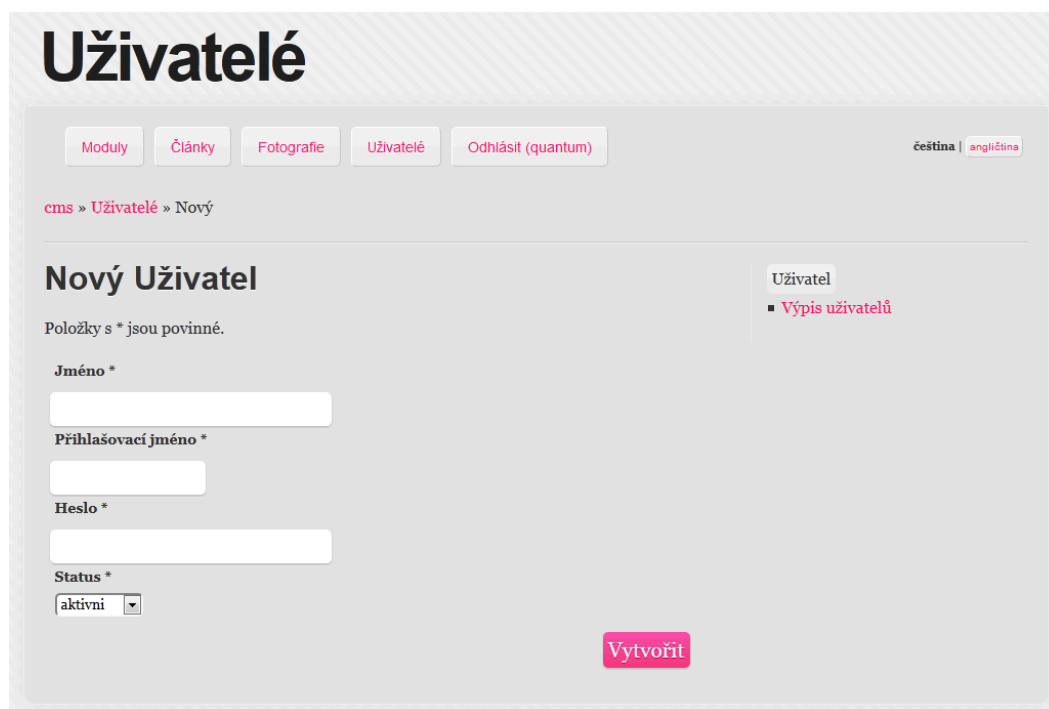
Pro vložení modulu je nutné mít modul nahraný na serveru (standartně jsou na serveru nahrané všechny implementované moduly). Na záložce moduly stačí zvolit "Nový modul", poté vybrat modul ze seznamu nenainstalovaných a potvrdit tlačítkem vytvořit.



Obrázek 1: Přidání modulu

6.5.2 Přidání uživatele

Postup je obdobný, jako u Modulu. V sekci uživatelé je v bočním menu volba "Nový uživatel". Jako jméno je vhodné zvolit skutečné jméno osoby účet využívající. Do kolonky "Přihlašovací jméno" se vkládá libovolný login, pomocí kterého se bude uživatel přihlašovat do systému. Nutné je také nezapomenout na heslo. Pokud bude uživatel označen jako neaktivní, nebude se moct do systému přihlásit.



The screenshot shows a web interface for creating a new user. At the top, there is a navigation bar with buttons for 'Moduly', 'Články', 'Fotografie', 'Uživatelé', and 'Odhlásit (quantum)'. The current page is 'Uživatelé' and the sub-page is 'Nový'. The main heading is 'Nový Uživatel'. Below the heading, there is a note: 'Položky s * jsou povinné.' The form contains the following fields: 'Jméno *' (text input), 'Přihlašovací jméno *' (text input), 'Heslo *' (text input), and 'Status *' (dropdown menu with 'aktivní' selected). A 'Vytvořit' button is located at the bottom right of the form. On the right side of the form, there is a sidebar with the heading 'Uživatel' and a sub-heading 'Výpis uživatelů'.

Obrázek 2: Vytvoření uživatele

6.5.3 Vytvoření nového článku

V modulu články je, po zvolení "Nový článek", rozsáhlý formulář, jehož povinné položky jsou "nadpis" a "url". Jako první je však nutné zvolit jazyk článku. Okno pro vložení obrázku pomocí URL se aktivuje stisknutím ikonky s vyobrazeným stromem v nástrojové liště editoru. Pokud je požadavek vložit obrázek z modulu Photo (fotografie), stačí stisknout tlačítko "Vložit obrázek" pod editorem a v zobrazeném okně vybrat miniaturu požadovaného obrázku. Ačkoliv je mnoho položek nepovinných, velice se doporučuje náležitě vyplnit všechny, neboť mají zásadní vliv na SEO.

Vytvořit Článek

Nový článek

Položky s * jsou povinné.

Jazyk *

Nadpis *

Obsah

Rich text editor toolbar with options: Bold, Italic, Underline, ABC, Bulleted list, Numbered list, Styles, Paragraph, Undo, Redo, Link, Unlink, Image, Table, Source, Help, and a text area below.

Text mode

Vložit obrázek

Úřl *

Článek

- [Výpis článků](#)
- [Podrobný výpis článků](#)

Obrázek 3: Vytvoření článku

Články

[Moduly](#)
[Články](#)
[Fotografie](#)
[Uživatelé](#)
[Odhlášt \(Martin Chapčák\)](#)
čeština | [angličtina](#)

[cms](#) » [Články](#) » [Podrobný výpis článků](#)

Podrobný výpis článků

Můžete vkládat porovnávací znaky (<, <=, >, >=, <> or =) na začátek každého vyhledávacího pole pro specifikaci vyhledávání.

[Pokročilé vyhledávání](#)

Jazyk	Nadpis	Vytvořeno	Upraveno	Úřl	Status	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
en	CMS	27. duben 2012	28. duben 2012	cms_english	aktivni	
cs	CMS	27. duben 2012	28. duben 2012	cms_cesky	aktivni	
en	Framework	27. duben 2012	28. duben 2012	framework_english	aktivni	
cs	Framework	27. duben 2012	28. duben 2012	framework_cesky	aktivni	

Obrázek 4: Filtr článků

7 Závěr

Tématem bakalářské práce je seznámení se s PHP frameworkem Yii a demonstrace jeho benefitů na příkladech. V teoretické části o vývoji webových aplikací byly popsány důvody pro použití frameworku Yii. Důležitý popis jednotlivých komponent, spolu s ukázkami implementace tvoří největší část této práce a má největší vypovídací hodnotu o práci s frameworkem. Ačkoliv je celé téma velice rozsáhlé, práce ho s úspěšností stručně shrnuje a poskytuje tak cenný zdroj informací, který zásadně napomáhá k rychlejšímu pochopení frameworku.

CMS systém, který byl v rámci práce naprogramován, vyniká modulovou architekturou a jednoduchostí. Systém má poměrně velký potenciál nasazení na menších projektech a předpokládá se jeho postupné rozšiřování. Mezi moduly, které by měly být implementovány patří modul pro správu kategorií, pomocí kterého by se velké množství článků uspořádalo do přehledné stromové struktury. Současný systém s tímto rozšířením počítá a pro toto rozšíření má již připravenou architekturu v databázi.

Tvorba bakalářské práce byla velice různorodá a náročná. Právě proto zásadně přispěla k mé lepší orientaci v internetových technologiích. Důkladné seznámení s frameworkem Yii pokládám za milník, který určitě ovlivní směr mé profesní orientace.

Martin Chapčák

Seznam obrázků

1	Přidání modulu	37
2	Vytvoření uživatele	38
3	Vytvoření článku	39
4	Filtr článků	39

Seznam výpisů zdrojového kódu

1	Připojení k SQLite	14
2	Připojení k MySQL	14
3	Zavedení Gii	15
4	Předání parametru modulu	16
5	Přidání vlastní komponenty	17
6	Vlastní widget	18
7	Widget ve view	19
8	Přidání behavior v main.php	20
9	Specifikace behavior	20
10	Generování AJAX odkazu pro smazání záznamu	21
11	AJAX tlačítko pro obnovení výpisu	21
12	Action podporující AJAX	22
13	Dropdownlist využívající AJAX	22
14	Přidání filtru do controlleru	23
15	Autentizace uživatele	23
16	Ověření hesla pro přihlášení	24
17	Oprávnění pro controller	24
18	Oprávnění které jsou uloženy v databázi	25
19	Uprava URL parseru	26
20	Nastavení jazyka	27
21	Překlad menu	28
22	Automatický překlad popisků	28
23	Překlad popisků do angličtiny	29

24	Překlad popisků do češtiny	29
25	Pravidla pro parsování url	30
26	Vlastní pravidla pro parsování url	30
27	Url parser využívající aktuální jazyk	31
28	Cashování proměnných	32
29	Cashování rendrované stránky	33
30	Zpracování logu	34

8 Reference

- [1] *Content management system* [online]. c2012 ,[cit. 2012-04-24]
<http://en.wikipedia.org/wiki/Content_management_system>.
- [2] *Yii* [online]. c2012 , [cit. 2012-04-24] <<http://en.wikipedia.org/wiki/Yii>>
- [3] *Yii mieša karty PHP frameworkov* [online]. c2012 , [cit. 2012-04-24]
<<http://www.zdrojak.cz/clanky/yii-miesa-karty-php-frameworkov/>>
- [4] *PHP* [online]. c2012 , [cit. 2012-04-24]
<<http://cs.wikipedia.org/wiki/PHP>>
- [5] *Class reference* [online]. c2012 , [cit. 2012-04-24]
<<http://www.yiiframework.com/doc/api/>>
- [6] *The Definitive Guide to Yii* [online]. c2012 , [cit. 2012-04-24]
<<http://www.yiiframework.com/doc/guide/>>
- [7] *Building a Blog System using Yii* [online]. c2012 , [cit. 2012-04-24]
<<http://www.yiiframework.com/doc/blog/>>
- [8] *Yii framework forum* [online]. c2012 , [cit. 2012-04-24]
<<http://www.yiiframework.com/forum/>>
- [9] *Model–view–controller* [online]. c2012 , [cit. 2012-04-24]
<<http://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>>
- [10] *PHP Manual* [online]. c2012 , [cit. 2012-04-24]
<<http://www.php.net/manual/en/index.php>>
- [11] *Web application framework* [online]. c2012 , [cit. 2012-04-24]
<http://en.wikipedia.org/wiki/Web_application_framework>
- [12] *SQLite* [online]. c2012 , [cit. 2012-04-24] <<http://www.sqlite.org/>>

[13] *The SQL-92 standard* [online]. c2012 , [cit. 2012-04-24]

<<http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>>

[14] *Ajax in jQuery* [online]. c2012 , [cit. 2012-04-24]

<<http://api.jquery.com/category/ajax/>>

[15] *Internationalization* [online]. c2012 , [cit. 2012-04-24]

<<http://www.yiiframework.com/doc/guide/1.1/en/topics.i18n>>

A Příloha na CD/DVD

CMS systém, i s frameworkem Yii. Systém nevyžaduje dodatečné nastavení, lze spustit pouze na webovém serveru.