

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

## MODULÁRNÍ CMS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ HORKÝ

BRNO 2015



**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA INFORMAČNÍCH TECHNOLOGIÍ**  
**ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ**

**FACULTY OF INFORMATION TECHNOLOGY**  
**DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA**

## **MODULÁRNÍ CMS**

MODULAR CMS

### **BAKALÁŘSKÁ PRÁCE**

BACHELOR'S THESIS

### **AUTOR PRÁCE**

AUTHOR

**LUKÁŠ HORKÝ**

### **VEDOUCÍ PRÁCE**

SUPERVISOR

**Ing. IGOR SZÖKE, Ph.D.**

BRNO 2015

## **Abstrakt**

Úkolem této bakalářské práce je vytvoření webového systému pro generování a správu webových portálů. Systém umožňuje snadnou instalaci, nastavení a správu obsahu webu a operuje nad modulárním prostředím, které umožňuje rozšiřitelnost prostřednictvím modulů. Součástí systému je vestavěný editor pro tvorbu šablon stránek.

## **Abstract**

The goal of this bachelor's thesis is to develop a content management system for generating and managing web portals. System offers an easy installation, settings and administration of content and operates over a modular environment, which provides expandability via modules. System includes a built-in editor for creating page templates.

## **Klíčová slova**

CMS, web, systém, webový redakční systém, HTML, PHP, MySQL

## **Keywords**

CMS, web, system, content management system, HTML, PHP, MySQL

## **Citace**

Lukáš Horký: Modulární CMS, bakalářská práce, Brno, FIT VUT v Brně, 2015

# Modulární CMS

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szökeho, PhD. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Lukáš Horký  
20. května 2015

## Poděkování

Děkuji svému vedoucímu Ing. Igoru Szökemu, PhD. za jeho vedení a odborné podněty, které mi při tvorbě této bakalářské práce pomohly.

© Lukáš Horký, 2015.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>3</b>
<b>2</b>	<b>Současné systémy pro správu obsahu</b>	<b>4</b>
2.1	WordPress	5
2.2	Drupal	5
2.3	Joomla!	5
2.4	Proč vytvářet další systém	5
2.4.1	Šablony a tvorba webu na míru	5
2.4.2	Neměnná struktura obsahu webu	6
2.4.3	Kontrola nad chováním modulů	6
<b>3</b>	<b>Použité technologie</b>	<b>8</b>
3.1	Jazyk HTML	8
3.2	Jazyk CSS	8
3.3	Jazyk PHP	9
3.4	Databáze MySQL	9
<b>4</b>	<b>Návrh a implementace</b>	<b>10</b>
4.1	Strukturování informací	10
4.2	Databázová struktura	11
4.2.1	SQL injection	12
4.3	Instalátor	13
4.4	Jádro systému	13
4.4.1	Předzpracování požadavků	13
4.4.2	Vykreslování uzlů	14
4.5	Návrh administrace	15
4.5.1	Dashboard	15
4.5.2	Command list	16
4.6	Uživatelé, skupiny a pravomoce	16
4.7	Moduly a jejich správa	17
4.7.1	Chování ovládacích prvků	18
4.8	Nástroje pro vývojáře	19
4.8.1	Vykreslování dynamických tabulek	20
4.8.2	Práce s formuláři	21
4.9	Editor šablon a editor témat	23
4.9.1	Editor šablon	23
4.9.2	Editor témat	24
4.10	Dodatečné nástroje	25

4.10.1 Správce souborů . . . . .	25
4.10.2 Správce logů . . . . .	25
4.10.3 Správce systémových konstant . . . . .	27
<b>5 Testování a první nasazení</b>	<b>28</b>
5.1 Podněty uživatelů . . . . .	28
<b>6 Závěr</b>	<b>30</b>
<b>A Metriky kódu</b>	<b>32</b>
<b>B Obsah CD</b>	<b>33</b>
<b>C ER diagram systému</b>	<b>34</b>
<b>D Plakát</b>	<b>36</b>

# Kapitola 1

## Úvod

Úkolem této práce bylo vytvoření modulárního systému pro správu obsahu ve webovém prostředí, který umožňuje snadné vytvoření a údržbu webových portálů. Kromě klasických nástrojů pro nastavení a správu obsahu a uživatelů pak systém řeší problematiku strukturalizace a prezentace informací prostřednictvím vestavěného editoru šablon a editoru témat.

Následující text poskytuje čtenáři základní informace o procesu návrhu a implementace systému Constructo CMS a o jeho stavu v aktuální verzi ke květnu roku 2015.

Kapitola 2 pojednává o současných nejrozšířenějších systémech pro správu obsahu ve webovém prostředí, jejich silných a slabých stránkách a jejich nedostacích. Podkapitola 2.4 se pokouší tyto nedostatky shrnout a opodstatnit tvorbu zcela nového systému na již zdánlivě zaplněném trhu.

Kapitola 3 obsahuje seznam a popis technologií, které byly použity při tvorbě systému Constructo CMS.

V kapitole 4 jsou popsány návrh a implementace systému Constructo CMS, jeho databázová struktura a samotné jádro systému. Dále zde najdete návrh modulů a jejich ovládacích prvků, včetně příkladů jejich užití. Následují editory pro správu struktury webového portálu a jeho uzlů a prezentování obsahu uzlů uživateli. V sekci 4.9.1 je popsán editor šablon pro administraci hierarchického rozmístění všech prvků na webu a sekce 4.10 pak obsahuje popis editoru témat.

Kapitola 5 pojednává o prvním nasazení a testování systému Constructo CMS. V samotném závěru se pak nachází popis aktuálního stavu systému, souhrn dosažených výsledků a plán vývoje projektu do budoucna.

## Kapitola 2

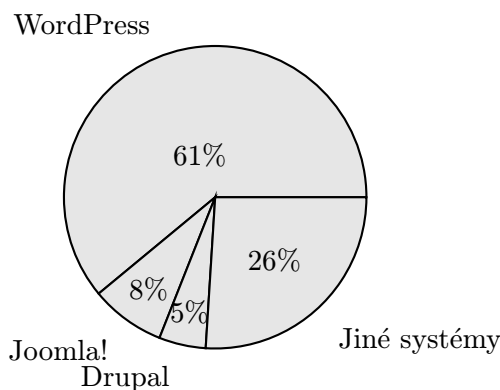
# Současné systémy pro správu obsahu

Systémem pro správu obsahu (někdy též nazývaným redakční či publikační systém) se rozumí software zajišťující správu dokumentů. V současnosti se tento pojem vztahuje převážně na webový obsah a tyto systémy pak lze chápat jakožto webové aplikace, jejichž funkcionality zahrnuje převážně:

- Tvorbu, editaci a mazání dokumentů (článků, stránek) skrze webové rozhraní
- Správu uživatelů a řízení přístupu k dokumentům
- Správu diskuzí či komentářů (fóra, komentování obsahu)
- Správu souborů, obrázků (galérie fotografií) či jiných multimédií

Tyto systémy pak obvykle poskytují správcům webového portálu zpětnou vazbu ve formě statistik přístupů a další nástroje pro sledování dění na webu. Dle pravidelné analýzy na portálu W3Techs<sup>1</sup> jsou třemi nejpoužívanějšími redakčními systémy v rámci 1 milionu nejnavštěvovanějších webových portálů systémy WordPress, Joomla! a Drupal.

<sup>1</sup>Analýza je dostupná na adrese: [http://w3techs.com/technologies/overview/content\\_management/all](http://w3techs.com/technologies/overview/content_management/all)



Obrázek 2.1: Podíly CMS nejnávštěvovanějších webů na trhu dle W3Techs



## 2.1 WordPress

WordPress je v dnešní době nejrozšířenější systém pro správu obsahu. Jeho původní myšlenkou bylo převážně poskytnutí nástroje pro tvorbu jednoduchých blogů, nicméně postupem času se s příchodem nových verzí jeho funkce rozrostly do podoby komplexního CMS. Jeho hlavní vizitkou je snadná instalace a nastavení a uživatelská přívětivost. Je vhodný pro blogy, korporátní webové stránky a obecně pro weby malých a středních rozměrů. Není vhodný pro příliš velké projekty či pro tvorbu stránek se striktními požadavky přesně na míru.

## 2.2 Drupal

Drupal je systém pro správu obsahu, jehož nejsilnější stránkou je rozmanitost a technická pokročilost. Nespotřebává tolik systémových zdrojů [7] a je možno jej použít pro tvorbu portálu jakýchkoliv rozměrů, od osobních blogů po rozsáhlé vícejazyčné portály obsahující desetitisíce stránek. Jeho základní funkce a nastavení jsou uživateli poskytnuty prostřednictvím jednoduchých rozhraní, nicméně pro využití plného potenciálu systému je třeba znalost jazyků HTML a PHP. Není doporučen pro úplné začátečníky. [9]

## 2.3 Joomla!

Joomla! se často považuje za kompromis mezi systémy WordPress a Drupal. Jedná se o systém pro správu obsahu, který si nevyžaduje takové technické znalosti, jako Drupal, ale přesto nabízí spoustu pokročilejších funkcí. Nejsilnější stránkou systému Joomla! je pravděpodobně možnost snadné tvorby vlastních sociálních sítí, případně komerčních webů, bez nutné znalosti programovacích jazyků.

## 2.4 Proč vytvářet další systém

Open-source systémů pro správu obsahu je v dnešní době celá řada. Na první pohled se může zdát, že již není možné vymyslet nic nového, co by již existující systémy neumožňovaly. V této kapitole bude proto rozepsáno několik oblastí, které nejsou v rozšířených systémech příliš pokryté. Těmto problematikám pak bylo věnováno největší úsilí při tvorbě této bakalářské práce.

### 2.4.1 Šablony a tvorba webu na míru

Téměř všechny moderní systémy pro správu obsahu umožňují snadný výběr z celé řady webových šablon (WordPress - téměř 3000<sup>2</sup>, Joomla! - více než 2000<sup>3</sup>, Drupal - více než 2000<sup>4</sup>).

Problém ovšem nastává v okamžiku, kdy máme přesný požadavek na vzhled a rozložení webových stránek pro prezentaci informací uživateli, který není možno splnit za pomoci žádné vestavěné či doinstalované šablony. V takovém případě má administrátor webového portálu několik možností:

<sup>2</sup>WordPress témata dostupná na: <https://wordpress.org/themes/>

<sup>3</sup>Joomla! témata dostupná na: <http://www.siteground.com/joomla-templates.htm>

<sup>4</sup>Drupal témata dostupná na: [https://www.drupal.org/project/project\\_theme](https://www.drupal.org/project/project_theme)

- Použít existující šablonu a tu se snažit upravit do co nejvíce přijatelné podoby
- Napsat vlastní šablonu, což si vyžaduje znalost PHP a jádra systému
- Objednat si šablonu za peníze

Některé méně známé systémy pro správu obsahu<sup>5</sup> se pokoušejí poskytnout nástroje pro editaci vzhledu konkrétních stránek či celých šablon v reálném čase. Programy, jako Microsoft FrontPage také umožňují měnit vzhled struktury stránek bez nutné znalosti webových jazyků.

Ve všech případech jsou ovšem tyto možnosti nedokonalé a čím větší abstraktní vrstva se nachází mezi editorem a výsledným zdrojovým kódem, tím více jsou možnosti těchto editorů omezené. Obecně pak ve výsledku platí, že pokud chce mít správce absolutní kontrolu nad strukturou a vzhledem webu, úpravě šablon na úrovni zdrojových kódů se nevyhne.

Tato problematika je v rámci bakalářské práce řešena prostřednictvím editoru šablon a editoru témat, jejichž návrh je blíže popsán v kapitole 4.9.

### 2.4.2 Neměnná struktura obsahu webu

Méně kvalitní systémy pro správu obsahu často neumožňují jakýkoliv zásah do struktury obsahu (či typů obsahu) na webovém portálu. Není neobvyklé, že není možno rozlišovat typy stránek jinak, než pomocí tagů<sup>6</sup> či že se obsah stránek vytváří pouze pomocí WYSIWYG editoru<sup>7</sup> bez jakékoliv strukturalizace prezentovaných informací.

Tuto problematiku některé pokročilejší systémy řeší různými způsoby. V systému Drupal je možno vytvářet různé typy uzlů (např. článek, stránka, epizoda seriálu...) a těm přiřazovat různá pole reprezentující konkrétní informace (např. číslo dílu seriálu, obrázek produktu, datum narození). Způsob reprezentace těchto dat je však následně velmi omezený a připomíná výpis formuláře pro tisk. Pokud si administrátor webového portálu přeje s těmito údaji v rámci výstupu naložit specifickým způsobem, je nucen sáhnout po modulech jako *Contemplate*<sup>8</sup>, které umožňují vytvářet „podšablony“ pro jednotlivé typy obsahu, nebo případně je nucen upravovat šablonu na úrovni PHP souborů na serveru. To si však vyžaduje pokročilou znalost jazyka PHP a samotného jádra systému.

V rámci bakalářské práce je tohle řešeno hierarchickým strukturováním informací jednotlivých uzlů webu, které je blíže popsáno v kapitole 4.1. Prezentace informací je následně umožněna formou editoru šablon a editoru témat, viz kapitola 4.9

### 2.4.3 Kontrola nad chováním modulů

Všechny rozšířené systémy pro správu obsahu operují nad modulárním prostředím<sup>9</sup> (WordPress - přes 35000 pluginů<sup>10</sup>, Joomla! - téměř 9000 rozšíření<sup>11</sup>, Drupal - téměř 30000 modulů<sup>12</sup>). Tyto pluginy či moduly často poskytují vlastní grafické rozhraní (příklad: Hodnocení článků prostřednictvím hvězd, kontaktní formulář, miniaturní galerie obrázků), nicméně

<sup>5</sup>Zde např. systémy Odo Website Builder 1.0, OpenWGA 6.2.6, Silex 1.6 Carbon či eFront 3.6.11

<sup>6</sup>Tagy, neboli značky, slouží pro označování obsahu webu (nejčastěji článků) několika slovy, které souvisí s daným tématem. Následně je možno filtrovat nebo vyhledávat na základě těchto značek.

<sup>7</sup>WYSIWYG editory jsou editory, které umožňují způsob editace dokumentů takovým způsobem, že verze zobrazená při úpravě je totožná s výslednou verzí dokumentu.

<sup>8</sup>Modul Contemplate je dostupný na adrese: <https://www.drupal.org/project/contemplate>

<sup>9</sup>Umožňují expanzi funkcionality webu prostřednictvím modulů nebo pluginů.

<sup>10</sup>WordPress pluginy dostupné na: <https://wordpress.org/plugins/>

<sup>11</sup>Joomla! rozšíření dostupná na: <http://extensions.joomla.org/>

<sup>12</sup>Drupal moduly dostupné na: <https://www.drupal.org/project/project>

umístění těchto ovládacích prvků často není v režii administrátora. Pokud aktivujeme plugin pro hodnocení, objeví se na konci všech článků; aktivace kontaktního formuláře vloží tento prvek např. do postranního menu. Občas je možno částečně ovlivnit umístění či podmínku pro zobrazení těchto pluginů v rámci jejich samotného nastavení, nicméně pokud chce mít správce webového portálu absolutní kontrolu, nezbyde mu, než ovládací prvek ručně vložit do šablony stránky nebo zdrojového kódu bloku v jazyce PHP.

V rámci bakalářské práce je řešení této problematiky podrobněji rozebráno v kapitole [4.7](#) a [4.9](#).

## Kapitola 3

# Použité technologie

V této kapitole jsou rozepsány technologie, které byly pro tvorbu bakalářské práce použity.

### 3.1 Jazyk HTML

*HyperText Markup Language*, dále jen HTML, je značkovací jazyk pro tvorbu hypertextových dokumentů<sup>1</sup> v systému *World Web Wide* pro publikaci dokumentů na internetu.

Aktuální verzí je k říjnu 2014 verze 5.0, která po patnáctileté odmlce navázala na verzi 4.01. Tato verze již ukončuje závislost na SGML [2] (univerzální značkovací jazyk) a kromě opravení některých chyb předešlé verze poskytuje i spoustu nových sémantických prvků[8], jako např.:

- audio a video pro multimediální obsah
- canvas pro renderování dynamické grafiky
- nové typy pro input, jako url, email, nummber, date...

### 3.2 Jazyk CSS

*Cascading Style Sheets*, dále jen CSS, je jazyk pro popis způsobu zobrazení elementů na stránkách napsaných v některém ze značkovacích jazyků, jako HTML či XML. Jeho hlavním smyslem je oddělení struktury a obsahu dokumentu od samotné definice vzhledu a tím zvýšení přehlednosti.[6]

Aktuální verzí je verze CSS3, která se pojí se standardem HTML5, jenž ji plně využívá. Oproti předchozí verzi CSS2.1 přináší např. tyto možnosti:

- animování elementů (dříve možno jen přes DHTML<sup>2</sup>)
- načítání písma z externího zdroje
- 2D a 3D transformace
- vrhání stínů, zakulacování okrajů, ořezávání...

---

<sup>1</sup>Hypertextové dokumenty jsou takové dokumenty, které nejsou lineární. Obsahují hypertextové odkazy a odkazují na jiné informace v systému.

<sup>2</sup>Dynamické HTML je kombinace technologií, které se používají při tvorbě dynamických a interaktivních webových stránek. Nejčastěji využívaný je JavaScript.

### 3.3 Jazyk PHP

PHP (rekurzivní zkratka pro *PHP: Hypertext Preprocessor* či starší *Personal Home Page*) je skriptovací, objektové orientovaný programovací jazyk určený převážně<sup>3</sup> pro programování dynamických internetových stránek či webových aplikací. Jedná se o interpretovaný jazyk<sup>4</sup>, jehož interpret běží na webovém serveru a na svém výstupu generuje na základě zdrojových kódů výstupní HTML stránky. Jazyk PHP je objektové orientovaný a je tak možné použít návrhové vzory.<sup>[3]</sup>

Jazyk PHP je nezávislý na platformě a operačním systému (až na několik systémově závislých funkcí) a programy psané v tomto jazyce jsou tak přenositelné bez jakýchkoliv úprav.

PHP podporuje rozšiřování prostřednictvím knihoven. V tomto ohledu je dobré zmínit knihovnu pro práci s databází MySQL, která je více popsána v kapitole 3.4.

Aktuální verze PHP ke květnu 2015 je 5.6.9.

### 3.4 Databáze MySQL

MySQL je volně šiřitelný<sup>5</sup> multiplatformní (je schopen běžet pod systémy Windows, Linux, Solaris, OS X a FreeBSD) databázový systém. Komunikace této databáze s externími aplikacemi probíhá prostřednictvím jazyka SQL s některými rozšířeními.

Velmi často je MySQL v rámci webových serverů používáno v kombinaci s operačním systémem Linux, HTTP serveru Apache a za přítomnosti interpretu jazyka PHP. Tato kombinace je pak označena zkratkou *LAMP* (**L**inux, **A**pache, **M**ySQL, **P**HP)<sup>[5]</sup>.

MySQL je schopno spolupracovat s PHP prostřednictvím PHP knihoven `php_mysql` nebo `php_mysqli`.

Aktuální verze MySQL k září 2014 je 5.6.21.

---

<sup>3</sup>Je možné jej využít i např. v příkazové řádce či pro programování desktopových aplikací.

<sup>4</sup>Je vyhodnocován za běhu, nekompiluje se do podoby spustitelného binárního souboru.

<sup>5</sup>MySQL je k dispozici ve dvou licencích - bezplatná GPL licence i komerční licence.

## Kapitola 4

# Návrh a implementace

V této kapitole je popsán návrh a následná implementace samotného systému pro správu obsahu Constructo CMS.

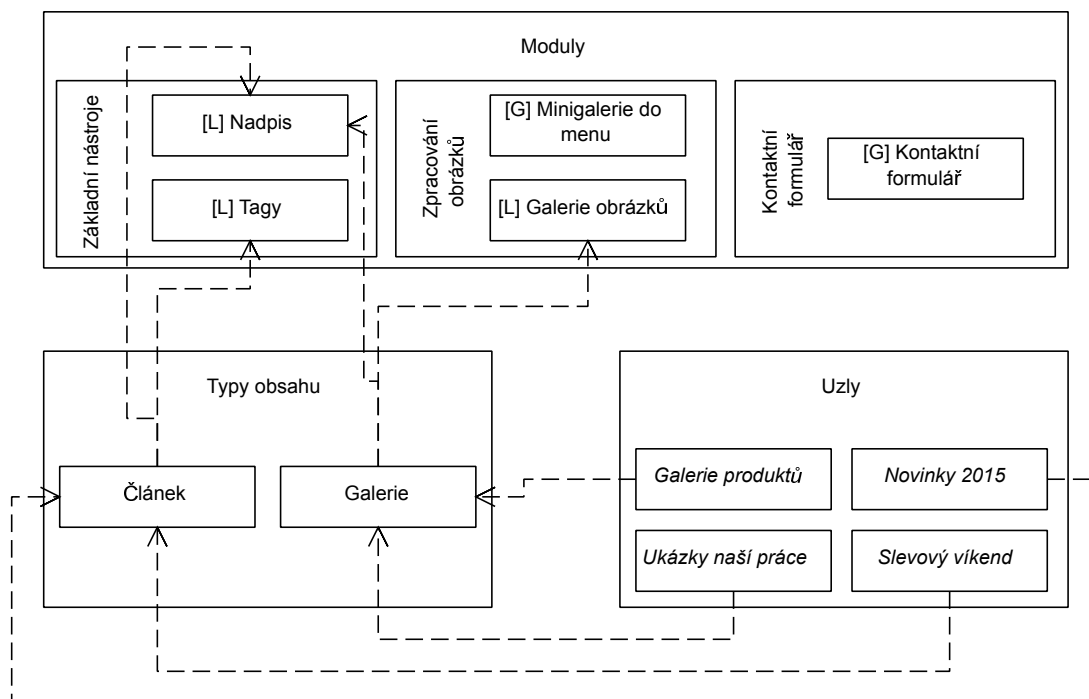
### 4.1 Strukturování informací

V kapitole 2.4.2 byl popsán problém s některými méně kvalitními systémy pro správu obsahu, které mají pevně danou strukturu informací, jenž jsou na výsledném webovém portálu prezentovány (např. základem je článek, ten obsahuje nadpis, tělo a tagy...). Ne vždy je však tento princip žádoucí. Proto byl pro Constructo CMS navržen obecný systém, který umožní správci webového portálu tvořit a strukturalizovat typy obsahů a jejich jednotlivé položky prostřednictvím ovládacích prvků na webu:

- Webová struktura sestává z libovolného počtu typů obsahu.
- Na webu je nainstalován předem neznámý počet modulů.
- Jeden modul může poskytovat libovolný počet lokálních a globálních ovládacích prvků.
  - Lokální ovládací prvek pracuje s daty konkrétního uzlu webového portálu. Příklad: Nadpis článku, obrázek produktu, jméno osoby...
  - Globální ovládací prvek pracuje se stejnými daty napříč webovým portálem. Příklad: Kontaktní formulář, vertikální menu, logo stránky...
- Typ obsahu sestává s libovolného počtu lokálních ovládacích prvků.
- Webový portál sestává z libovolného počtu uzlů.
- Uzel je právě jednoho typu obsahu. Sestává z URL adresy a poskytuje instance všech lokálních ovládacích prvků daného typu.

Na obrázku 4.1 je znázorněna jako příklad jedna z možných struktur webu. Je možno zde vidět provázání typů obsahu jednotlivých uzlů s lokálními ovládacími prvky nainstalovaných a aktivovaných modulů. Podrobněji jsou moduly a jejich správa popsány v kapitole 4.7.

Pro jednotlivé typy obsahu je možno vytvořit odlišné šablony. V případě obrázku 4.1 je tedy možné (a žádoucí), že galerie je ve výsledku vykreslena zcela odlišně, než obyčejný článek.



Obrázek 4.1: Příklad webové struktury. [L] značí lokální ovládací prvky modulu, [G] značí globální ovládací prvky.

## 4.2 Databázová struktura

Databázová struktura byla navržena s ohledem na požadavky na některé mechanismy týkající se strukturování informací do předem neznámé podoby (viz kapitola 4.1, případně i 4.9). V příloze C je možno vidět ER diagram aktuální databázové struktury systému. Stručný přehled jednotlivých entit:

- **content** - Data všech uzlů webového portálu.
  - Pakliže záznam nemá žádného rodiče, jedná se o záznam samotného uzlu a v rámci dat je zde uložena jeho URL adresa.
  - Pokud záznam má rodiče, jedná se o data lokálního ovládacího prvku pro daný typ uzlu.
- **content.types** - Typy obsahu webového portálu spolu s přiřazením lokálních ovládacích prvků k typům.
  - Nemá-li záznam rodiče, jedná se o typ obsahu uzlu.
  - Pokud záznam má rodiče, jde o lokální ovládací prvek strukturálně přidružený k danému typu. Jeden ovládací prvek může být přidělen k více typům obsahu.
- **dash.tiles** - Slouží pro ukládání stavu domovské stránky administrace systému (rozmístění dlaždic).
- **logs** - Záznamy systémových hlášení.
- **modules** - Všechny nainstalované moduly a jejich stav (aktivní / neaktivní).

- Příklad kódu na straně serveru:
  - `"SELECT * FROM users WHERE name = ' " . $input . "';"`
- Příklad uživatelského vstupu:
  - `$input = "1' OR '1' = '1"`
- Výsledný SQL kód:
  - `"SELECT * FROM users WHERE name = '1' OR '1' = '1';"`

Obrázek 4.2: Ukázka techniky SQL injection.

- `permissions` - Seznam uživatelských pravomocí.
- `scheme_structures` - Rekurzivně uložená data konkrétních šablon webu.
- `schemes` - Všechny šablony webového portálu.
- `settings` - Systémové konstanty.
- `theme_structures` - Rekurzivně uložená data konkrétních témat webu.
- `themes` - Všechna témata webového portálu.
- `user_groups` - Uživatelské skupiny.
- `users` - Všichni uživatelé webového portálu.

Uložení dat globálních ovládacích prvků je zcela v režii daných rodičovských modulů. Každý modul má možnost po instalaci vytvořit v databázi nové tabulky pro vlastní potřeby.

Data lokálních ovládacích prvků jsou však automaticky ukládána do `content` při úpravě obsahu uzlu webu. Ovládací prvek pak tato data obdrží automaticky při vykreslení stránky v rámci rekurzivního načtení všech dat daného uzlu.

### 4.2.1 SQL injection

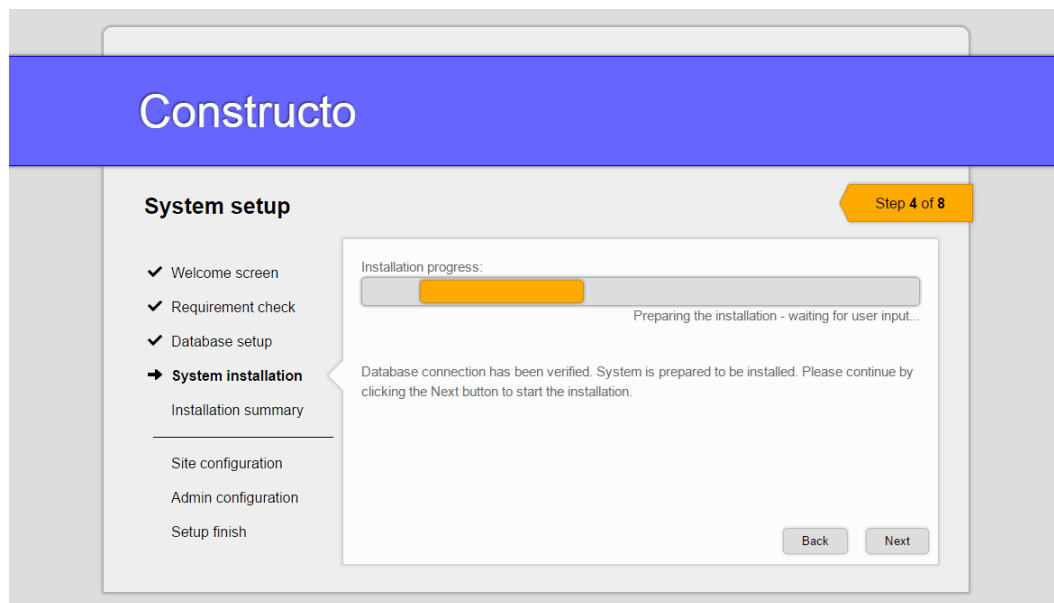
SQL injection je známá technika napadení databázové vrstvy webové aplikace vsunutím kódu skrze neošetřený vstup aplikační vrstvy. To vede k vykonání vlastního SQL dotazu útočníkem.

V ukázce 4.2 můžete vidět zneužití nechráněného vstupu metodou SQL injection. Dotaz nevrátí pouze záznam, ve kterém jméno uživatele odpovídá vstupu, ale všechny existující uživatele z dané tabulky v databázi.

V horším případě může útočník použít vlastní sérii SQL příkazů a získat úplnou kontrolu nad celou databází. Může všechna data nejen číst, ale i měnit či mazat. Podobných průniků a následných důsledků je samozřejmě celá řada.

Systém Constructo CMS poskytuje vlastní funkce pro práci s databází, mezi nimiž je i funkce `database_escape`, která umožní „escapovat“ všechny nebezpečné znaky z aplikačního vstupu a připravit tak řetězec pro bezpečnou práci na straně databázové vrstvy.





Obrázek 4.3: Ukázka procesu instalace systému.

## 4.3 Instalátor

Instalátor systému Constructo CMS se automaticky spustí v okamžiku, kdy uživatel vstoupí na libovolnou webovou stránku v rámci domény webového portálu, za předpokladu, že systém ještě není nainstalován.

Během instalace systém automaticky prověří, jestli jsou splněny všechny požadavky, jež systém pro správný běh vyžaduje. Pakliže je vše v pořádku, je třeba vyplnit údaje pro připojení k databázi (server, databáze, uživatel a heslo). Následuje samotný proces instalace, jež sestává převážně z tvorby databázové struktury a vyplnění některých tabulek počátečními údaji.

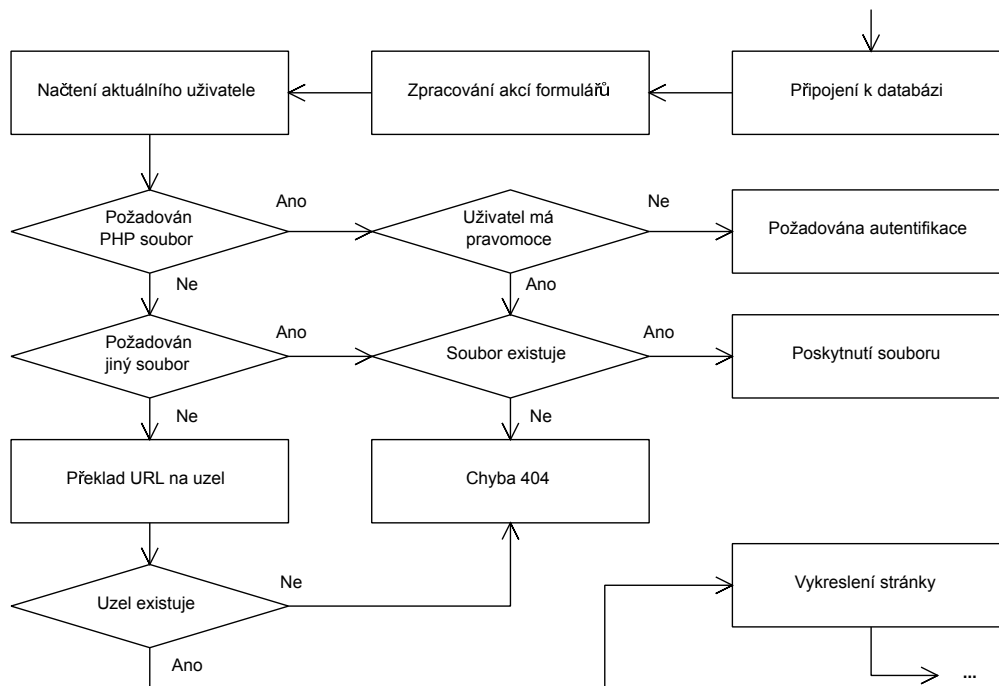
Po dokončení instalace je uživatel vyzván, aby vyplnil některé základní údaje, jako je název webu, jeho popis či prvotní uživatelský účet a superheslo systému. Poté je uživatel přeměrován do domovské stránky administrace a systém je připraven k použití.

## 4.4 Jádro systému

### 4.4.1 Předzpracování požadavků

Na obrázku 4.4 je znázorněno schéma jednotlivých kroků, které se vykonají, obdržel-li systém požadavek na nějaký dokument. Všechny požadavky jsou s výjimkou obrázků automaticky přeměrovány na hlavní systémový soubor `index.php`, který následně rozhoduje o vykonání potřebné akce.

Pokud se jedná o požadavek na existující soubor na serveru, systém tento soubor uživateli poskytne. V případě PHP souboru se ale nejprve provede ověření, že uživatel má k souboru přístup. To je z toho důvodu, aby nebylo možno přistoupit k systémovým souborům (např. soubory potřebné pro administraci) neautorizovanými uživateli a následně vynucení akce systému pouze na základě znalosti hierarchie systému. Pakliže uživatel nemá dostatečné pravomoce, přístup je mu zamítnut.



Obrázek 4.4: Schéma jednotlivých kroků při zaslání požadavku systému.

Není-li soubor nalezen, provede se překlad URL na uzel. Každý uzel má právě jednu URL adresu, nicméně přesměrování aliasů je možno realizovat pomocí modulu. Pakliže uzel existuje, provede se vykreslení stránky. Pokud ne, dojde k přesměrování uživatele na stránku s chybou 404.

#### 4.4.2 Vykreslování uzlů

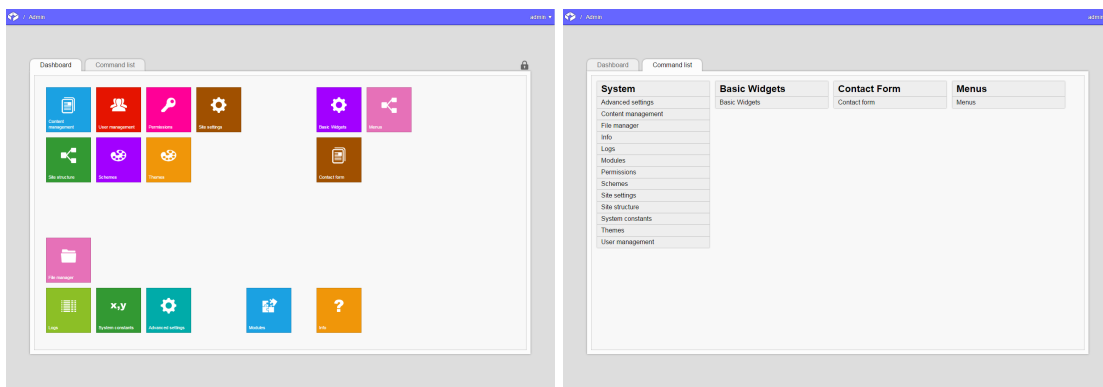
Při návrhu jádra systému bylo třeba dbát na následující:

- Všechny moduly musí mít informace o ostatních modulech ve stejném okamžiku, tedy nezávisle na pořadí inicializace jednotlivých ovládacích prvků.
- Všechny moduly musí být schopny spolu komunikovat v případě potřeby.
- Musí být možno umísťovat ovládací prvky jednotlivých modulů do předem neznámých oblastí v šabloně webu a v předem neznámém pořadí.

Aby bylo docíleno nezávislosti ovládacích prvků modulů na pořadí jejich inicializace, je samotná inicializace prováděna před generováním výsledné stránky a umísťování ovládacích prvků modulů na ní. Všechny ovládací prvky tak mají v době svého vykreslení všechny údaje nejen o samotném uzlu a aktuálním uživateli, ale i o aktivních modulech a jiných použitých ovládacích prvcích.

Před samotným vykreslením stránky systém nejprve zjistí typ obsahu aktuálního uzlu. Od toho se odvíjí použitá šablona, která může být odlišná pro jednotlivé typy obsahu. Po nalezení šablony systém následně hledá aktivní téma, které je při procesu vykreslení použito.

Samotné vykreslování stránky probíhá takovým způsobem, že systém nejprve vyhledá kořenový záznam uzlu v datech struktury aktivní šablony. Tento záznam nese jméno *ROOT*



Obrázek 4.5: Ukázka domovské stránky administračního rozhraní. *Dashboard* vlevo, *Command list* vpravo. Nové moduly automaticky vytváří nové dlaždice a položky seznamu příkazů.

a nemá žádného předka. Kořenový záznam odpovídá globálnímu bloku, který následuje hned za značkou *body* při vykreslování stránky.

Jakmile systém nalezl kořenový záznam, všechny ostatní zanořené bloky stránky jsou hledány rekurzivně. To znamená, že ke každému zaznamenanému bloku jsou vyhledáváni jeho potomci a pro ty se proces opakuje. Pokud se jedná o záznam bez potomka, může se jednat buďto o prázdný blok, o lokální ovládací prvek uzlu a nebo o globální ovládací prvek webového portálu. V případě ovládacích prvků je následně volána funkce pro vykreslení dat ovládacího prvku, o čemž více pojednává kapitola 4.7.

V případě globálního ovládacího prvku je načtení patřičných dat zcela v režii rodičovského modulu, jelikož se předpokládá, že se jedná o komplexní datové celky. Lokální ovládací prvky však obdrží patřičná data uzlu automaticky. Systém vyhledá v záznamech dat uzlu takový záznam, který má aktuální uzel jako svého rodiče a jeho typ obsahu odpovídá lokálnímu ovládacímu prvku typu uzlu s patřičným jménem. Ve výsledku se jedná o právě jeden datový řetězec, jehož interpretace a vykreslení jsou zcela řízeny ovládacím prvkem.

## 4.5 Návrh administrace

Hlavní administrační stránka systému Constructo CMS sestává ze dvou možných zobrazení, mezi nimiž je možno přepínat prostřednictvím záložek. První zobrazení je tzv. *Dashboard*, který obsahuje všechny systémové stránky pro administraci ve formě dlaždic. Druhé zobrazení je *Command List*, jenž shrnuje všechny stránky pod rodičovské moduly. Ukázky obou rozvržení můžete zhlédnout na obrázku 4.5. Způsob tvorby vlastní administrační stránky v rámci nového modulu je popsán v kapitole 4.7

### 4.5.1 Dashboard

Dashboard je styl zobrazení administrátorské domovské stránky systému Constructo CMS, které je inspirováno uživatelským rozhraním *Modern UI* (dříve *Metro*[1]) společnosti Microsoft, jenž se aktuálně používá v systémech Windows Phone, Windows 8 a bude možné jej použít i v systému Windows 10.

Jedná se o rozhraní, k jehož zásadám patří jednoduchost [4] a zaměření na obsah a typografii. Všechny systémové stránky jsou reprezentovány dlaždicemi s ikonou a popisem

ve zvolené barvě. Každý administrátor má svůj vlastní Dashboard a má tak možnost vlastního přizpůsobení domovské stránky pro administraci. Po odemknutí dlaždic kliknutím na zámek v pravém horním rohu administračního rozhraní je možno metodou *Drag and drop*<sup>1</sup> přesouvat umístění všech dlaždic na libovolnou pozici a případně vytvářet nové skupiny dlaždic.

#### 4.5.2 Command list

Command list, neboli *Seznam příkazů* je styl vyobrazení administrační nabídky, který všechny stránky administrace shrnuje do skupin dle modulů, jež tyto stránky poskytují.

Toto zobrazení není možno nijak přizpůsobovat a je společné pro všechny administrátory webového portálu.

### 4.6 Uživatelé, skupiny a pravomoce

Většina webových redakčních systémů umožňuje hlavnímu administrátorovi vytvářet sekundární účty pro správce a moderátory webového portálu a těm nastavovat příslušné pravomoce do administračních sekcí. Systém Constructo CMS není výjimkou. Administrátor má možnost nejen vytvářet libovolný počet uživatelů, ale tyto uživatele i sdružovat do uživatelských skupin. Následně je možné přiřadit příslušné pravomoce jak uživatelské skupině, tak i konkrétnímu uživateli<sup>2</sup>.

Tím se odstranila potřeba vytvářet uživatelské skupiny pouze pro jednotlivé uživatele v případě menších webových portálů, ale s ponecháním možnosti přiřazení pravomocí skupinám v případě portálů větších.

V případě, že je uživatel členem více uživatelských skupin, stačí, aby při přístupu k funkci systému byly pravomoce přiřazeny alespoň jedné této skupině nebo případně samotnému uživateli. Pravomoce se nejprve hledá na konkrétním uživateli a poté cyklicky ve všech skupinách, jichž je uživatel členem.

Pakliže se uživatel snaží přistoupit k sekci webu, ke které nemá přístup, můžou nastat dvě situace:

- **Jedná se o anonymního návštěvníka** - v takovém případě mu je nabídnut přihlašovací formulář. Po přihlášení jsou znovu ověřovány pravomoce uživatele a uživatel je buďto připuštěn ke stránce systému, nebo nastane druhá možnost.
- **Jedná se o přihlášeného návštěvníka** - pakliže je uživatel již přihlášen, ale přesto nemá dostatečné pravomoce pro přístup na danou stránku, je uživateli nabídnut formulář pro použití *superhesla*.

*Superheslo* je globální heslo pro přístup k veškeré funkcionalitě systému. Důvodem jeho zavedení je zamezení problematických situací, kdy administrátor omylem smaže svůj vlastní účet, odebere pravomoce svému účtu či uživatelské skupině nebo např. vyprázdní celou administrátorskou skupinu. V takovém okamžiku administrátor ztrácí přístup k veškeré administraci, a tudíž není schopen ani sám sobě pravomoce opět přidělit.

Tato situace je v jiných systémech pro správu obsahu řešena nejrůznějšími oklikami. Systém Drupal například automaticky uživateli s ID 1 přiděluje nejvyšší možné pravomoce.

<sup>1</sup>Metoda Drag and drop spočívá v kliknutí myši na element, přetažení do cílové oblasti a uvolnění myši.

<sup>2</sup>Tím se systém Constructo CMS liší např. od systému Drupal, kde je možno přiřazovat pravomoce pouze uživatelským skupinám, nikoliv konkrétním uživatelům.

Pokud je ale tento účet smazán, nezbyvá nic jiného, než oprava na úrovni samotné databáze a ruční vytváření záznamů.

Z toho důvodu systém Constructo CMS poskytuje systémové superheslo, které je možno použít i když v systému neexistuje žádný uživatel, uživatelská skupina či pokud jsou všem jmenovaným odebrány veškeré pravomoce. I anonymní uživatel může použít superheslo a tím získat přístup k veškeré administraci.

Z toho samozřejmě plyne i bezpečnostní riziko. Superheslo by vždy mělo být velmi silné (Constructo CMS neomezuje maximální počet znaků v hesle).

## 4.7 Moduly a jejich správa

Většina modulárních systému pro správu obsahu ve webovém prostředí poskytuje pluginy ve formě stáhnutelných souborů (nejčastěji archivy), které se umístí na server, v případě nutnosti rozbali a aktivují v administraci. Ne vždy je však možno spravovat přesné umístění instancí takovýchto modulů na stránce. To v případě rozšíření bez grafického rozhraní (např. logovací moduly) nevádí, ale často v případě pluginů pro interakci s uživatelem správce webového portálu vyžaduje přesné umístění takového pluginu. (Pozn.: více o této problematice pojednává kapitola [2.4.3](#))

Tato problematika je v bakalářské práci řešena tím způsobem, že všechny moduly, které poskytují grafické rozhraní, jenž je možno umístit na stránku (např. kontaktní formulář, hodnocení článku...), tento úkon provádějí prostřednictvím lokálních a globálních ovládacích prvků. Ovládací prvek je abstraktní blok, který je možno umístit do jakéhokoliv jiného bloku v šabloně webu prostřednictvím editoru šablon (viz kapitola [4.9](#)). Každý blok může být rodičem neomezeného počtu ovládacích prvků, ale ovládací prvky nemohou obsahovat zanořené bloky nebo jiné ovládací prvky.

Samotná aktivace a nastavení modulů se provádí prostřednictvím administračního rozhraní, jako je tomu zvykem i u jiných systémů pro správu obsahu. Při aktivaci se navíc provede instalace modulu, pakliže se jedná o jeho první aktivaci.

Každý modul musí splňovat určité požadavky, aby jej systém byl schopen správně rozpoznat a použít všechny jeho ovládací prvky a stránky administrace. Požadavky jsou následující:

- V kořenovém adresáři modulu musí existovat soubor `info.php`.
  - V rámci tohoto souboru musí existovat třída `module_jmeno`, kde `jmeno` je strojové jméno modulu, které odpovídá jménu adresáře s modulem.
  - Tato třída musí implementovat rozhraní `module_info`, jehož požadavkem je implementace funkce `get_info()`. Tato funkce vrací pole s informacemi o daném modulu, jako je čitelné jméno, verze, seznam všech ovládacích prvků, které modul poskytuje, a jiné.
  - Dále musí tato třída implementovat funkci `install()`, která je zavolána v okamžiku aktivace modulu. Zde může každý modul vytvořit potřebné dodatečné tabulky v databázi, pokud je to potřeba.
- Pokud si modul žádá administrační stránky, je potřeba vytvořit adresář `admin` v kořenovém adresáři modulu. Všechny PHP soubory v rámci tohoto adresáře budou následně při instalaci modulu prozkoumány a pokud splňují požadavky, jsou zařazeny do administračního rozhraní webového portálu.

- Každá taková stránka musí obsahovat třídu `dashpage_modul_stranka`, kde `modul` je strojové jméno modulu a `stranka` je adresa stránky.
  - \* Znaky *pomlčka* („-“) a *vlnovka* („~“) je třeba v rámci jména třídy převést na znak *podtržítka* („\_“)
  - \* URL adresa stránky pro administraci modulu bude převzata z názvu souboru.
  - \* Pakliže chce autor modulu vytvořit podstránku, neděje se tak pomocí vytváření podadresářů, ale pomocí speciálního znaku *vlnovka* („~“) v názvu souboru. Soubor `settings~font.php` modulu `menu` bude mít ve výsledku adresu `/modules/menu/settings/font`.
  - \* Domovskou stránku administrace modulu je možno vytvořit pomocí souboru `index.php`. Její URL adresa bude následně pouze `/modules/menu`, nicméně výraz `index` stále musí být součástí jména třídy.
- Tato třída musí implementovat rozhraní `module_page`, které obsahuje funkci `get_info()`. Tato funkce vrací pole s informacemi o konkrétní stránce administrace. Je zde možno určit, jestli tato stránka má mít vlastní dlaždici v domovské stránce administrace webového portálu; pokud ano, jakou má mít barvu a ikonu; jaký je nadpis stránky či zda-li má být stránka vycentrovaná či širokoúhlá (např. editor šablon a témat).
- Dále musí tato třída obsahovat funkci `render_page()`, kterou systém zavolá v okamžiku, kdy administrátor navštíví patřičnou stránku a systém připraví záhlaví stránky a prostor pro vykreslení administrace.

Adresář s modulem musí být následně umístěn do složky `modules` v kořenovém adresáři na serveru webového portálu.

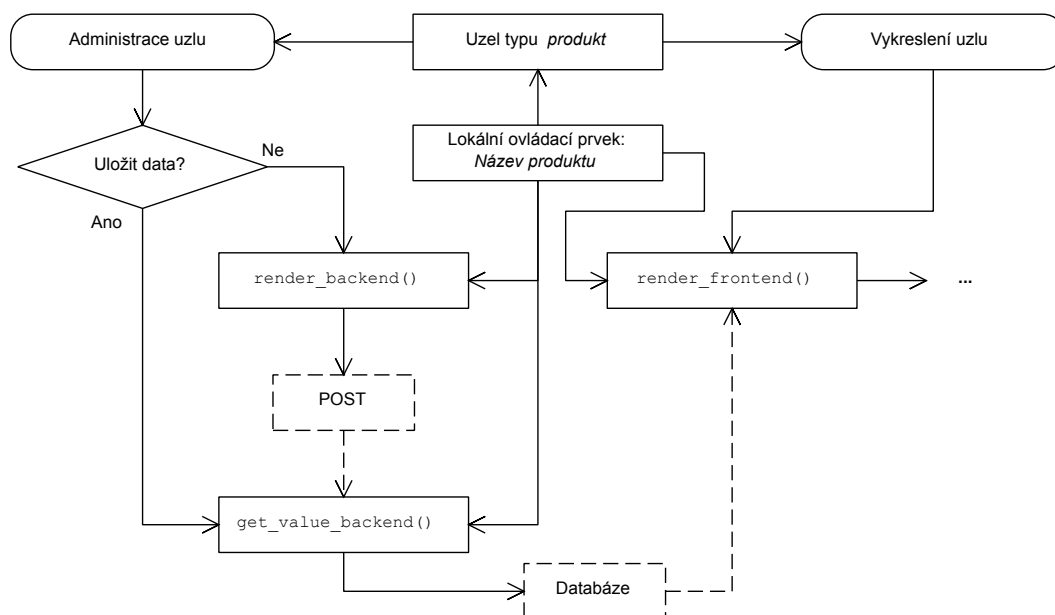
Pakliže modul poskytuje globální nebo lokální ovládací prvky (jejich rozdíl je popsán v kapitole 4.7.1, je potřeba toto definovat již v souboru `info.php` v kořenovém adresáři modulu v rámci výstupního pole funkce `get_info()`. Dále je v režii samotného modulu, aby systému poskytl všechny potřebné třídy reprezentující dané ovládací prvky. Ty můžou být implementovány přímo v souboru `info.php`, nebo klidně kdekoliv jinde a následně být domovskému suboru poskytnuty skrze PHP funkce `require` či `include`.

Všechny tyto třídy ovládacích prvků musí ovšem splňovat následující požadavky:

- Jméno třídy musí být ve formátu `module_jmeno_prvek`, kde `jmeno` je strojové jméno modulu a `prvek` jméno ovládacího prvku. Pokud tedy modul `menus` poskytuje ovládací prvek `widget_vertical_menu`, třída implementující tento prvek musí nést jméno `module_menus_widget_vertical_menu`.
- Třída musí implementovat rozhraní `widget`, které si vyžaduje implementaci funkcí `get_info()`, `render_backend()`, `render_frontend()` a `get_value_backend()`. Účel těchto funkcí je rozebrána v sekci 4.7.1.

#### 4.7.1 Chování ovládacích prvků

Ovládací prvky modulů mohou být dvojího druhu. Jedná se o prvky *lokální* a *globální*. Lokální prvky jsou takové ovládací prvky, které pracují s daty konkrétního uzlu webového portálu. Jedná se tedy o prvky, jako je nadpis článku, tagy, obrázek produktu, jméno osoby



Obrázek 4.6: Názorná ukázka práce s daty v rámci uzlu a konkrétního ovládacího prvku.

atp. Globální prvky jsou oproti tomu neměnné napříč celým webovým portálem. Může se jednat o logo společnosti, vertikální menu či kontaktní formulář.

V obou případech ovládací prvky implementují funkci `get_info()`, která vrací pole s informacemi o daném ovládacím prvku, jako je název, popis, odkaz do administrace či v neposlední řadě právě typ ovládacího prvku (lokální / globální).

Globální ovládací prvky se nevztahují ke konkrétním uzlům webového portálu a proto nepotřebují implementovat uživatelské rozhraní, skrze které by administrátor vyplňoval data při správě obsahu jednotlivých uzlů. Globální ovládací prvky musí implementovat pouze funkci `render_frontend()`, která je použita při vykreslování konkrétní stránky webu, na které se globální ovládací prvek nachází. Načtení potřebných dat rodičovského modulu je zcela v režii ovládacího prvku.

Lokální ovládací prvky ovšem fungují na jiném principu. Jelikož jsou relativní vzhledem ke konkrétnímu uzlu, musí poskytovat administrátorovi uživatelské rozhraní, skrze které je možno vyplnit data při správě konkrétních uzlů. Pokud například modul pro galerii obrázků poskytuje ovládací prvek *portrét*, který je následně přiřazen typu obsahu *osoba*, je nutné, aby správce webu měl možnost portrét ke každé osobě přiřadit.

K tomu slouží funkce `render_backend()`, která je zavolána při editaci položek určitého uzlu v administraci. Musí poskytovat - nejčastěji prostřednictvím formuláře - způsob, jak informaci k uzlu přiřadit. Samotnou hodnotu, která se jakožto jeden datový řetězec uloží do databáze, pak ovládací prvek poskytuje funkcí `get_value_backend()`, která je zavolána po uložení formuláře pro editaci položek uzlu. Podrobněji je tato problematika znázorněna ve schématu 4.6.

## 4.8 Nástroje pro vývojáře

System Constructo CMS má již ve svém jádře zabudované některé užitečné funkce, které mohou vývojáři modulů použít. Mezi takové funkce například patří funkce pro výtisk a

zpracování tabulek, která je popsána v kapitole 4.8.1, a dále třída pro výtisk a práci s formuláři, jež je rozebrána v kapitole 4.8.2.

Tyto funkce vždy začínají předponou `core_` a je možno je použít jak v rámci administračních stránek, tak i v rámci vykreslování ovládacích prvků uživateli na skutečných stránkách.

### 4.8.1 Vykreslování dynamických tabulek

Pokud administrátor potřebuje vytisknout výsledky SQL dotazu z databáze, může k tomu využít vestavěnou funkci `core_table_output`. Tato funkce si vyžaduje následující parametry:

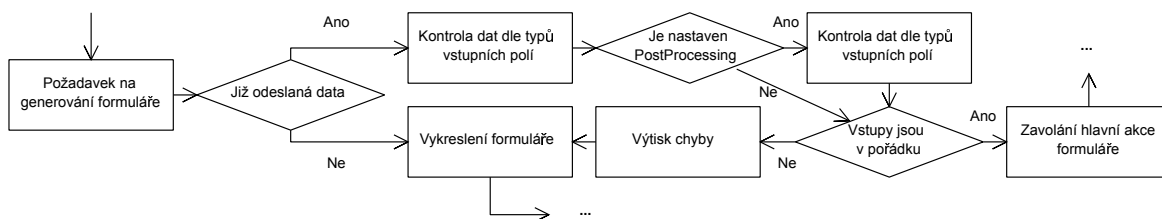
- `sql` - SQL dotaz poslaný do databáze.
- `header_array` - Pole s nadpisy sloupců, pokud administrátor nechce použít názvy použité v databázi. Formát pole pak je `array('sql_column' => 'Column name', 'column2' => 'Another column');`.
- `sort_array` - Pole s logickými hodnotami určujícími, které sloupce je možno seřazovat a které nikoliv. Standardní hodnota pro všechny sloupce je `true`, tedy pravda.
- `page_limit` - Číselná hodnota určující, kolik záznamů se má zobrazit na jednu stránku, než se výsledky rozdělí a objeví se prvek s odkazy na další stránky.
- `page_table_identifier` - Jedinečný identifikátor tabulky v rámci stránky. Používá se v případě, že je na aktuální stránce více, než jedna tabulka vytvořena touto funkcí a je potřeba rozlišit *GET* argumenty v situaci, kdy uživatel pracuje s oběma tabulkami současně.
- `where_statement` - Logická hodnota určující, zda-li SQL dotaz obsahuje klauzuli *WHERE*. Pokud ano, očekává se, že SQL dotaz předaný prvním argumentem bude obsahovat výraz `{{filter}}` na vhodném místě, který bude systém moci nahradit za klauzuli *AND* spolu s dosazením nové podmínky na základě filtru, který uživatel aplikoval.
- `filter_array` - Pole s logickými hodnotami, jež určují, na které sloupce může být aplikován filtr.
- `rewrite_array` - Pole s funkcemi, jež jsou aplikovány na jednotlivé záznamy v konkrétních sloupcích místo jejich standardního výtisku. Je tak možné některé hodnoty z databáze nahradit např. ikonami, je-li to žádoucí. Funkce dostane jako svůj jediný parametr hodnotu záznamu a očekává se, že vrátí náhradní řetězec, který bude vytisknut.
- `bonus_columns` - Pole s přidávanými sloupci na konec tabulky. Každý záznam pak obsahuje název sloupce a funkci pro výtisk jednotlivých buněk. Tato funkce má k dispozici všechny předchozí hodnoty na řádku. Používá se pro vykreslení sloupce s ovládacími prvky, jako „Upravit“, „Smazat záznam“, atp.

Pomocí zavolání této funkce a předání příslušných parametrů pak systém vygeneruje tabulky, jaké jsou vidět na obrázku 4.7. Stav tabulky (filtr, seřazení...) systém ukládá do *GET* parametrů URL adresy.



USER MANAGEMENT						
<b>Users:</b>						<input type="button" value="Add user"/>
UID	E-mail	User name	Created	State	Operations	
1		admin	2015-05-14 22:36:01	Normal	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
2		designer	2015-05-15 00:01:13	Normal	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
3		user0	2015-05-15 09:17:56	Normal	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
4		user1	2015-05-15 09:18:10	Normal	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
5		user2	2015-05-15 09:19:23	Normal	<input type="button" value="Edit"/>	<input type="button" value="Remove"/>
<b>User groups:</b>						<input type="button" value="Add group"/>
UGID	Group name	Readable name	Description	Users	Operations	
1	admins	Site administrators	?	1	<input type="button" value="Edit"/>	<input type="button" value="Remove"/> <input type="button" value="Manage users"/>
2	designers	Designers	?	1	<input type="button" value="Edit"/>	<input type="button" value="Remove"/> <input type="button" value="Manage users"/>
3	power_users	Power Users	!	3	<input type="button" value="Edit"/>	<input type="button" value="Remove"/> <input type="button" value="Manage users"/>

Obrázek 4.7: Příklad dynamických tabulek vytvořených pomocí systémové funkce `core_table_output`.



Obrázek 4.8: Jednotlivé kroky při generování a zpracování formuláře třídy `formCreator`.

#### 4.8.2 Práce s formuláři

Pro práci s formuláři byla v systému Constructo CMS vytvořena třída `formCreator`. Tato třída umožňuje tvorbu a následné zpracování hodnot z formuláře prostřednictvím vytvoření instance této třídy a použití následujících funkcí:

- `setHeader` - Nastaví text hlavičky nad formulářem.
- `setFooter` - Nastaví text patičky pod formulářem. Nepřepíše informaci o povinných polích.
- `addInput` - Přidá do formuláře další vstupní pole. Aktuálně jsou ve verzi 1.1.14 podporované následující typy polí: textové pole, pole pro heslo, pole pro e-mail, pole pro uživatele, textová víceřádková oblast, seznam. Při vytváření je třeba funkci předat tři parametry - typ pole, popis a pole s volitelnými argumenty. Prostřednictvím volitelných argumentů je možno nastavit některé podmínky vstupů, jako je minimální a maximální délka, původní hodnota, zda je pole povinné, zda má systém ověřit existenci e-mailu/uživatele v databázi, atp.
- `addMultiple` - Funguje stejně, jako `addInput`, ale uživatel má možnost u dané volby

SITE SETTINGS

Site settings: [Advanced settings](#)

**Site name:** \*

**Site description:** \*

**Anonymous user name:** \*

**Home page:** \*

**Path to page 404:** \*

**Site scheme:**

**Site theme:**

**Superpassword:**

\* = required field  
Leave superpassword field blank for no change.

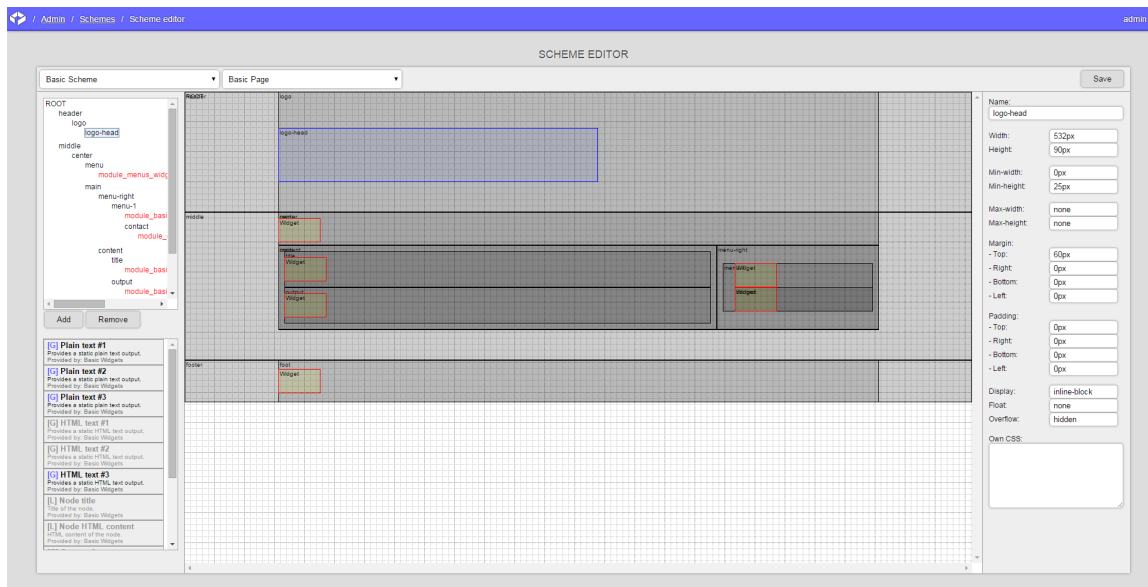
Obrázek 4.9: Příklad formuláře vytvořeného třídou `formCreator`.

přidávat neomezený počet polí. Např. skupina uživatelů může obsahovat libovolné množství uživatelů.

- `addButton` - Přidání vlastního tlačítka do formuláře. Tlačítko pro odeslání dat je součástí formuláře automaticky.
- `setSubmit` - Nastavení textu tlačítka pro odeslání dat.
- `addHidden` - Přidání skrytého prvku formuláře, jehož data budou po potvrzení odeslána.
- `addOutput` - Přidání prvku formuláře ve formě neměnného výstupu, jehož hodnota nemůže být editována.
- `setAction` - Nastavení funkce, jenž je spuštěna po odeslání dat, pokud jsou všechny vstupy v pořádku.
- `setPostProcessing` - Nastavení funkce, která se provede po automatické kontrole všech vstupů, ale před zavoláním závěrečné funkce `action`. Používá se pro dodatečnou kontrolu splnění pravidel vstupů, jenž nejsou definována jádrem systému.

Na obrázku 4.9 můžete vidět příklad formuláře, jenž byl vytvořen pomocí třídy `formCreator`. Schéma 4.8 pak znázorňuje jednotlivé kroky při požadavku o vygenerování a zpracování formuláře.

Některé systémové formuláře (např. přihlašování uživatelů) mají přednost před ostatními formuláři a jsou zpracovávány před samotným generováním webové stránky. Důvod je ten, že výsledná akce formuláře může ovlivnit vykreslení stránky. Pokud například administrátor přistupuje k administračnímu rozhraní, ale není přihlášen, je vykreslen formulář pro přihlášení. Po zadání údajů a potvrzení formuláře dochází k novému načtení stránky, kdy je formulář nejprve zpracován a až poté je rozhodnuto, zda má návštěvníkovi být zobrazena administrační nabídka či zda jsou jeho pravomoce nedostatečné.



Obrázek 4.10: Ukázka editoru šablon.

## 4.9 Editor šablon a editor témat

Součástí systému jsou dva editory - editor šablon a editor témat. Tyto editory umožňují kompletní správu rozvržení a vzhledu webového portálu, aniž by byl administrátor nucen psát či upravovat šablony v jazyce PHP.

### 4.9.1 Editor šablon

Účelem tohoto editoru je editace samotného rozvržení webu a jeho jednotlivých bloků. Šablony samy o sobě tedy ovlivňují výslednou strukturu HTML dokumentu, nikoliv však její samotné vyobrazení kromě pozicování<sup>3</sup>. Součástí šablony nejsou textury, barvy, fonty ani nic podobného.

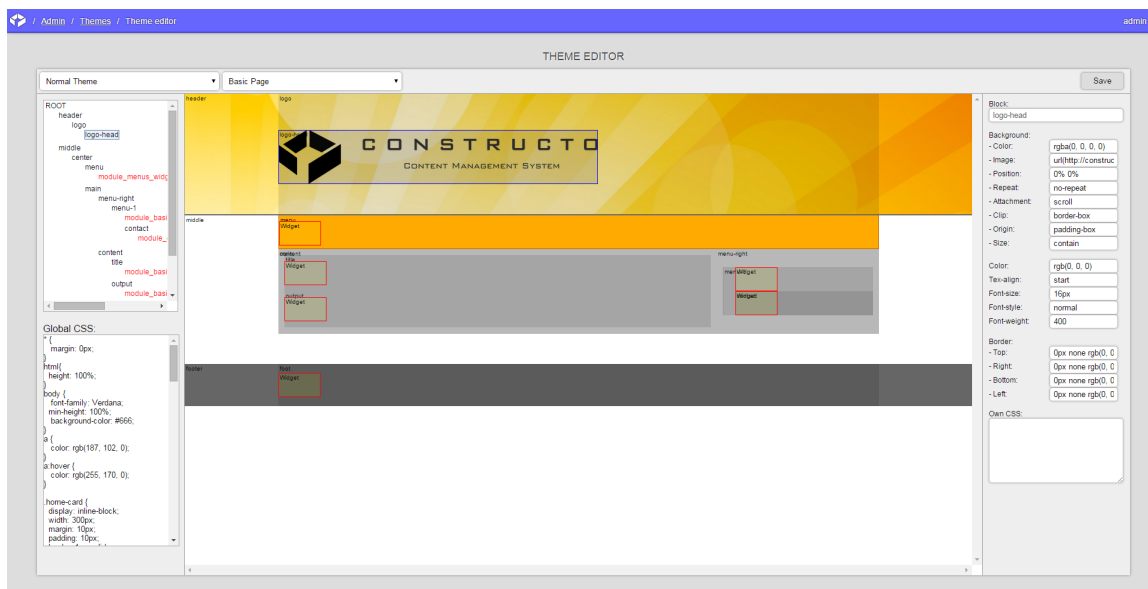
Prostřednictvím šablony je možno vytvářet a zanořovat bloky na stránce, do kterých je pak možné vkládat jednotlivé ovládací prvky modulů na konkrétní místa. Blok stránky může obsahovat libovolný počet zanořených bloků a libovolný počet ovládacích prvků modulů. Samotné ovládací prvky však nemohou obsahovat žádné vložené bloky ani jiné ovládací prvky.

Šablona se liší dle typu obsahu konkrétního uzlu. Je tedy možné definovat zcela odlišné rozvržení pro uzly typu „článek“ a pro uzly typu např. „produkt“.

Na obrázku 4.10 můžete vidět ukázkou aktuální šablony pro oficiální stránky systému Constructo. Editor sestává z celkem 4 oblastí. V horní oblasti je třeba nejprve zvolit šablonu, jež budeme editovat, a posléze typ obsahu, ke kterému se šablona má vztahovat (různé typy obsahu mohou používat různé šablony). Jakmile provedeme volbu, editor automaticky pomocí AJAXe načte strukturu námi zvolené šablony.

Levý panel následně sestává ze dvou menu. V prvním menu vidíme hierarchickou strukturu všech bloků a ovládacích prvků šablony. Tlačítka *Add* a *Remove* je možné přidávat a

<sup>3</sup>Problematika pozicování již spadá do CSS, nicméně už v editoru šablon je třeba mít kontrolu nad přesným umístěním elementů.



Obrázek 4.11: Ukázka editoru témat.

mazat vybrané bloky struktury. Všechny změny se okamžitě projevují v hlavní části editoru, kde je možno vidět náhled naší struktury v měřítku 1:1. Výběr elementů je možno provádět v rámci této hlavní oblasti, nebo v menu s hierarchivkou strukturou.

Ve spodní části levého menu se nachází seznam všech globálních a lokálních ovládacích prvků, které je možno v rámci naší šablony použít. Ovládací prvek do šablony je možno umístit výběrem nějakého bloku a následným kliknutím na tlačítko s ovládacím prvkem. Tím dojde ke vložení ovládacího prvku do vybraného bloku. Mazání ovládacích prvků se provádí stejně, jako mazání bloků.

V pravé části editoru se nachází oblast s vlastnostmi právě vybraného bloku. Je možné zde měnit CSS vlastnosti související s pozicováním bloku, jako je **padding**, **margin**, šířka či výška bloku atp. Není zde možno měnit samotný vzhled bloků, jako je pozadí, barva fontu, ohraničení a jiné vizuální vlastnosti. K tomu slouží editor témat, viz sekce 4.10.

Pakliže máme v editoru aktuálně vybraný některý ovládací prvek, v pravé části se místo blokových vlastností zobrazí informace o daném prvku (jaký modul jej poskytuje, jaký je rychlý odkaz do administrace...)

Uložení se provede kliknutím na tlačítko *Save* v pravém horním rohu editoru šablon.

#### 4.9.2 Editor témat

Editor témat slouží pro definování vzhledu ke struktuře, která je vytvářena prostřednictvím editoru šablon. Téma nijak neovlivní rozmístění jednotlivých elementů na stránce a jejich strukturu, pouze je možno upravit jejich grafický vzhled, jako jsou textury, barvy a jiné vlastnosti, jenž jsou realizovány prostřednictvím CSS.

Důvod oddělení témat od šablon je ten, že jedna šablona může ve výsledku mít více témat (např. jedno rozvržení webu může mít normální a vánoční vzhled, který se bude lišit pouze v některých texturách).

Na obrázku 4.11 je ke zhlédnutí editor témat. Ovládání je téměř identické, jako u editoru šablon; v horní části se nejprve vybere téma, jenž chceme editovat a následně typ obsahu, k jakému téma vytváříme. Po výběru se opět automaticky načte struktura našeho vybraného

tématu.

V levé části kromě hierarchické struktury (tentokrát bez tlačítek pro přidávání a odebrání bloků) je možno vidět místo seznamu ovládacích prvků oblast pro vlastní globální CSS kód, který je možno uplatnit v rámci tématu jako celku.

V pravé části pak vidíme odlišné vlastnosti právě vybraného bloku oproti editoru šablon; tentokrát již zde můžeme měnit vizuální aspekty všech částí webu, jako jsou pozadí, barvy, rámy atp. Případně je možno použít vlastní CSS kód provázaný ke konkrétnímu prvku pomocí textové oblasti v pravém dolním rohu.

## 4.10 Dodatečné nástroje

Při vytváření systému Constructo CMS bylo vytvořeno několik dalších nástrojů, jenž se ukázali být užitečnými, ač jejich implementace nebyla původně v plánu a o jejich tvorbě bylo rozhodnuto dodatečně v průběhu implementace jiných částí systému na základě nově vzniklých požadavků.

### 4.10.1 Správce souborů

Jakožto součást systému byl implementován vestavěný správce souborů, pomocí nějž je možno procházet, vytvářet a mazat adresáře na webovém serveru a nahrávat či stahovat samotné soubory.

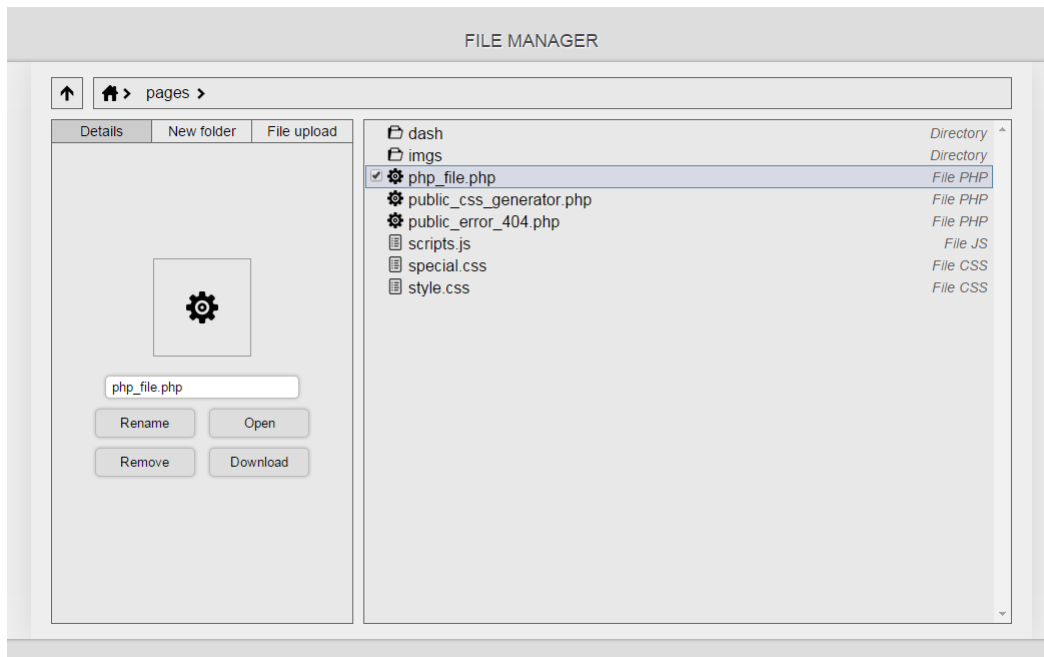
Správce vznikl z důvodu usnadnění nahrávání obrázků a jiných souborů na webový portál a kvůli zjednodušení procesu instalace modulů. Pro použití správce souborů není potřeba použít údaje pro FTP připojení, správce funguje pouze na základě PHP funkcí, jako `glob`, `unlink`, `mkdir` a podobných funkcí, jenž operují nad aktuálním souborovým systémem.

Z toho důvodu není možno správce souborů použít pro připojení se na jiný webový server. Nevýhodou je omezení pouze na webový server, na kterém systém Constructo CMS běží. Výhodou ovšem zůstává okamžitý přístup k souborům webu bez nutnosti použití služby FTP.

### 4.10.2 Správce logů

Součástí systému Constructo CMS je také správce logů, v němž je možno procházet všechny záznamy o aktivitách na webovém portálu, jako je např. přihlášení uživatele, nenalezení dokumentu či použití superhesla. Každý záznam obsahuje následující položky:

- **Typ události** - Poznámka, varování, chyba či kritická chyba.
- **Událost** - Krátký popis události.
- **Zpráva** - Podrobnější popis události.
- **Uživatel** - Který uživatel vyvolal danou událost.
- **IP adresa** - IP adresa uživatele, ze které uživatel provedl návštěvu.
- **URL adresa** - Adresa stránky, ze které k události došlo.
- **Datum a čas** - Kdy k události došlo.



Obrázek 4.12: Vestavěný správce souborů.

ID	Type	Event	Message	User	IP Address	URL Address	Time
65	Notification	User has logged in		admin	212.4.135.27	/~/admin?type=list	2015-05-20 01:51:56
64	Warning	Page not found		Anonymous	212.4.135.27	/files/source/Constructo%201.14.zip	2015-05-19 22:26:19
63	Warning	Page not found		Anonymous	212.4.135.27	/files/source/Constructo%201.14.zip	2015-05-19 22:25:54
62	Notification	User has logged in		admin	212.4.135.27	/~/admin	2015-05-19 20:24:08
61	Notification	User has logged in		admin	212.4.135.27	/~/admin	2015-05-19 18:03:15
60	Notification	User has logged in		admin	212.4.135.27	/~/admin	2015-05-19 12:36:01
59	Warning	Page not found		Anonymous	66.249.65.118	/robots.txt	2015-05-19 05:55:11
58	Warning	Page not found		Anonymous	66.249.65.118	/robots.txt	2015-05-18 22:00:03
57	Warning	Page not found		Anonymous	66.249.65.115	/robots.txt	2015-05-18 17:36:34
56	Warning	Page not found		Anonymous	66.249.65.121	/robots.txt	2015-05-17 21:14:32
55	Warning	Page not found		Anonymous	66.249.78.32	/robots.txt	2015-05-17 00:06:01
54	Warning	Page not found		Anonymous	66.249.78.207	/robots.txt	2015-05-16 21:08:13
53	Warning	Page not found		Anonymous	66.249.64.182	/m/	2015-05-16 16:32:47
52	Warning	Page not found		Anonymous	66.249.64.182	/mobile/	2015-05-16 16:30:17
51	Warning	Page not found		Anonymous	66.249.64.182	/robots.txt	2015-05-16 16:30:17
50	Warning	Page not found		Anonymous	66.249.64.182	/robots.txt	2015-05-16 05:51:02
49	Warning	Page not found		Anonymous	66.249.64.192	/robots.txt	2015-05-15 22:02:14
48	Warning	Page not found		admin	212.4.135.27	/files/source/Constructo%201.14.zip	2015-05-15 12:16:50
47	Warning	Page not found		admin	212.4.135.27	/files/source/Constructo%201.14.zip	2015-05-15 12:15:39
46	Warning	Page not found		admin	212.4.135.27	/files/source/Constructo%201.14.zip	2015-05-15 12:15:05
45	Warning	Page not found		admin	212.4.135.27	/files/source/Constructo%201.14.zip	2015-05-15 12:14:04
44	Warning	Page not found		admin	212.4.135.27	/files/source/Constructo%201.14.zip	2015-05-15 12:12:32
43	Warning	Page not found		admin	212.4.135.27	/download	2015-05-15 12:04:05
42	Warning	Page not found		admin	212.4.135.27	/download	2015-05-15 11:58:05
41	Warning	Page not found		admin	212.4.135.27	/gallery	2015-05-15 08:20:50
40	Warning	Page not found		Anonymous	212.4.135.27	/	2015-05-15 06:33:52
39	Warning	Page not found		admin	212.4.135.27	/	2015-05-15 06:33:50
38	Warning	Page not found		admin	212.4.135.27	/	2015-05-15 06:32:52
37	Notification	User has logged in		admin	212.4.135.27	/~/admin	2015-05-15 06:09:20
36	Warning	Page not found		admin	212.4.135.27	/	2015-05-15 00:31:38
35	Warning	Page not found		Anonymous	212.4.135.27	/home2	2015-05-15 00:28:48
34	Warning	Page not found		admin	212.4.135.27	/~/admin/theme/files/presentation/imgs/background.jpg	2015-05-15 00:26:54
33	Warning	Page not found		Anonymous	212.4.135.27	/	2015-05-15 00:07:59
32	Warning	Page not found		Anonymous	212.4.135.27	/download	2015-05-15 00:03:11
31	Warning	Page not found		Anonymous	212.4.135.27	/	2015-05-15 00:03:04
30	Warning	Page not found		admin	212.4.135.27	/	2015-05-14 23:59:32
29	Warning	Page not found		admin	212.4.135.27	/	2015-05-14 23:58:03
28	Warning	Page not found		admin	212.4.135.27	/	2015-05-14 23:57:45
27	Warning	Page not found		admin	212.4.135.27	/	2015-05-14 23:57:44
26	Warning	Page not found		admin	212.4.135.27	/	2015-05-14 23:57:38

Obrázek 4.13: Ukázka systémových logů.

Vytvoření nových záznamů je možno provést funkcí `core_log()` i z libovolného modulu. Mazat záznamy však není možné ani ze samotného správce logů. Všechny záznamy o aktivitách uživatelů tak zůstávají v záznamech na vždy. V některé z následujících verzí se plánuje automatické promazávání záznamů logu po vypršení určité časové lhůty (např. půl roku).

### 4.10.3 Správce systémových konstant

Pro absolutní kontrolu nad systémem Constructo CMS byl vytvořen správce systémových konstant. Většina nastavení je administrátorovi poskytnuta v rámci stránky *Site Settings*, popřípadě *Advanced Settings*, nicméně pokud je nutno změnit některou ze systémových konstant ručně, správce systémových konstant to umožňuje. Změna některých záznamů však může být mírně nebezpečná a je tedy doporučeno tento úkon provádět jen správci, kteří mají se systémem zkušenosti.

## Kapitola 5

# Testování a první nasazení

Testování systému nejprve probíhalo na lokálním stroji, kde byl nainstalován interpret PHP, databáze MySQL a následně i samotný systém Constructo CMS.

Po čase se ale ukázalo, že testování bez skutečné komunikace přes internet není ideální a nepřináší reálné výsledky a lokální interpret PHP se navíc nechoval zcela identicky k interpretům, jenž jsou používány na reálných webových serverech. Jednalo se zejména o překlad adres a relativních cest v rámci systémových souborů nebo o získávání IP adresy návštěvníka pro účelu logování informací, které bylo na lokálním stroji zcela nefunkční.

Z toho důvodu byla v rámci bakalářské práce koupena doména `constructo.cz` spolu s hostingem u společnosti *WEDOS*, kde byl systém Constructo CMS nainstalován a veškeré testování a vývoj tak již probíhal na skutečném webovém serveru.

Aktuálně se systém nachází ve verzi 1.1.14 a ze stránek `constructo.cz` se stal oficiální portál pro tento systém. Náhled aktuálních stránek je dostupný na obrázku [5.1](#)

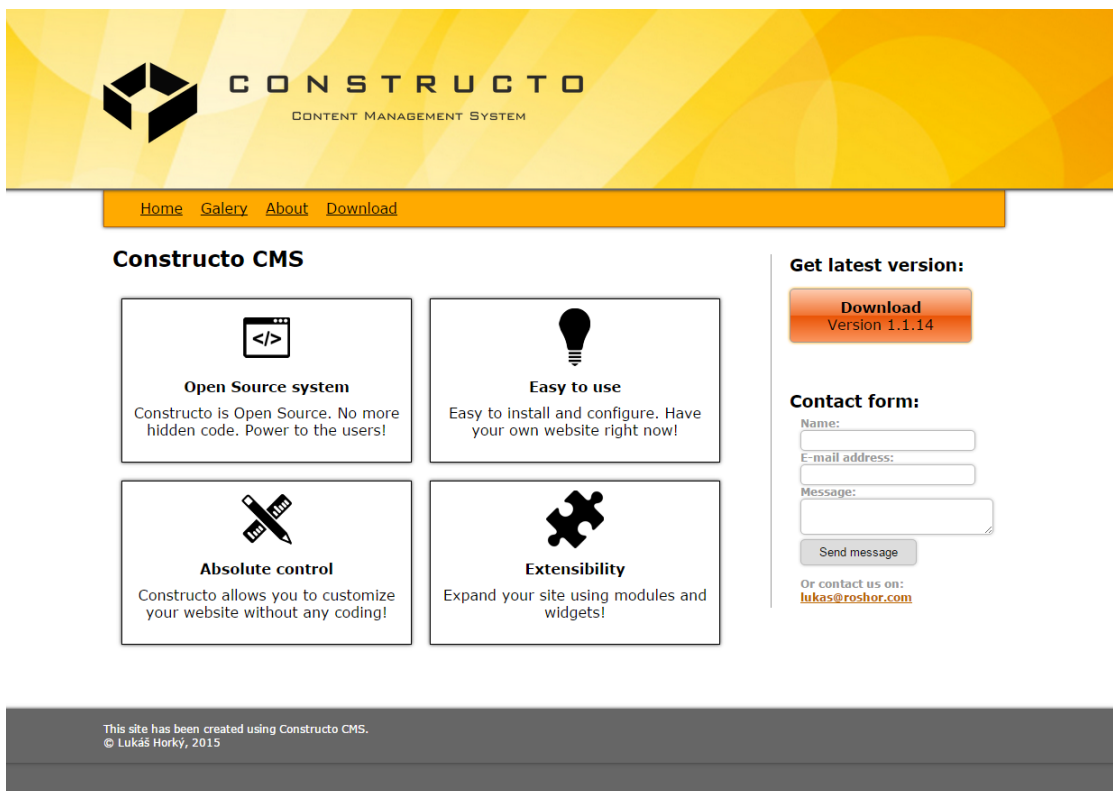
### 5.1 Podněty uživatelů

Vzhledem k podstatě a rozsahu této bakalářské práce bylo získání podnětu uživatelů, jenž byli ochotni systém Constructo CMS vyzkoušet, mírně komplikované z důvodu menšího vzorku zúčastněných osob. I tak se ale podařilo získat některé cenné náměty na vylepšení, jenž budou zahrnuty v budoucích verzích systému. Jedná se zejména o:

- Podpora vícejazyčných webových portálů.
- Způsob automatizování tvorby uživatelů či možnost jejich samostatné registrace (momentálně je možno řešit prostřednictvím modulů).
- Možnost referencovat uzly mezi sebou navzájem (taktéž je možno řešit vytvořením vlastního lokálního ovládacího prvku modulu).
- Funkce pro automatické zasílání novinek uživatelům.
- Šablony, témata, více vestavěných modulů...

Na oficiálních stránkách systému Constructo CMS byl vytvořen kontaktní formulář, jehož prostřednictvím mohou uživatelé stále posílat nové připomínky a náměty na zlepšování. Bude na ně brán zřetel při tvorbě nových verzí systému.





Obrázek 5.1: Ukázka oficiálních stránek systému Constructo CMS.

## Kapitola 6

# Závěr

Cílem této práce bylo vytvoření modulárního systému pro správu obsahu ve webovém prostředí, který umožňuje snadné vytvoření a údržbu webových portálů.

Kromě klasických nástrojů pro správu obsahu a uživatelů systém Constructo CMS obsahuje vestavěný editor šablon pro vytváření a editaci struktury bloků a ovládacích prvků modulů a editor témat pro správu vizuálního vzhledu webového portálu. Podařilo se úspěšně navrhnout a implementovat jádro systému tak, aby bylo funkcionality webu možno rozšiřovat prostřednictvím modulů. Problematika strukturalizace a prezentace informací uživateli byla vyřešena tvorbou typů obsahu uzlů a lokálními (a částečně i globálními) ovládacími prvky modulů, jež je možno v rámci editoru šablon libovolně rozmisťovat.

Systém umožňuje v rámci správy uživatelů také vytváření uživatelských skupin. Právo k jednotlivým prvkům administrace webového portálu je možno přiřadit jak konkrétnímu uživateli, tak i uživatelské skupině.

Součástí systému byl dále implementován i správce souborů pro snadné nahrávání obrázků a jiného obsahu na webový server a pro snadnou instalaci nových modulů. Správa samotných modulů, jejich aktivace a deaktivace je prováděna skrze správce modulů. Po aktivaci může modul poskytovat libovolné množství ovládacích prvků, jež je možno použít pro rozšíření struktury konkrétních typů obsahu (tj. uzlů) webu.

Systém dále obsahuje logovací nástroj umožňující zaznamenávat libovolné události ze strany návštěvníků (např. přihlášení, nenalezený dokument, použití superhesla...) a také editor systémových konstant pro absolutní kontrolu nad webovými stránkami.

Aktuální verze systému 1.1.14 splňuje všechny formální požadavky, jež byly na systém kladeny v počátcích jeho tvorby. Do budoucna je v plánu na základě uživatelských podnětů možnost vytváření vícejazyčných webových portálů a rozšíření sady vestavěných modulů.

Systém je možno stáhnout z oficiálních stránek [www.constructo.cz](http://www.constructo.cz).

# Literatura

- [1] Metro (uživatelské rozhraní). 2013, [online]. [cit. 2015-05-11] Dostupné z: <http://goo.gl/rt10s3>.
- [2] Brown, T. B.; Butters, K.; Panda, S.: *HTML5 Okamžitě*. Computer Press, 2014, iISBN 978-8-025-14296-7.
- [3] Böhmer, M.: *Návrhové vzory v PHP*. Computer Press, 2012, iISBN 978-8-025-13338-5.
- [4] Dawson, A.: *Výjimečný webdesign: jak tvořit osobité, přitažlivé, použitelné weby*. Computer Press, 2012, iISBN 978-8-025-13719-2.
- [5] Kofler, M.; Öggl, B.: *PHP 5 a MySQL 5: průvodce webového programátora*. Computer Press, 2007, iISBN 978-8-025-11813-9.
- [6] Lazaris, L.: *CSS Okamžitě*. Computer Press, 2014, iISBN 978-8-025-14176-2.
- [7] Mening, R.: Wordpress vs Joomla vs Drupal. 2013, [online]. [cit. 2015-05-11] Dostupné z: <http://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>.
- [8] Pieters, S.: HTML5 Differences from HTML4. <http://www.w3.org/TR/2014/NOTE-html5-diff-20141209/>, 2014, [online]. [cit. 2015-05-11].
- [9] Polzer, J.: *Drupal 7 - Podrobný průvodce tvorbou a správou webů*. Computer Press, 2011, iISBN 978-8-025-13445-0.

# Příloha A

## Metriky kódu

- Velikost nezabalených zdrojových kódů: 863 kB
- Velikost .ZIP archívu se zdrojovými kódy: 352 kB
- Počet .PHP souborů: 70
- Počet .CSS souborů: 6
- Počet .JS souborů: 6
- Počet ostatních souborů: 31
- Napsáno řádků kódu: cca 12 000 (včetně komentářů)

# Příloha B

## Obsah CD

Adresářová struktura:

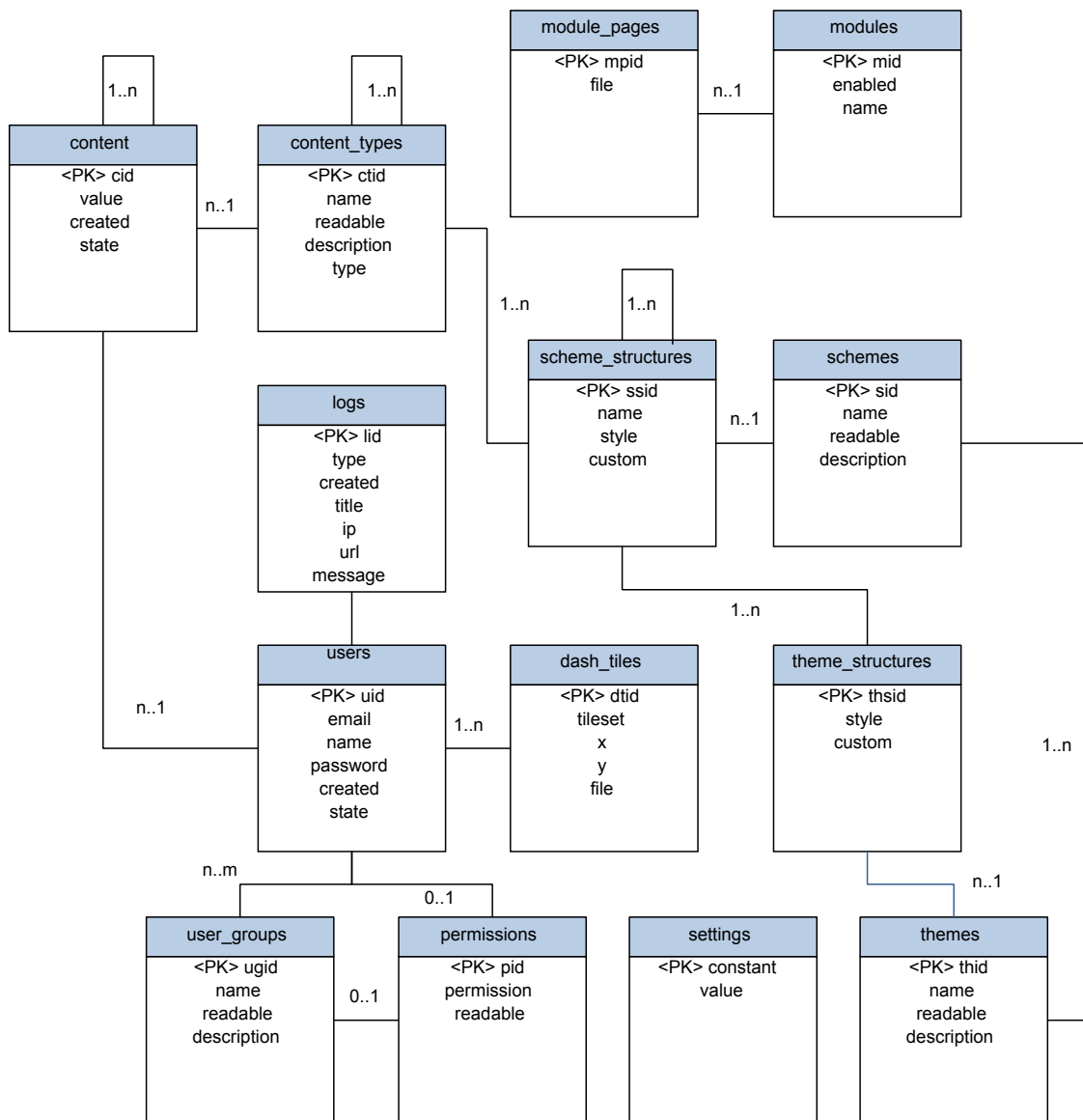
- **Source** - adresář se zdrojovými kódy systému Constructo CMS 1.1.14.
- **Documentation** - adresář s dokumentací bakalářské práce.
  - PDF - výsledný PDF soubor.
  - LaTeX - zdrojové kódy dokumentace v L<sup>A</sup>T<sub>E</sub>Xu.
- **Video** - adresář s demonstračním videem.
- **Poster** - adresář s plakátem.

## Příloha C

# ER diagram systému

Informace:

- Počet entit: 14
- Vztahy mezi entitami: 15



Obrázek C.1: ER diagram systému Constructo CSM. Nejsou zde zahrnuty entity vytvořené po instalaci některých modulů.

**Příloha D**

**Plakát**





# CONSTRUCTO

CONTENT MANAGEMENT SYSTEM

**Bakalářská práce:** Modulární CMS

**Autor:** Lukáš Horký (xhorky21)

**Vedoucí:** Ing. Igor Szöke, Ph.D

Vysoké učení technické v Brně,  
Fakulta informačních technologií

[www.constructo.cz](http://www.constructo.cz)

~ 2015 ~

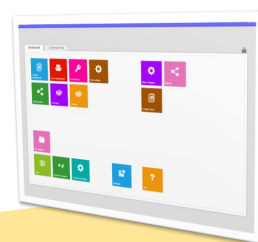
## 1.) Co je Constructo CMS?

Constructo CMS je nový modulární systém pro správu obsahu ve webovém prostředí. Umožňuje snadné vytvoření a konfiguraci webového portálu. Jedná se samozřejmě o open source software a instalace ti zabere jen chvilíčku!



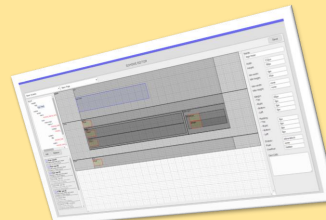
## 2.) Proč bych to měl použít?

Protože Constructo CMS ti umožní vytvořit stránky přesně na míru. Obsahuje vestavěný editor šablon a editor témat, kde si vyrobíš vlastní strukturu a vzhled webovek tak, jak potřebuješ. Nemusíš napsat ani řádku v PHP a nejsi nucen spoléhat se na šablony, které vyrobil někdo jiný.



## 3.) A co ještě to umí?

Spoustu dalších věcí. Systém obsahuje kromě editorů šablon a témat třeba vestavěný správce souborů, pokročilou správu uživatelů a uživatelských skupin, nástroj pro editaci struktury webu a jeho uzlů a mnoho dalšího. A co to neumí, to si doinstaluješ přes moduly.



## 4.) Kde to vůbec seženu?

Constructo CMS seženeš na oficiálních webových stránkách [www.constructo.cz](http://www.constructo.cz). Systém stáhneš a hned ho můžeš použít! Nic neplatíš, na nic nečekáš. Žádná registrace ani jiné nesmysly. Udělej si svoje webovky ještě dnes!



Více informací na:

[www.constructo.cz](http://www.constructo.cz)

Obrázek D.1: Plakát k systému Constructo CMS.