

Katedra informatiky  
Přírodovědecká fakulta  
Univerzita Palackého v Olomouci

# BAKALÁŘSKÁ PRÁCE

Webová aplikace pro publikování sémanticky  
organizovaného obsahu.



2016

Vedoucí práce:  
Mgr. Martin Trnečka

Tomáš Mikula

Studijní obor: Aplikovaná informatika,  
prezenční forma

## **Bibliografické údaje**

Autor: Tomáš Mikula  
Název práce: Webová aplikace pro publikování sémanticky organizovaného obsahu.  
Typ práce: bakalářská práce  
Pracoviště: Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci  
Rok obhajoby: 2016  
Studijní obor: Aplikovaná informatika, prezenční forma  
Vedoucí práce: Mgr. Martin Trnečka  
Počet stran: 36  
Přílohy: 1 CD/DVD  
Jazyk práce: český

## **Bibliographic info**

Author: Tomáš Mikula  
Title: Web application for publishing semantically organized content.  
Thesis type: bachelor thesis  
Department: Department of Computer Science, Faculty of Science, Palacký University Olomouc  
Year of defense: 2016  
Study field: Applied Computer Science, full-time form  
Supervisor: Mgr. Martin Trnečka  
Page count: 36  
Supplements: 1 CD/DVD  
Thesis language: Czech

## Anotace

*Výsledkem práce je webová aplikace Semantic CMS pro publikaci článků na internetu zaměřená na sémantickou organizaci obsahu. Sémantická navigace je realizována za pomoci sémantické mapy témat, která vznikla jako reakce na nedostatky v oblasti navigace u aktuálně používaných webových aplikacích pro správu obsahu. Webová aplikace Semantic CMS je navržena s cílem malého a jednoduše ovladatelného redakčního systému pro správu obsahu. K realizaci byl použit webový framework Django.*

## Synopsis

*The result of my thesis is a web application Semantic CMS used for the publication of an articles on the internet while focused on semantic organization of the content. Semantic navigation is created with the help of semantic map of topics, which was developed as a response to deficiencies in the navigation area of the currently used applications for content administration. Web application Semantic CMS is developed with the goal of small and easy controlled content managment system. For the execution of the project was used web framework Django.*

**Klíčová slova:** webová aplikace; sémantická navigace; sématická mapa

**Keywords:** web application; semantic navigation; semantic map

Děkuji vedoucímu bakalářské práce Mgr. Martinu Trnečkovi a aktivní komunitě frameworku Django, za užitečné rady.

*Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.*

datum odevzdání práce

podpis autora

# Obsah

<b>1 Motivace</b>	<b>7</b>
1.1 Problémy webových stránek s ohledem na sémantiku . . . . .	7
1.2 Organizace článků . . . . .	8
1.3 Možné vylepšení – Sémantické kategorie . . . . .	9
<b>2 Teoretický koncept</b>	<b>10</b>
2.1 Systém sémantických kategorií . . . . .	10
2.2 Systém typu obsahu . . . . .	11
2.3 Systém klíčových slov . . . . .	11
<b>3 Semantic CMS – Webová aplikace</b>	<b>12</b>
3.1 Design projektu . . . . .	12
3.2 Použité technologie . . . . .	14
3.3 Seznam aplikací v Semantic CMS . . . . .	16
3.4 Podrobný popis – Administrační část . . . . .	18
3.4.1 Správce obsahu . . . . .	18
3.4.2 Vytvoření/Editace článku . . . . .	19
3.4.3 Editor sémantické mapy . . . . .	20
3.4.4 Správce typů obsahu . . . . .	22
3.4.5 Nastavení . . . . .	22
3.4.6 Přihlášení/Zapomenuté heslo . . . . .	23
3.5 Podrobný popis – Část pro čtení článků . . . . .	23
3.5.1 Hlavní stránka . . . . .	23
3.5.2 Detailní výpis článku . . . . .	24
3.5.3 Detail článku . . . . .	24
3.6 Instalace . . . . .	26
3.7 Demo Semantic CMS . . . . .	27
<b>4 Možnosti rozvíjení aplikace</b>	<b>28</b>
<b>Závěr</b>	<b>32</b>
<b>Conclusions</b>	<b>33</b>
<b>A Obsah přiloženého CD/DVD</b>	<b>34</b>
<b>Literatura</b>	<b>35</b>

## Seznam obrázků

1	Sémantická mapa – reprezentace pomocí DAG . . . . .	11
2	Sémantická mapa – První koncept na papír . . . . .	12
3	Sémantická mapa – První grafický koncept . . . . .	13
4	Sémantická mapa – Finalní grafický návrh . . . . .	13
5	čtení článků na mobilním telefonu . . . . .	14
6	Content Manager – Rychlé akce v přehledu článků . . . . .	18
7	Create new article – Základní záhlaví . . . . .	19
8	Create new article – Našeptávání klíčových slov . . . . .	20
9	Semantic editor – Ovládací prvky editoru . . . . .	21
10	Hlavní stránka – Náhled sémantické mapy . . . . .	23
11	Detail článku – Informace o autorovi . . . . .	25
12	Detail článku – Související články . . . . .	25

## Seznam tabulek

1	Kontroléry pro editor sémantické mapy . . . . .	21
---	---	----

## Seznam zdrojových kódů

1	Výňatek ze souboru Base.py – Použité aplikace třetích stran . . . . .	18
2	Příklad – Vrcholy grafu po serializaci . . . . .	22
3	Ukázkový produkční konfigurační soubor . . . . .	30
4	Ukázkový produkční konfigurační soubor . . . . .	31

# 1 Motivace

V první kapitole se budu věnovat sémantické navigaci a motivaci pro vytvoření webové aplikace pro publikování sémanticky orientovaného obsahu. Na webových stránkách se aktuálně vyskytuje několik problému, které mě dovedly k vytvoření základního konceptu mé bakalářské práce. Tyto problémy, které popíši v následujících sekcích mají přímý i nepřímý dopad na kvalitu navigace na webu. Později se také budu věnovat návrhu možných řešení daných problémů.

## 1.1 Problémy webových stránek s ohledem na sémantiku

### Nejednoznačnost webové prezentace

Nejednoznačnost webové stránky či blogu patří k jednomu z největších problémů, který mě vedl k psaní této práce.

Nový návštěvník často nemá dostatečné informace o zaměření právě prohlížených webových stránek. Tento problém se snaží mnoho webových stránek řešit za pomoci obecného popisu témat. Popis však často neodpovídá realitě a je nepřesný.

Návštěvník v tomto případě potřebuje více času, aby se mohl zorientovat a zhodnotit, zda má o daný web vůbec zájem. Pro dlouhodobého čtenáře je tento problém irelevantní. Ideální by tedy bylo, aby nově příchozí návštěvník viděl, jaká témata jsou pro daný web stěžejní a jak se vyvíjel samotný obsah webu.

### Hledání relevantních článků

Hledání článků patří k nejčastější aktivitě návštěvníka webové stránky. Aktuálně používané systémy navigace neposkytují návštěvníkovi dostatečné informace o propojení jednotlivých témat.

V dnešní době je tento problém řešen na straně internetového vyhledávače. Většina přístupů čtenáře na web je přímo skrze vyhledávanou frázi ve vyhledávači. I přes to, že dnešní vyhledávače dosahují dobrých výsledků, zastávám názor, že by měl být problém řešen rovněž na samotné webové stránce. Jednoduše zobrazitelný vzájemný vztah jednotlivých článků (kategorií) usnadní návštěvníkovi orientaci na webu.

### Doporučování souvisejících článků

Návštěvník, který již našel odpovídající článek, potřebuje jednoduchý a přesný systém doporučení souvisejících článků, které by jej mohly zajímat. Takový systém má pozitivní vliv na prodloužení stráveného času návštěvníka na webové stránce.

Dnes je doporučování článků běžnou záležitostí. Často je však počet hodnot omezen na úkor přesnosti. V případě použití sémantické organizace článků můžeme určit příbuzné články jednodušeji a rychleji.

## 1.2 Organizace článků

Webová stránka vyžaduje určitý systém organizace článků, sloužící k zvýšení přehlednosti v publikovaném obsahu. Systém organizace článku určuje jakým způsobem bude čtenář interagovat s obsahem webu. Dnes se jich na internetu vyskytuje několik, nejčastěji v zájemné kombinaci. Každý systém poskytuje určitá pozitiva a negativa. V této kapitole se pokusím o jejich shrnutí.

### Systém archivu

Jednotlivé články jsou rozděleny dle data publikování, nejčastěji dle roku a měsíce. Tento systém neposkytuje žádné sémantické informace. Slouží pouze k vyhledávání článků pro dané období.

### Systém kategorií

Jednotlivé články jsou umístěny do kategorií. Článek může spadat do více kategorií najednou. Pro zachování jednoduchosti a přehlednosti se často volí menší počet obecnějších kategorií. Obecnost kategorií na druhou stranu vede k menší přesnosti. Jedná se však o jednoduchý a poměrně praktický systém organizace článků. Vícenásobné zanořování kategorií přináší větší přesnost a určitý stupeň vzájemného vztahu jednotlivých kategorií. Na druhou stranu však snižuje výslednou přehlednost pro návštěvníka.

### Systém klíčových slov

Každý článek má přiřazené jedno nebo více klíčových slov, které libovolným způsobem souvisí s daným článkem. Konkrétní klíčové slovo se v systému vyskytuje pouze jednou. Zaručuje tak jednoznačnou identifikaci článků, které mají klíčové slovo přiřazeno. Dnes velice oblíbený a rozšířený systém. Návštěvník může ve všech klíčových slovech vyhledávat, nebo procházet nejpopulárnější z nich.

Systém klíčových slov neumožňuje zaznamenat relevanci určitého klíčového slova vzhledem k článku. Článek tedy může obsahovat jak velice relevantní, tak velice tématicky vzdálená klíčová slova.

### Systém zajímavých/populárních článků

V angličtině známý jako „featured articles“. Systém, který upřednostňuje některé články (dle určitého aspektu) více než ostatní. Je možné využít manuální volbu, kdy autor sám označí vybrané články, nebo využít jiných dat (například počtu přečtení) k určení vybraných článků. Tento systém může vést k ukrytí článků, o které nebyl v čas jejich publikování velký zájem.



### 1.3 Možné vylepšení – Sémantické kategorie

Výše zmíněné systémy postrádají hlubší sémantickou organizaci. Systém sémantických kategorií řeší tento problém pomocí sémantické mapy. Jednotlivé kategorie mají jasně udaný vzájemný vztah (rodiče, potomci). Detailnějšímu popisu tohoto systému se budu věnovat v sekci [2.1](#).

## 2 Teoretický koncept

Na základě problémů popsaných v předchozí kapitole jsem navrhnul teoretický koncept. Hlavním cílem tohoto konceptu je poskytnout čtenáři přehlednější navigaci na webu. Pro splnění tohoto cíle bylo nutné vybrat systém organizace článků. Jako nejlepší kombinace již rozšířených systémů a potřebného vylepšení se jeví spojení klíčových slov, typu článků a sémantických kategorií. V následujících sekcích budu podrobněji popisovat jednotlivé dílčí systémy.

### 2.1 Systém sémantických kategorií

Stěžejní systém pro sémantickou navigaci na webové stránce. Sémantické kategorie jsou reprezentovány pomocí sémantické mapy, která zachycuje vztahy mezi jednotlivými kategoriemi. Na rozdíl od klasické struktury (viz. sekce 1.2) kategorie poskytuje jednoduchý a intuitivní mechanismus pro navigaci na webu. Čtenář vidí kategorie a jejich vzájemné vztahy ve větším kontextu. Procházení takové struktury kategorií by mělo být podobné prohlížení klasické mapy.

#### Sémantická mapa

Vnitřní reprezentace pomocí orientovaného acyklického grafu (Directed acyclic graph – DAG [1]). Kategorie jsou v grafu reprezentovány pomocí vrcholů (viz. obrázek 1).

S využitím DAG můžeme implementovat systém, ve kterém jednoduše udržujeme informaci o vzájemném vztahu kategorií. Jednotlivé hrany grafu značí vzájemný vztah mezi dvěma kategoriemi. Velikost jednotlivých vrcholů pak poměrově značí počet potomků.

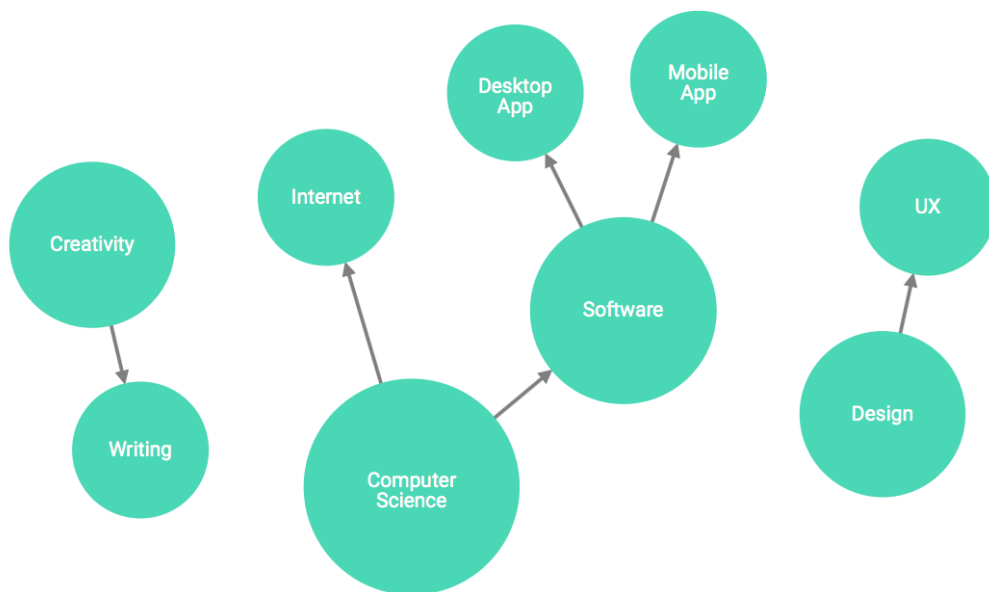
Každý článek může být umístěn do libovolného počtu kategorií. Automaticky je článek součástí nadřazených kategorií, avšak s nižším významem, než kategorie přímo přiřazené.

Takto tvořená mapa poskytuje návštěvníkovi jednoduchý nástroj pro orientaci v obsahu. Odpadá zde problém zanořených kategorií, protože uživatel je schopný vidět celou mapu (případně dostatečně velkou část).

#### Problém tvorby sémantické mapy

Celková kvalita systému se odvíjí od přesnosti samotné sémantické mapy. V mém konceptu je sémantická mapa vytvářena samotným autorem obsahu. Tento přístup vede k přesnější a lidštější podobě výsledné sémantické mapy. Má však i své úskalí.

Sémantická mapa by neměla být navrhovaná s předstihem. Měla by vždy odpovídat aktuálnímu stavu obsahu. Pokud tento přístup nedodržíme, nastává problém nejednoznačnosti (problém popsán v sekci 1.1) a čtenář nemá aktuální informace o obsahu webové stránky. Pokud vytvoříme příliš obecnou sémantickou mapu, nastává problém obecných kategorií (problém popsán v sekci 1.2). Jelikož



Obrázek 1: Sémantická mapa – reprezentace pomocí DAG

sémantickou mapu tvoří sám autor, je nutné poskytnout dostatečně jednoduchý a intuitivní editor sémantické mapy (editor je popsán v sekci 3.4.3).

### Určení podobných článků s využitím sémantické mapy

S využitím sémantické mapy můžeme jednoduše porovnat tématické překrytí dvou a více článků a určit tak jejich podobnost. Tato výhoda se dá použít pro doporučení dalších článků návštěvníkovi (viz. sekce 3.5.3).

## 2.2 Systém typu obsahu

Systém doplňující sémantické kategorie o další sémantickou informaci – typ obsahu (např. recenze, úvaha). Každý článek může mít přiřazený svůj typ. Jednotlivé typy článků si opět definuje samotný autor obsahu.

## 2.3 Systém klíčových slov

Systém popsán v sekci 1.2. Slouží k rozšíření sémantických kategorií. Zamezuje vytváření jednoúčelových kategorií v sémantické mapě. Klíčová slova autor používá v případě, že nepotřebuje vytvářet samostatnou kategorii v sémantické mapě.

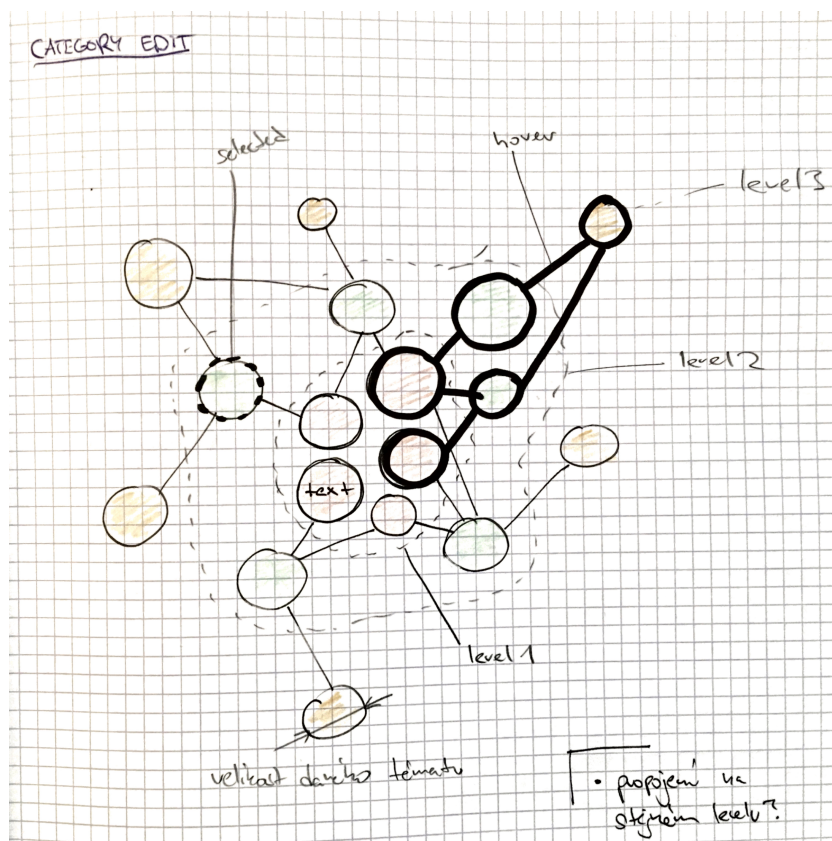
### 3 Semantic CMS – Webová aplikace

Dle teoretického konceptu v předchozí kapitole jsem navrhl webovou aplikaci s názvem „Semantic CMS“ (CMS – Content Management System). Hlavní důraz byl kladen na jednoduchost aplikace a přehlednost navigace. Semantic CMS poskytuje základní funkce redakčního systému rozšířené o sémantickou navigaci. Aplikace se dělí na dvě hlavní části. Administrační část pro autora článků (backend) a samotnou část pro čtení článků (frontend).

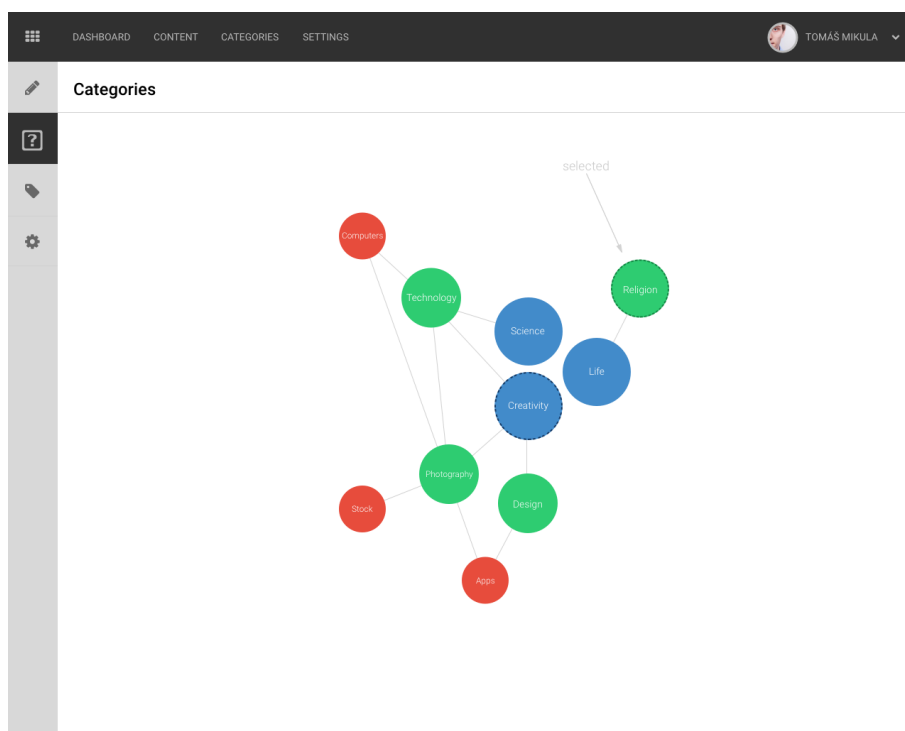
#### 3.1 Design projektu

Dříve než se dostanu k technickým podrobnostem, rád bych se věnoval samotnému návrhu projektu. Velký důraz jsem kladl na samotnou přívětivost aplikace. Ať už se jedná o pochopitelnost ovládacích prvků, nebo pohodlnost při používání na různých zařízeních.

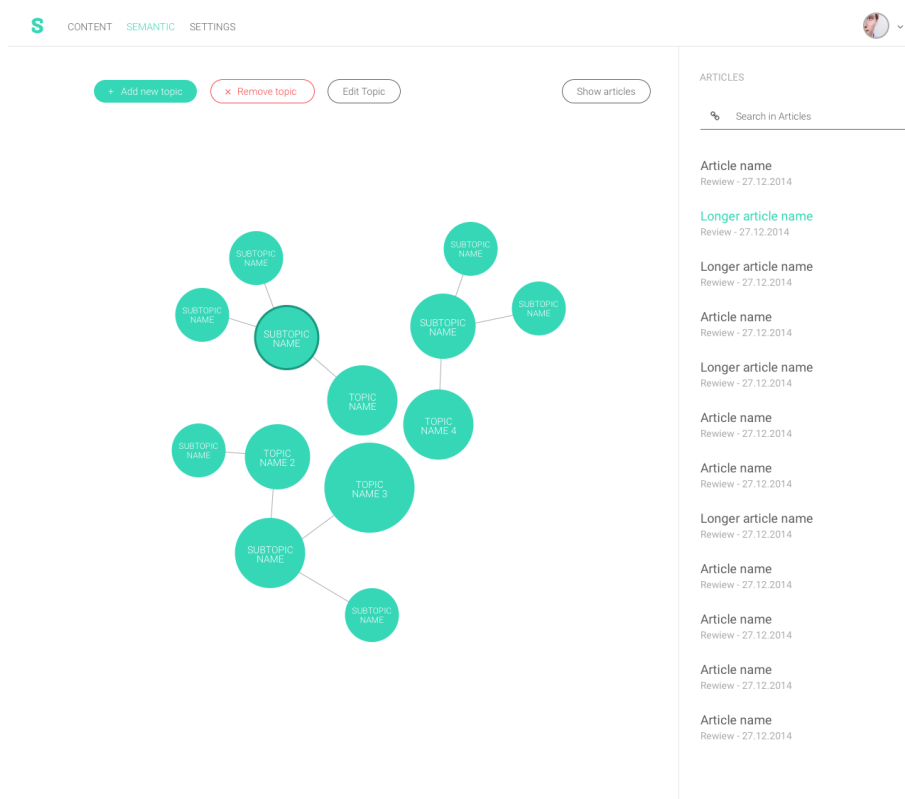
Proces návrhu zahrnoval hrubý návrh jednotlivých obrazovek na papír (například editor sémantické mapy obrázek 2). Následovaly první koncepty v grafickém editoru (obrázek 3). Poslední fází návrhu byl finální grafický návrh, který mohl být převeden do funkčního řešení (obrázek 4).



Obrázek 2: Sémantická mapa – První koncept na papír



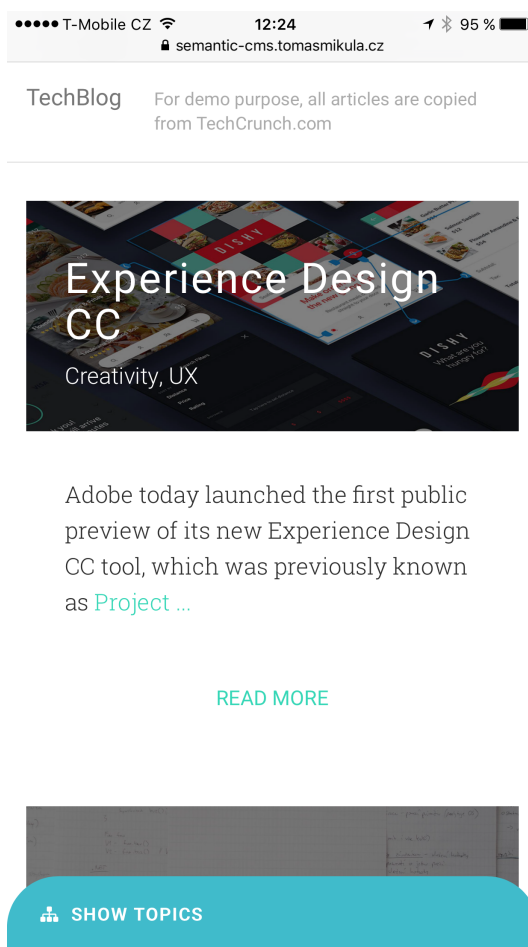
Obrázek 3: Sémantická mapa – První grafický koncept



Obrázek 4: Sémantická mapa – Finalní grafický návrh

Projekt je navržen v populárním flat design stylu [2] s důrazem na přehlednost aplikace. Napříč celou aplikací se vyskytují různé ikony, které zjednodušují komunikaci s uživatelem. Pro tyto účely jsem použil knihovnu Font Awesome popsanou v sekci 3.2.

Frontend projektu využívá responzivní designu [3]. Responsive design slouží k správnému zobrazení webu na různých velikostech zařízení (zobrazení na telefonu – obrázek 5). Tohoto efektu je docíleno za pomoci jednotného HTML s různými CSS styly. Pro jednodušší práci s responsive designem jsem použil Bootstrap framework (viz. sekce 3.2).



Obrázek 5: čtení článků na mobilním telefonu

## 3.2 Použité technologie

### Django web framework

Django [4] je moderní webový framework postavený na platformně Python. Mezi jeho hlavní přednosti patří rychlý vývoj a usnadnění stereotypních částí vývoje.

Již ve své standardní distribuci nabízí mnoho součástí, mezi které patří například autentizace uživatele, RSS a mnoho dalšího.

Velký důraz je v Django kladen na bezpečnost. Pomáhá vývojářům vyhnout se běžným bezpečnostním chybám jako je SQL injection, nebo cross-side scripting. Django je škálovatelné od nejmenších projektů až po ty největší.

Django framework používá architekturu Model-view-controller (MVC). Jednotlivé modely jsou reprezentované pomocí Python tříd s danými datovými typy. Samotný controller vytvoří potřebnou databázovou strukturu na libovolném databázovém backendu a zajistí rovněž veškerou komunikaci. Jednotlivé views se starají o prezentaci výsledných dat uživateli.

Projekt vytvořený pomocí Django Frameworku se skládá z jedné nebo více aplikací. Tato vlastnost umožňuje budovat univerzální aplikace, případně používat součásti z již existujících projektů.

Ve své práci používám nejnovější verzi frameworku Django 1.9 běžící na Python 3.4. Django framework však dále udržuje podporu rozšířené verze Python 2.7.

Django framework hodnotím velice kladně. Framework poskytuje dostatek prostředků pro tvorbu moderních webových aplikací. Při učení samotného frameworku jsem ocenil nadprůměrně dobře zpracovanou dokumentaci a aktivní podporu komunity.

## Django REST Framework

Knihovna [5] pro Django Framework sloužící k tvorbě webového API a ORM (Objektově relační mapování) serializace Django objektů. Mezi velké výhody patří API, které lze procházet skrze web. Koncový zákazník, pro kterého je web vyvíjen si tak může jednoduše výsledné API prohlédnout. Ke knihovně rovněž existuje kvalitně zpracovaná dokumentace.

## PostgreSQL

Semantic CMS využívá k uložení dat databázi PostgreSQL [6]. Je však důležité podotknout, že webová aplikace dokáže využívat libovolnou databázi podporovanou Django frameworkem (například populární MySQL). Volba tohoto řešení není pro výsledný projekt důležitá. PostgreSQL jsem vybral převážně z důvodu, že se jedná o open-source projekt.

## Bootstrap framework

Populární webový mobile-first HTML, CSS, JavaScript framework [7] pro tvorbu responzivního webu<sup>1</sup>. Největší výhoda spočívá v takzvaném „grid systému“. Obsah webové stránky je umístěn do kontejneru, který má vždy šířku 12ti sloupců

---

<sup>1</sup>Jeden HTML dokument se za pomoci CSS přizpůsobí velikosti různých zařízení.

(šířka sloupce závisí na velikosti obrazovky). Do takového kontejneru umísťujeme prvky, u kterých definujeme jejich velikost relativně k velikosti a počtu jednotlivých sloupců.

Bootstrap rovněž nabízí rozsáhlou knihovnu UI (User Interface) prvků. Tato knihovna UI prvků urychluje prototypování webových aplikací. V této práci jsem knihovnu nevyužil a UI prvky si navrhl sám.

## SASS

CSS preprocesor [8] sloužící k rozšíření možností klasického CSS. Přidává možnost použití proměnných, jednoduší zanořování stylů a rozdělení projektu na několik SCSS souborů, které se po výsledném zkompilování přeloží na jeden CSS soubor. Umožňuje definici takzvaných mixins. Mixin získá argumenty, které se následně předají do jeho těla a celé tělo se přeloží na výsledný CSS kód. Mezi další výhody patří možnost dědičnosti stylů z jiných definic a použití matematických operátorů. Preprocesor SASS mi umožnil psát přehledně strukturované CSS styly. Jeho nevýhodou je však určitá redundance výsledného CSS, která vzniká při překladu.

## jQuery

Dnes nejnámější JavaScriptová knihovna [9]. Nabízí velké množství funkcí, které ulehčí práci se samotným JavaScriptem. Za pomoci této knihovny jsem realizoval například modální okna, výběr data/času a mnoho dalšího.

## D3.js

JavaScriptová knihovna [10] pro vizualizaci dat. Knihovna využívá HTML, SVG a CSS. Tuto knihovnu jsem použil pro realizaci sémantické mapy a jejího editoru.

## Font Awesome

Ikonkový font a CSS Toolkit [11], který mi poskytl dostatečně velké množství ikon. Celý projekt je vydán pod licencí open-source.

## 3.3 Seznam aplikací v Semantic CMS

V této kapitole se budu věnovat jednotlivým aplikacím, z kterých se výsledný projekt v rámci Django Frameworku skládá. Každá aplikace řeší vlastní komunikaci s databází na základě poskytnutého datového modelu (Model) a definici pohledů (Views) pro zobrazování dat do výsledných šablon (Templates). Dělení na aplikace má význam pouze z hlediska přehlednosti a modulovatelnosti jednotlivých Django projektů.



## Article

Aplikace, která obsahuje veškerou funkcionalitu pro práci s obsahem (například views pro frontend, filtrování a další). Mezi nedůležitější součásti každého článku patří název, podtitulek, typ obsahu (viz. sekce 2.2), sémantické kategorie, klíčová slova a samotný obsah článku. Články bylo nutné rozdělit na publikované a rozepsané koncepty. Pro serializaci článků do formátu JSON je zde využit Django REST Framework. Aplikace rovněž obsahuje definice filtrů, které jsou použity při filtrování článků (viz. sekce 3.4.1).

## Semantic

Aplikace spravující samotnou sémantickou mapu. Skládá se ze dvou modelů, `Semantic` a `SemanticEdge`. `Semantic` udržuje informace o samotné sémantické kategorii, tedy o vrcholech grafu. `SemanticEdge` pak definuje samotné hrany grafu. Pro operace s orientovaným acyklickým grafem jsem použil volně šiřitelnou knihovnu `Django-Dag`. Tuto knihovnu jsem si musel pro své účely upravit. Ve výchozí verzi nebyla optimalizovaná pro Python verze 3.4. Upravenou verzi je možné najít na repozitáři [12].

Probíhá zde rovněž serializace celé sémantické mapy (viz. sekce 3.4.3) do formátu JSON (JavaScript Object Notation), která je nutná pro vizualizaci samotné mapy případně její editaci. O samotnou serializaci se stará Django REST Framework (viz. sekce 3.2).

## Keywords a Flags

Dvě menší aplikace starající se o klíčová slova a typy článků. U aplikace `Keywords` stojí za zmínku našeptávání klíčových slov realizovaných pomocí aplikace `Django Taggit`.

## Semantic Admin

Aplikace obsahující veškeré pohledy nutné pro správu samotného webu. Obsahuje rovněž definice jednotlivých formulářů na editaci a vytváření instancí objektů z aplikací popsaných výše. Samotné administrační části se budu věnovat v sekci 3.4.

## Aplikace třetích stran

Projekt obsahuje ostatní podpůrné aplikace třetích stran. Seznam používaných aplikací lze vidět v základním konfiguračním souboru `semantic_cms/semantic/settings/base.py` (výňatek ze souboru – zdrojový kód 1).

```

1     INSTALLED_APPS = (
2         ...
3         #3party
4         'django_dag',
5         'widget_tweaks',
6         'datetimewidget',      #Widget pro výběr data a času
7         'taggit',              #Implementace klíčových slov a našeptávání
8         'django_filters',      #Možnost filtrování objektů z~databáze
9         'rest_framework',
10        'redactor',            #WYSIWYG HTML editor
11        'imagekit',            #Práce s~obrázky
12        'betterforms',         #Širší možnosti komentářů
13        'password_reset',      #Reset uživatelského hesla
14        ...
15    )

```

Zdrojový kód 1: Výňatek ze souboru Base.py – Použité aplikace třetích stran

## 3.4 Podrobný popis – Administrační část

Administrační část (backend) poskytuje prostředky pro správu celé webové aplikace. V horní části administrační části se nachází navigační lišta. Mimo samotnou navigaci poskytuje rychlé zobrazení spravované webové stránky. Uživatel zde rovněž nalezne možnost odhlášení.

Většina ovládacích prvků obsahuje tooltip s doplňující nápovědou zobrazitelný po najetí myši. V následujících kapitolách se budu podrobněji věnovat všem významnějším součástím backendu.

### 3.4.1 Správce obsahu

Hlavní stránka administrace, slouží k rychlému přehledu a správě obsahu webové stránky. Umožňuje rovněž vytvořit nový článek. Již existující články jsou vypsané v přehledném seznamu. Jeho rozumná velikost je zaručena stránkováním po pěti článcích.

Samotný výpis článku lze filtrovat pomocí dostupného filtru. Mezi dostupné parametry patří article type (typ článku), article status (statut článku – publikovaný/koncept) a datum vytvoření. Mezi existujícími články lze rovněž vyhledávat (s ohledem na název článku, obsah a přiřazená klíčová slova).



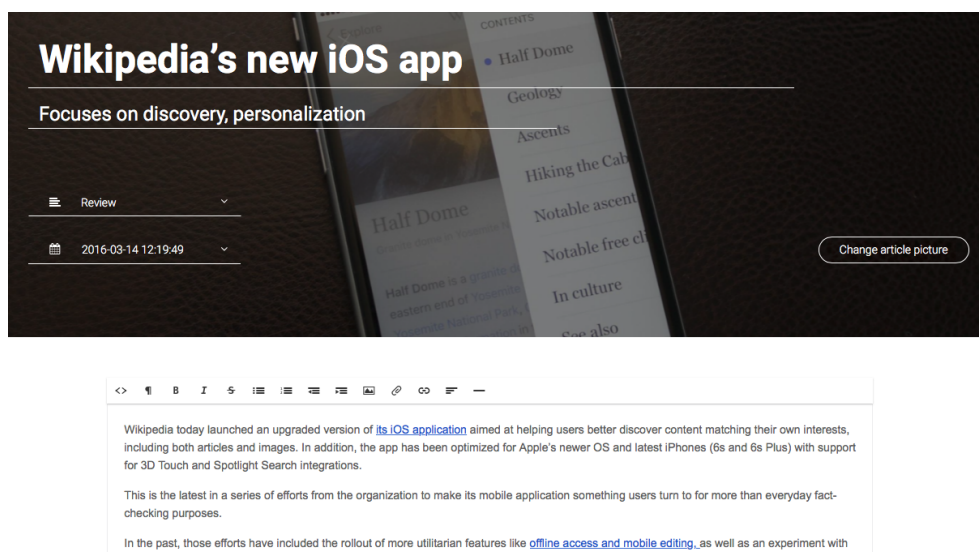
Obrázek 6: Content Manager – Rychlé akce v přehledu článků

U každého článku se nachází ovládací sekce (obrázek 6) s informací o počtu

sémantických kategorií přiřazených k článku a rychlými akcemi (správce sémantických kategorií pro daný článek, editace článku a smazání článku). V případě, že se jedná o koncept článku, je u něj zobrazena značka „Draft“.

### 3.4.2 Vytvoření/Editace článku

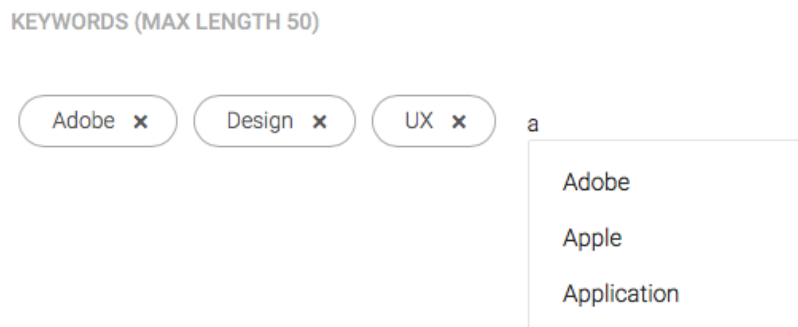
Stránka umožňující vytvoření nového článku. Poskytuje formulář na vyplnění všech dat popsanych v sekci 3.3 (obrázek 7). Autor zde může nahrát hlavní fotku článku, která se později využije v jeho záhlaví. Pro psaní obsahu článku je zde dostupný WYSIWYG HTML editor, který umožní snadné psaní i pro méně zkušeného uživatele.



Obrázek 7: Create new article – Základní záhlaví

Za zmínku stojí také sekce pro vkládání samotných klíčových slov. Při vkládání je poskytnuto našeptávání již existujících klíčových slov (obrázek 8). Našeptávání klíčových slov je realizované pomocí kontrolleru našeptávání klíčových slov (metoda GET) a jQuery UI knihovny, konkrétně jQuery UI autocomplete. SEO URL neboli přívětivý tvar URL adresy je automaticky generován z názvu článku. Pokud článek s touto SEO URL již existuje, autor musí návrh ručně editovat.

Po dokončení článku je možné přejít do editoru sémantické mapy a přiřadit k článku jednotlivé kategorie. V tomto případě se článek uloží jako koncept, článek je nutné později publikovat. Dále je možné článek pouze uložit jako koncept, rovnou publikovat, nebo ukončit psaní článku bez uložení změn.



Obrázek 8: Create new article – Našeptávání klíčových slov

### 3.4.3 Editor sémantické mapy

Stránka je rozdělena na dvě části. Levá (větší část) obsahuje editor sémantické mapy. Pravý sloupec umožňuje prohledávat a přiřazovat články do sémantických kategorií.

Sémantická mapa je realizována SVG grafikou, která je plně funkční i na mobilních zařízeních. Pro samotné vykreslování grafu jsem použil již zmíněný D3.js framework (viz. sekce 3.2). Komunikace mezi frontendem vytvořeném v D3.js a jQuery probíhá pomocí AJAX (Asynchronous JavaScript and XML). AJAX umožní za pomoci XMLHttpRequest aktualizovat sémantickou mapu bez nutnosti obnovení stránky. Při načítání editoru jQuery zažádá pomocí AJAX o data z API realizovaného pomocí Django REST Frameworku (viz. sekce 3.2).

Data jsou poskytována ze dvou adres. Adresa `/api/semantic_node/` slouží k získání serializovaných objektů reprezentujících vrcholy grafu. Adresa `/api/semantic_edge/` slouží k získání hran mezi vrcholy. Data jsou předána ve formátu JSON (viz. příklad 2).

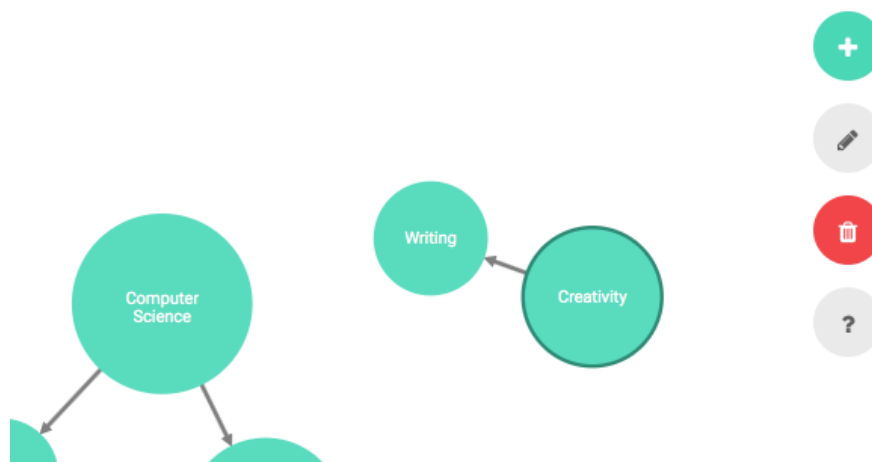
Po načtení dat dochází k jejich vykreslení. Každá změna v datech je okamžitě uložena na server. Pro tento účel se nachází na adrese `/admin/semantic/nazev_controlleru` sada kontrolerů komunikujících za pomoci metody POST sloužících k ukládání změn. Kontrolery se nacházejí za administrační autentizací a na rozdíl od API k nim nelze veřejně přistupovat. Popis jednotlivých kontrolerů naleznete v Tabulce 1.

Mezi funkce sémantického editoru patří přidání/odebrání/editace kategorií (vrchol grafu), přidání/odebrání vztahů mezi kategoriemi (hrana grafu), přiřazení/odebrání článku do/z kategorie. Podrobnější ovládání je popsáno v samotné nápovědě editoru.

Ovládání editoru (obrázek 9) je zajištěno pomocí kruhových tlačítek umístěných v pravém horním rohu editoru. Vždy jsou zobrazena pouze tlačítka, která mají v daném stavu editoru smysl, ostatní jsou skryta. Sémantickou mapu je možné přibližovat/oddalovat a libovolně posouvat. Pohyb v sémantické mapě je možný za pomoci myši.

Tabulka 1: Kontroléry pro editor sémantické mapy

Adresa kontroleru	Vyžaduje	Poskytuje
request_article_nodes/	Article ID	Serializované kategorie, které patří k danému článku.
save_article_nodes/	Article ID, Article Nodes (JSON)	Uloží k danému článku poskytnuté kategorie.
save_graph/	Nodes (JSON), Edges (JSON)	Uloží kompletní graf do systému.
add_edge/	Edge	Uloží přidanou hranu do systému.



Obrázek 9: Semantic editor – Ovládací prvky editoru

```

1  [
2    {
3      "id": 47,
4      "name": "Mobile devices",
5      "type": "semantic",
6      "number_of_descendants": 0,
7      "slug": "mobile-devices"
8    },
9    {
10     "id": 46,
11     "name": "Hardware",
12     "type": "semantic",
13     "number_of_descendants": 1,
14     "slug": "hardware"
15   },
16 ]

```

Zdrojový kód 2: Příklad – Vrcholy grafu po serializaci

Po výběru článku z pravého sloupce je možné článek přiřadit do označené kategorie. Pro větší přehlednost je možné články prohledávat.

### 3.4.4 Správce typů obsahu

Jednoduchá stránka sloužící k správě jednotlivých typů článku. Podobně jako Content Manager (sekce 3.4.1) používá seznam pro zobrazení všech dostupných typů článku a nabízí rychlé operace pro editaci a smazání. Součástí stránky je formulář pro vytvoření nového typu článku. Delká názvu typu článku je omezena na 25 znaků.

### 3.4.5 Nastavení

Stránka umožňující různé nastavení webové aplikace. Pro tento účel obsahuje několik záložek:

- **Blog** – nastavení samotného názvu blogu a jeho podtitul. Tato informace se zobrazuje v navigační liště frontendu.
- **Uživatelský profil** – editace informací o samotném autorovi. Konkrétně pak jméno, bio<sup>2</sup>, e-mail a profilový obrázek.
- **Změna hesla** – změna uživatelského hesla.
- **Reset databáze** – záložka sloužící pouze pro prezentační účely systému. Umožňuje resetovat veškerý obsah databáze do výchozí podoby.

<sup>2</sup>Krátký odstavec popisující autora článku.

### 3.4.6 Přihlášení/Zapomenuté heslo

Obrazovka sloužící pro přihlášení uživatele. Autentizační systém využívá autentizační aplikaci Django Frameworku, čímž je zaručena požadovaná bezpečnost. V případě zapomenutého hesla je zde dostupný odkaz na jeho obnovení.

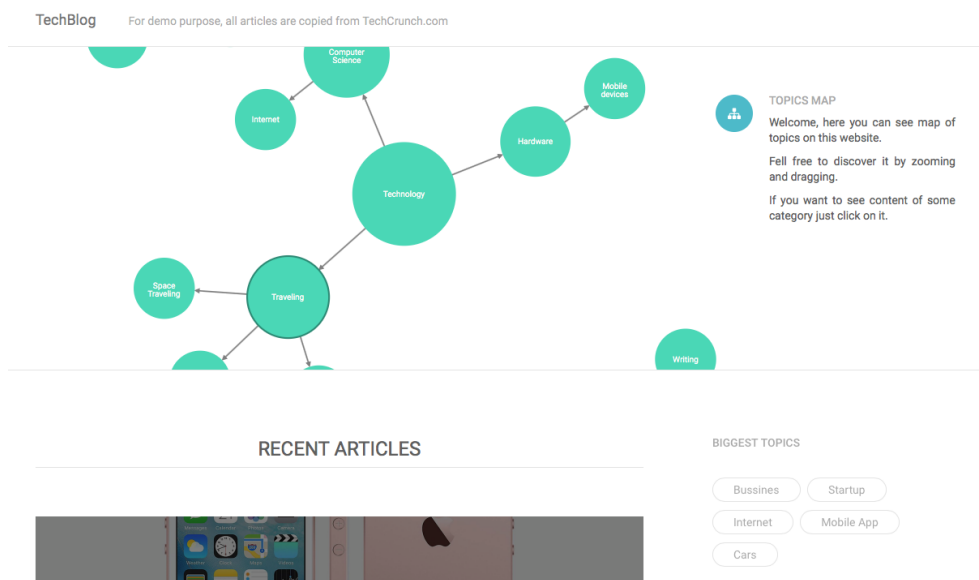
## 3.5 Podrobný popis – Část pro čtení článků

Část webové aplikace sloužící k interakci s čtenářem. Slouží pro prezentaci článků a procházení sémantické mapy z pohledu návštěvníka. Tato část aplikace je kompletně responzivní a plně funkční na mobilních zařízeních.

Všechny stránky obsahují horní panel s názvem blogu a podtitulem. V liště je ponecháno místo na případné odkazy či statické stránky (viz kapitola 4). Patička je rovněž na každé stránce stejná. Obsahuje copyright a odkaz na vstup do administrace.

### 3.5.1 Hlavní stránka

První stránka kterou uvidí nový návštěvník. Stránce dominuje náhled sémantické mapy s popisem (obrázek 10). Ačkoli se jedná o náhled, je sémantická mapa plně funkční. Vedle popisu je umístěno modré kolečko s ikonkou, které informuje nově přichozícího uživatele o tom, jaký symbol bude pro sémantickou mapu v budoucnu používán.



Obrázek 10: Hlavní stránka – Náhled sémantické mapy

Zbytek stránky je rozdělen na dva sloupce. Boční panel obsahuje největší

sémantické kategorie<sup>3</sup>, nejpoblárnější klíčová slova<sup>4</sup> a tlačítka na zobrazení plnohodnotné sémantické mapy přes celou stránku. Při posunu stránky se boční panel neposouvá a zůstává na svém místě (je fixní).

V levém sloupci jsou umístěny náhledy článků řazené od nejnovějšího. Náhled článku obsahuje:

- Úvodní fotku.
- Nadpis článku.
- Podnadpis článku.
- Datum publikace (pouze první článek).
- Sémantické kategorie do kterých článek spadá.
- Úryvek z článku.
- Odkaz na zobrazení celého článku „Read More“.

Ve spodní části stránky se nachází stránkování výpisu článku.

### 3.5.2 Detailní výpis článku

Univerzální stránka, která slouží k výpisu článků k danému sémantickému tématu, klíčovému slovu, nebo typu článku. Umožňuje čtenáři procházet pouze obsah z dané skupiny. Tato stránka je použita například při zobrazení konkrétní sémantické kategorie skrze sémantickou mapu.

Výpis náhledů článků je identický s výpisem náhledů na hlavní stránce. Detailní výpis již neobsahuje náhled sémantické mapy. Sémantickou mapu lze zobrazit pouze pomocí tlačítka pro zobrazení. V případě, že se čtenář nachází v detailním výpisu sémantické kategorie, je kategorie v sémantické mapě zvýrazněna.

### 3.5.3 Detail článku

Záhlaví stránky obsahuje název článku, podnadpis, datum publikace, sémantické kategorie a hlavní fotku článku. Na konci článku se nachází informace o autorovi (jméno, fotografie, biografický text – obrázek 11).

Pod informacemi o autorovi následuje výpis klíčových slov a sémantických kategorií k danému článku. Pod dodatečnými informacemi je pak zobrazena čtveřice doporučených článků (obrázek 12). Doporučovací algoritmus porovnává sémantické kategorie aktuálního článku se všemi články. Pro větší pokrytí témat jsou při porovnávání rovněž vzaty v úvahu kategorie umístěné o úroveň výše.

---

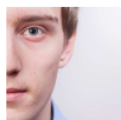
<sup>3</sup>Největší počet přiřazených článků

<sup>4</sup>Nejčastěji přiřazená klíčová slova ke článkům



## AUTHOR

---



### Tomáš Mikula

I'm a creative guy from the Czech Republic who loves Photography and Graphic design. I'm currently working as Stocksy.com photographer.

## KEYWORDS

---

Apple

iPhone

## ALL TOPICS

---

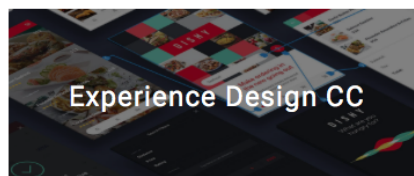
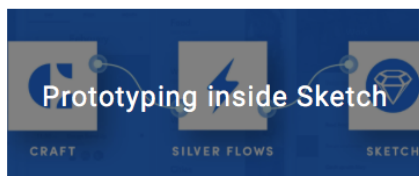
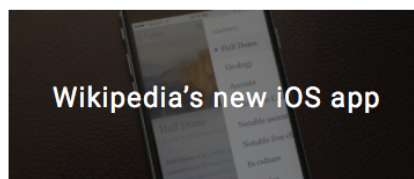
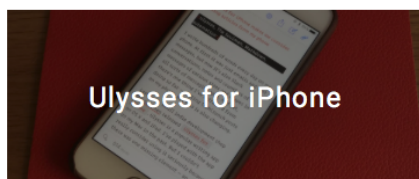
Mobile devices

Bussines

Obrázek 11: Detail článku – Informace o autorovi

## SIMILAR ARTICLES

---



Obrázek 12: Detail článku – Související články

## 3.6 Instalace

V následující kapitole budu popisovat instalaci Semantic CMS na produkční UNIS server s podporou HTTPS.

Aplikaci je rovněž možné spustit lokálně pro účel vývoje. `semantic_cms/semantic/settings/kofiguracni_soubor.py`. Verze pro vývoj používá jiné konfigurační soubory než verze produkční. Konfigurační soubory se nacházejí v adresáři `semantic_cms/semantic/settings/kofiguracni_soubor.py` (Ukázkový produkční konfigurační soubor – zdrojový kód 3).

Výhodou rozdělených konfiguračních souborů je možnost využít jiné databázové servery, případně jinou konfiguraci samotného Django Frameworku. Podrobný popis konfigurace je dostupný v oficiální dokumentaci Django Frameworku [14]. Konfigurace, které jsou totožné pro všechny verze se nacházejí v základním konfiguračním souboru `semantic_cms/semantic/settings/base.py` z kterého dědí ostatní konfigurační soubory.

Instalace webové aplikace na produkční server začíná naklonováním vzdáleného GIT repozitáře [13] případně přepíčováním zdrojového kódu z příloženého CD. Obecně je pro Python aplikace doporučeno vytvořit virtuální prostředí, které umožní používat více verzí Pythonu a knihoven na jednom serveru. Doporučuji tedy nainstalovat `virtualenv` a vytvořit nové virtuální prostředí s Python verze 3.4. Ve vytvořeném virtuálním prostředí je nutné nainstalovat správce python balíčků `pip`. Seznam potřebných python balíčků je uveden v souboru `semantic_cms/requirements.txt`. Tento seznam lze přímo předat správci balíčků `pip` pomocí příkazu:

```
pip install -r requirements.txt
```

Mezi další kroky patří nastavení databázového serveru (vytvoření databáze a přidání příslušné konfigurace do aplikace). Aplikace je testována na databázovém systému PostgreSQL. Po vytvoření databáze a nastavení konfiguračního souboru, je nutné připravit migraci databáze za pomoci příkazu:

```
django-admin makemigrations --settings=semantic_cms.settings.  
production
```

A následně provést samotnou migraci pomocí příkazu:

```
django-admin migrate --settings=semantic_cms.settings.production
```

Dále je nutné vytvořit WSGI soubor. Tento postup je podrobně popsán v oficiální dokumentaci Django [15] (ukázkový WSGI soubor – zdrojový kód 4).

Dále následuje konfigurace webového serveru, který bude Django aplikaci obsluhovat. Tuto funkci může zastoupit Apache server případně Nginx. Konfigurace pro server Apache je popsána v oficiální dokumentaci Django [16]

Posledním krokem je vytvoření superuživatele, který umožní přihlášení do administrace. Uživatele je možné vytvořit pomocí příkazu:

```
django-admin createsuperuser
```

### 3.7 Demo Semantic CMS

Pro jednodušší prohlídku aplikace jsem vytvořil veřejně dostupnou prezentaci aplikace běžící na adrese <https://semantic-cms.tomasmikula.cz>. Pro účely demonstrace jsem využil články ze serveru TechCrunch.com. Do administrace je možné přihlášení na adrese <https://semantic-cms.tomasmikula.cz/admin> za pomoci přihlašovacích údajů:

```
User: admin Password: 8V2kLMdm
```

Demo aplikace rovněž disponuje možností obnovení obsahu do původního stavu. Pro jednoduchost je funkce realizována voláním interních příkazů Django Frameworku pro správu databáze. Toto řešení není rozhodně vhodné pro finální nasazení. Vhodnější by bylo například využití softwarového deamona Cron.

## 4 Možnosti rozvíjení aplikace

Jako každý jiný software i v případě mé aplikace se nabízí mnoho možných vylepšení. Těmto vylepšením se budu věnovat v následující kapitole.

### Propojení více webů pomocí jedné sémantické mapy

Mezi nejzajímavější rozšíření základního konceptu patří zdokonalení API a umožnění propojení několika webů za pomoci globální sémantické mapy. Tato koncepce by mohla být zajímavá v případě aplikování na podobný produkt jako je například webová služba Medium [17]. Medium umožňuje komukoli napsat libovolný článek. Autor nevlastní celý web, ale pouze svůj profil, pod kterým články publikuje.

Při aplikaci takových rozměrů by muselo být editování sémantické mapy značně ošetřeno. V ideálním případě by se globální mapa dělila na velké množství menších map. Takový systém by se však pravděpodobně neobešel bez správy ze strany provozovatele, případně komunity.

Jedno z možných řešení je použít systém pro verzování zdrojového kódu (např. GIT), který by umožnil uživatelům vytvářet dočasnou kopii aktuální globální mapy (fork). Po provedení úprav by uživatel zažádal o schválení jeho úprav. Tento přístup by ze začátku projektu vyžadoval velkou režii. S postupem času by však bylo nutné globální mapu upravovat méně a méně. Použití globální sémantické mapy (nebo její částí) napříč několika weby, umožní vidět ještě větší souvislosti mezi články.

Jednodušší verze tohoto konceptu pracuje pouze s myšlenkou umožnit za pomoci API naimportovat sémantickou mapu (nebo její část) z jiného webu a její změny nezaznamenávat zpět.

### Multiuživatelský webový portál

S možnostmi, které jsem popsal v předchozím konceptu přichází nutnost multiuživatelského prostředí. Celý systém by mohl fungovat jako webové aplikace, kde by se jednotliví autoři zaregistrovali a spravovali svoje vlastní články. Nabízí se rovněž možnost spolupráce a sdílení přístupu k editaci článku (např. korektura).

Django Framework nabízí velkou škálovatelnost. Takový systém by mohl být postaven na stejné technologii jako stávající projekt.

### Další využití uložených dat

Možnosti v případě využití samotné sémantické mapy taky nekončí. Velká sémantická mapa obsahuje mnoho informací, které je možno využít pro získání uživatelsky užitečných informací.

Jedná se například o generování rychlého přehledu nejdůležitějších témat na webu, případně generování tematických týdenních newsletterů pro čtenáře.

S tímto vylepšením přichází i možnost sběru většího množství dat, jako například počet přečtení a počet komentářů. Za pomoci těchto dat by bylo možné mapovat vývoj samotného blogu, o jakých tématech se psalo nejčastěji a kdy. Tyto informace mohou být užitečné pro autora webu.

Mezi další vylepšení by mohl patřit komplexnější algoritmus na doporučování souvisejících článků. Algoritmus by mohl graf prohledávat do větší hloubky pro nalazení většího množství článků. Tyto články by se poté mohly dále hodnotit na základě klíčových slov případně počtu přečtení/komentářů.

## Vizualizace sémantické mapy

Mezi další možná vylepšení patří rozšíření vizualizace sémantické mapy. Aktuální sémantická mapa prezentuje pouze základní informace o vztahu jednotlivých témat. Mluvíme zde například o zobrazení podrobnějších informací typu počet článků v dané kategorii, nebo možností přepnutí vizualizace. Velikost tématu v sémantické mapě aktuálně reprezentuje pouze počet podtémat. V případě možnosti přepínání by mohla reprezentovat rovněž počet článků, počet přečtení, nebo počet komentářů. Dále by bylo možné zavést typ a sílu vazby mezi kategoriemi. Například zda se jedná o rozšíření kategorie nebo její doplnění a jaká je síla samotné vazby (relevance).

## Vylepšený editor sémantické mapy

Editor sémantické mapy by bylo možné lépe optimalizovat. Jedná se především o používání editoru na dotykových zařízeních (ke spojení dvou témat je nutné použít klávesu Shift). Rovněž by bylo možné rozdělovat sémantickou mapu na menší pod-mapy a pracovat vždy jen s částí celkové mapy. Export a import částí sémantické mapy naznačený v konceptu 4 by zde mohl být použit jako dobrý nástroj pro migraci samotné mapy na jiné weby.

## Statické stránky

Statické stránky můžeme považovat za speciální případ obsahu. Do projektu jsem je nezahrnul, protože nejsou nutné pro sémantiku obsahu. V praktickém nasazení se však jedná o užitečnou věc. Častým využitím statických stránek je například stránka s kontaktními údaji.

## Sémantické filtrování článků

V budoucnu bych rovněž rád pro čtenáře implementoval možnost filtrování obsahu dle sémantických atributů. Jedná se o vyhledávání v člancích a filtrování výsledků podle výskytu klíčových slov, typu obsahu, příslušnosti do sémantické kategorie, stáří, oblíbenosti a jiných.

```

1  from .base import *
2
3  DEBUG = False
4
5  ALLOWED_HOSTS = ['semantic-cms.tomasmikula.cz']
6
7  BASE_DIR = os.path.dirname(os.path.dirname(os.path.dirname(os.path
      .abspath(__file__))))
8
9  STATIC_URL = '/static/'
10 STATIC_ROOT = os.path.join(BASE_DIR, 'static/')
11
12 MEDIA_URL = '/media/'
13 MEDIA_ROOT = os.path.join(BASE_DIR, 'media/')
14
15 STATICFILES_DIRS = []
16
17 CSRF_COOKIE_SECURE = True
18 SESSION_COOKIE_SECURE = True
19 SECURE_SSL_REDIRECT = True
20 SECURE_BROWSER_XSS_FILTER = True
21 SECURE_CONTENT_TYPE_NOSNIFF = True
22 SECURE_HSTS_SECONDS = 3600
23 SECURE_HSTS_INCLUDE_SUBDOMAINS = True
24 X_FRAME_OPTIONS = 'DENY'
25
26 DATABASES = {
27     'default': {
28         'ENGINE': 'django.db.backends.postgresql_psycopg2',
29         'NAME': 'semanticdb',
30         'USER': 'semantic',
31         'PASSWORD': '*****',
32         'HOST': 'localhost',
33         'PORT': '',
34     }
35 }

```

Zdrojový kód 3: Ukázkový produkční konfigurační soubor

```

1 """
2 WSGI config for semantic_cms project.
3
4 It exposes the WSGI callable as a module-level variable named ``
   application``.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/1.8/howto/deployment/wsgi/
8 """
9
10 import os, sys
11
12 from django.core.wsgi import get_wsgi_application
13
14 apache_configuration= os.path.dirname(__file__)
15 project = os.path.dirname(apache_configuration)
16 workspace = os.path.dirname(project)
17 sys.path.append(workspace)
18 sys.path.append(project)
19
20 sys.path.append('/home/tom')
21
22 os.environ.setdefault("DJANGO_SETTINGS_MODULE", "semantic_cms.
   settings.production")
23
24 application = get_wsgi_application()

```

Zdrojový kód 4: Ukázkový produkční konfigurační soubor

## Závěr

Výsledná webová aplikace Semantic CMS obsahuje základní nástroje pro tvorbu a používání sémantické navigace, prezentované uživatelsky přívětivou formou. S výslednou podobou aplikace jsem velice spokojen.

Prostředí pro čtení článků poskytuje návštěvníkovi nové možnosti navigace a zároveň zachovává již známé prvky. Jedná se tak o malý evoluční krok, který by mohl vést k nové podobě prezentace obsahu na internetu.

Podobného výsledku se mi povedlo dosáhnout v administrační části aplikace. Administrace si zachovává již známé a zaběhlé postupy správy webu. Editor sémantické mapy poskytuje příjemné prostředí pro správu sémantických kategorií.



## Conclusions

The resulting web application Semantic CMS contains basic tools for creating and using semantic navigation in a user friendly interface. I am very satisfied with the final version.

The environment for reading of the articles provides the visitor new options for navigating while retaining already known features. It is considered as a small step in the evolution that could lead to a new form of the content presentation on the internet.

I was able to achieve a similar result in the administration part of the application as well. Administration maintains very well known and frequently used methods of the web administration. Semantic map editor provides comfortable interface for the management of the semantic categories.

## A Obsah přiloženého CD/DVD

### **src/**

Kompletní zdrojový kód webové aplikace SEMANTIC CMS se všemi potřebnými (příp. převzatými) zdrojovými kódy, pro bezproblémové zkopírování na webový server. Pro testovací účely doporučuji využít demo (viz. kapitola 3.7).

### **lib/**

Všechny potřebné knihovny pro offline instalaci pomocí Python balíčkovací pip. Pokud to je možné, doporučuji instalaci z online zdrojů (viz. kapitola 3.6)

### **doc/**

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

U veškerých cizích převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na CD/DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce.

## Literatura

- [1] SEDGEWICK, Robert; WAYNE, Kevin Algorithms (4th Edition). Princeton: Addison-Wesley Professional [online]. 2002—2014. [cit. 2014-04-23]. Dostupné z: <http://algs4.cs.princeton.edu/home/>
- [2] CLUM, Luke: A Look at Flat Design and Why It's Significant [online]. 2013-05-13. [cit. 2016-04-30]. Dostupné z: <http://uxmag.com/articles/a-look-at-flat-design-and-why-its-significant>
- [3] W3school.com: HTML Responsive Web Design [online]. [cit. 2016-04-30]. Dostupné z: [http://www.w3schools.com/html/html\\_responsive.asp](http://www.w3schools.com/html/html_responsive.asp)
- [4] Django Software Foundation: Django Framework [online]. 2005–2016. [cit. 2016-03-18]. Dostupné z: <https://www.djangoproject.com>
- [5] CHRISTIE, Tom: Django REST Framework [online]. 2011-2016. [cit. 2016-03-20]. Dostupné z: <http://www.django-rest-framework.org>
- [6] The PostgreSQL Global Development Group: PostgreSQL [online]. 1996-2016. [cit. 2016-03-20]. Dostupné z: <http://www.postgresql.org>
- [7] Twitter: Bootstrap 3 [online]. 2011-2016. [cit. 2016-03-20]. Dostupné z: <http://getbootstrap.com>
- [8] CATLIN Hampton, WEIZENBAUM Natalie, EPPSTEIN Chris: SASS language [online]. 2006-2016. [cit. 2016-03-21]. Dostupné z: <http://sass-lang.com>
- [9] The jQuery Foundation: jQuery [online]. 2005-2016. [cit. 2016-03-21]. Dostupné z: <https://jquery.com>
- [10] BOSTOCK Mike: D3.js [online]. 2011-2016. [cit. 2016-03-21]. Dostupné z: <https://d3js.org>
- [11] GANDY Dave: Font Awesome [online]. [cit. 2016-03-30]. Dostupné z: <https://fontawesome.github.io/Font-Awesome/>
- [12] PASOTTI Alessandro, MIKULA Tomáš: Django-Dag. GitHub.com [online]. Dostupné z: <https://github.com/wilima/django-dag>
- [13] MIKULA Tomáš: Semantic CMS. GitHub.com [online]. Dostupné z: <https://github.com/wilima/Semantic-CMS>
- [14] Django Software Foundation: Django Settings. Django Documentation [online]. 2005–2016. [cit. 2016-04-23]. Dostupné z: <https://docs.djangoproject.com/en/1.9/ref/settings/>
- [15] Django Software Foundation: How to deploy with WSGI. Django Documentation [online]. 2005–2016. [cit. 2016-04-23]. Dostupné z: <https://docs.djangoproject.com/ja/1.9/howto/deployment/wsgi/>

- [16] Django Software Foundation: How to use Django with Apache and mod\_wsgi. Django Documentation [online]. 2005–2016. [cit. 2016-04-23]. Dostupné z: <https://docs.djangoproject.com/ja/1.9/howto/deployment/wsgi/modwsgi/>
- [17] WILLIAMS Ev: Medium.com [online]. Dostupné z: <https://medium.com>