

Návrh aplikace pro webové stránky měst

Patrik Valuch

Bakalářská práce
2019



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Patrik Valuch**
Osobní číslo: **A15249**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Softwarové inženýrství**
Forma studia: **prezenční**

Téma práce: **Návrh aplikace pro webové stránky měst**
Téma anglicky: **An Application Design for Cities' Websites**

Zásady pro vypracování:

1. Provedte rešerši existujících řešení.
2. Vypracujte stručný rozbor technologií, které budou použity k návrhu.
3. Provedte rozbor a analýzu požadavků na zvolené řešení.
4. Realizujte navrženou aplikaci.
5. Navrženou aplikaci vhodným způsobem popište.
6. Věnujte pozornost zabezpečení.



Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **ASP.net MVC with entity framework and CSS.** New York, NY: Springer Science+Business Media, 2016. ISBN 978-1484221365
2. **GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL. HTML5 a CSS3 pro webové designéry.** Brno: Zoner Press, 2011. Encyklopedie webdesignera. ISBN 978-80-7413-166-0.
3. **DUCKETT, Jon, Gilles RUPPERT a Jack MOORE. JavaScript & jQuery: interactive front-end web development.** Indianapolis, IN: Wiley, 2014. ISBN 978118531648.
4. **LAMBERT, Matt. Learning Bootstrap 4. 2nd Revised edition.** Birmingham: Packt Publishing, 2016. ISBN 978-1785881008.
5. **AQUINO, Chris a Todd GANDEE. Front-end web development: the Big Nerd Ranch guide.** Atlanta, GA: Big Nerd Ranch, 2016. ISBN 978-0134433943.

Vedoucí bakalářské práce:

Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

3. prosince 2018

Termín odevzdání bakalářské práce:

15. května 2019

Ve Zlíně dne 7. prosince 2018

doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Mgr. Roman Jašek, Ph.D.
garant oboru

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 10.5.2019

Patrik Valuch, v. r.

ABSTRAKT

Cílem této práce je návrh a následná realizace aplikace pro webové stránky měst a obcí.

Aplikace bude sloužit pro sledování a vytváření aktualit, akcí, poptávek, nabídek a diskuzí o problémech mezi občany. Cílem je začlenění klasického registrovaného uživatele do tvorby obsahu webu. Teoretická část práce bude obsahovat popis technologií použitých pro návrh a následnou realizace práce. Tato část bude také obsahovat rešerši existujících řešení portálů pro obce a města.

Praktická část se bude zabývat analýzou požadavků pro aplikaci a její následnou vhodnou realizací. Aplikace bude realizována pomocí technologií .NET MVC, HTML, CSS, SQL.

Klíčová slova: Webová aplikace, portál pro obce a města, .NET MVC, C#

ABSTRACT

The aim of this thesis is to design and implement the application for websites of towns and municipalities. The application will be used for monitoring and creating news, events, inquiries, offers and discussions about issues among citizens. The aim is to integrate a classic registered user into the creation of a web content.

The theoretical part of the thesis includes a description of technologies that are used for the design and subsequent implementation of the work. This part also contains recherche about already existing solutions for cities and towns.

The analytical part is followed by the analysis of requirements for application and its subsequent appropriate implementation. The application will be implemented using .NET MVC, HTML, CSS, and SQL technologies.

Keywords: Web application, portal for towns and cities, .NET MVC, C#

Chtěl bych poděkovat Ing. Petru Šilhavému, Ph.D. za odborné vedení mé bakalářské práce, odborný dohled a cenné rady, které mi pomohly tuto práci dokončit.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 CONTENT MANAGEMENT SYSTÉMY	11
1.1 OPEN-SOURCE	11
1.1.1 WORDPRESS.....	11
1.1.2 JOOMLA.....	13
1.1.3 DRUPAL.....	15
1.2 KOMERČNÍ ŘEŠENÍ.....	17
1.2.1 VISMO.....	17
1.2.2 GALILEO CONTENT MANAGER.....	19
2 MOŽNOSTI PŘI REALIZACI WEBOVÉ APLIKACE	22
2.1 PHP.....	22
2.2 ARCHITEKTURA MVC.....	22
3 POUŽITÉ TECHNOLOGIE	23
3.1 UML.....	23
3.2 ASP.NET MVC.....	24
3.2.1 MODEL.....	24
3.2.2 VIEW.....	25
3.2.3 CONTROLLER.....	25
3.3 HTML.....	25
3.4 CSS.....	26
3.5 SQL.....	28
3.6 ADO.NET ENTITY FRAMEWORK	30
3.7 JAVASCRIPT	31
4 RIZIKA WEBOVÉ APLIKACE	33
4.1 INJECTION	33
4.2 BROKEN AUTHENTICATION AND SESSION MANAGEMENT.....	33
4.3 CROSS-SITE SCRIPTING	34
4.4 INSECURE DIRECT OBJECT REFERENCES.....	35
4.5 SECURITY MISCONFIGURATION.....	35
4.6 SENSITIVE DATA EXPOSURE.....	36
4.7 MISSING FUNCTION LEVEL ACCESS CONTROL.....	37
4.8 CROSS-SITE REQUEST FORGERY	37
4.9 USING COMPONENTS WITH KNOWN VULNERABILITIES.....	38
4.10 UNVALIDATED REDIRECTS AND FORWARDS	39
II PRAKTICKÁ ČÁST	40
5 WEBOVÁ APLIKACE PRO STRÁNKY MĚST	41
5.1 ANALÝZA POŽADAVKŮ WEBOVÉ APLIKACE	41
5.1.1 FUNKČNÍ POŽADAVKY	42
5.1.2 NEFUNKČNÍ POŽADAVKY.....	44
5.1.3 USE CASE MODEL.....	45

5.2	NÁVRH DATABÁZE	57
6	UŽIVATELSKÁ PŘÍRUČKA	59
6.1	Z POHLEDU NEREGISTROVANÉHO UŽIVATELE	60
6.1.1	REGISTRACE	60
6.2	Z POHLEDU REGISTROVANÉHO UŽIVATELE.....	61
6.2.1	PŘIHLÁŠENÍ DO APLIKACE	61
6.2.2	ZMĚNA HESLA	62
6.2.3	ODHLÁŠENÍ Z APLIKACE	63
6.2.4	VYTVOŘENÍ AKTUALITY A PŘÍSPĚVKU V DISKUZI.....	64
6.2.5	VYTVOŘENÍ UDÁLOSTI	65
6.2.6	VYTVOŘENÍ NABÍDKY/POPTÁVKY	66
6.2.7	VYTVOŘENÍ KOMENTÁŘE	67
6.2.8	EDITACE PŘÍSPĚVKU	68
6.2.9	SMAZÁNÍ A NAHLÁŠENÍ PŘÍSPĚVKU	69
6.2.10	FILTROVÁNÍ PŘÍSPĚVKŮ	70
6.2.11	VYTVOŘENÍ HODNOCENÍ AKCE	71
6.2.12	NOTIFIKACE	72
6.3	Z POHLEDU ADMINISTRÁTORA	73
6.3.1	SCHVALOVÁNÍ PŘÍSPĚVKŮ	73
6.3.2	NOTIFIKACE	74
7	METODY ZABEZPEČENÍ	75
8	MOŽNÝ ROZVOJ APLIKACE	77
	ZÁVĚR	78
	SEZNAM POUŽITÉ LITERATURY	79
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	84
	SEZNAM OBRÁZKŮ	85
	SEZNAM TABULEK	88
	SEZNAM PŘÍLOH	89

ÚVOD

Každé město nebo obec má v dnešní době webové stránky. Jejich kvalita zpracování se jak po stránce vizuální, tak funkční hodně liší.

Většina webových stránek obcí a měst plní hlavně informativní funkci a málokterá webová stránka začlení běžného uživatele do tvorby jejího obsahu. Stránky umožňují zasílání aktualit a dalších informací pomocí mailu, ale neumožňují se k nim veřejně vyjadřovat. Stejně tak řešení různých problémů či nedostatků probíhá anonymní formou pomocí odeslání na mail obce či města a neumožňuje veřejnou diskuzi občanů na dané téma.

Důvodem zvolení bylo řešení problému ve vlastní obci a nemožnost vyjádření svého názoru na danou problematiku ze strany všech občanů obce.

Hlavní funkcí aplikace bude přidávání obsahu ze strany běžného uživatele na webových stránkách obce. Neregistrovaný uživatel se bude moci po stránce neomezeně pohybovat a číst si již vytvořené příspěvky, ovšem nebude moci příspěvky vytvářet nebo na ně reagovat. Registrovaný uživatel bude moci vytvářet příspěvky v několika sekcích a bude mít možnost na již vytvořené příspěvky reagovat pomocí komentářů. Zveřejnění příspěvků ve všech sekcích bude podléhat schválení administrátora.

K dokumentaci návrhu aplikace bude použit jazyk UML. Následná realizace bude pomocí technologie ASP.NET MVC s využitím jazyka C#. Vizuální reprezentace webové stránky bude pomocí jazyka HTML a CSS. Pro práci s databází bude využito jazyka SQL.

I. TEORETICKÁ ČÁST

1 CONTENT MANAGEMENT SYSTÉMY

Každá obec nebo město má své webové stránky a k jejich správě využívá redakční systém neboli content management systém (CMS). Po projití několika webových stránek obcí a měst jsem došel k názoru, že si obce a města nevybírají přímo CMS systém, ale ten je jim dodán jako součást kompletní realizace jejich webových stránek od zhotovitele. V této části jsem vybral několik zástupců – jak z řad open-source, tak z řad komerčních řešení.

1.1 OPEN-SOURCE

Velká část používaných CMS jsou open-source projekty. Do této kategorie se řadí systémy jako WordPress, Joomla a Drupal. Tyto systémy jsou na předních příčkách nejlepších CMS systémů. [1] Systémy jsou taktéž uvedeny na oficiálních internetových stránkách Ministerstva vnitra České Republiky jako nejpoužívanější redakční systémy a nejlepší redakční systémy v poměru cena/výkon v sekci „Jak na obecní web?“ [2].

1.1.1 WORDPRESS

WordPress je open-source systém pro správu obsahu. Díky kompletnímu sepsání zdrojového kódu v dokumentaci je umožněno upravovat si jeho součásti podle svých představ a tím se podílet na rozvoji WordPressu jako takového. Tím se na vývoji podílí široká veřejnost, zatímco u komerčních produktů pouze pár jedinců [3].

Systém pracuje na operačním systému podporujícím hypertext preprocessor (PHP) a databázi MySQL. Software je založen na programovacím jazyce PHP a databázích MySQL [4].

Obrovskou výhodou WordPressu je aktivní komunita. Díky aktivní komunitě není problém najít návody, dokumentaci nebo informace téměř o každém aspektu vývoje ve WordPressu [5]. S desítkami miliónů stránek a s nejvíce aktivní vývojářskou a komunitní podporou lze WordPress označit za jeden z nejúspěšnějších projektů v moderní software historii [1].

Výhody

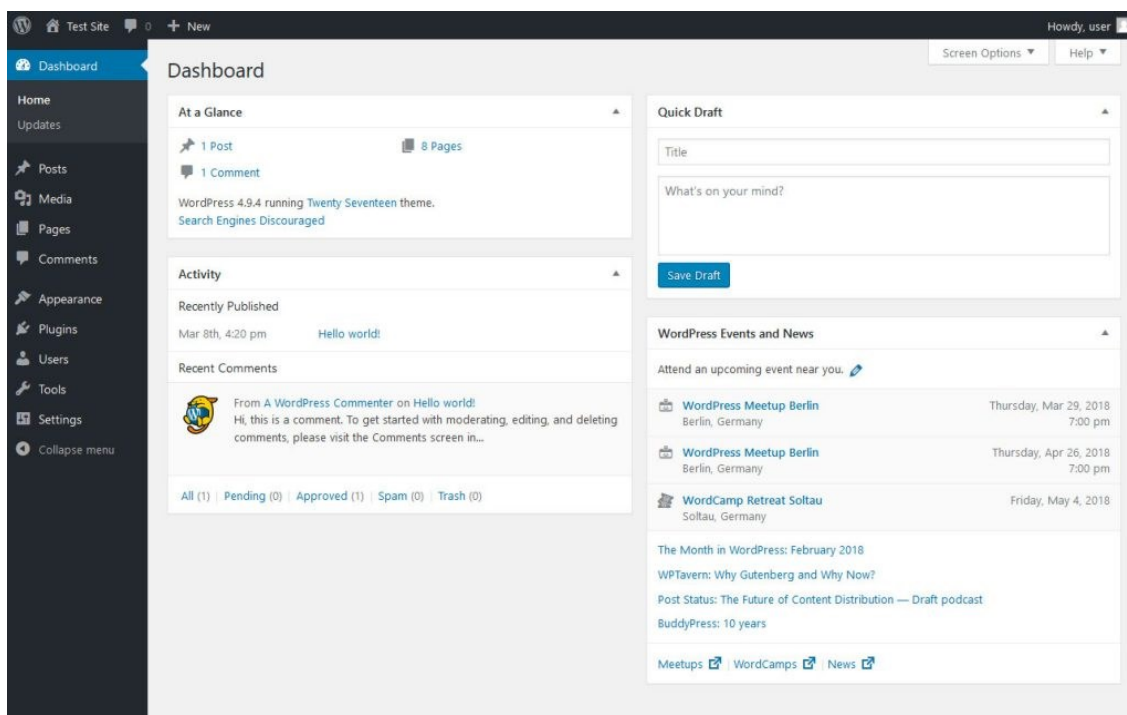
WordPress je uživatelsky velice přívětivý a dovolí i začátečníkům začít velice rychle, bez znalosti programovacích jazyků či programování obecně [6]. Platforma je variabilní a

obsahuje tisíce vytvořených šablon s možností tvorby vlastních. Široká škála placených, neplacených pluginů a specializovaných témat umožňuje tvorbu různých typů webů [5]. Další nespornou výhodou je snadná instalace.

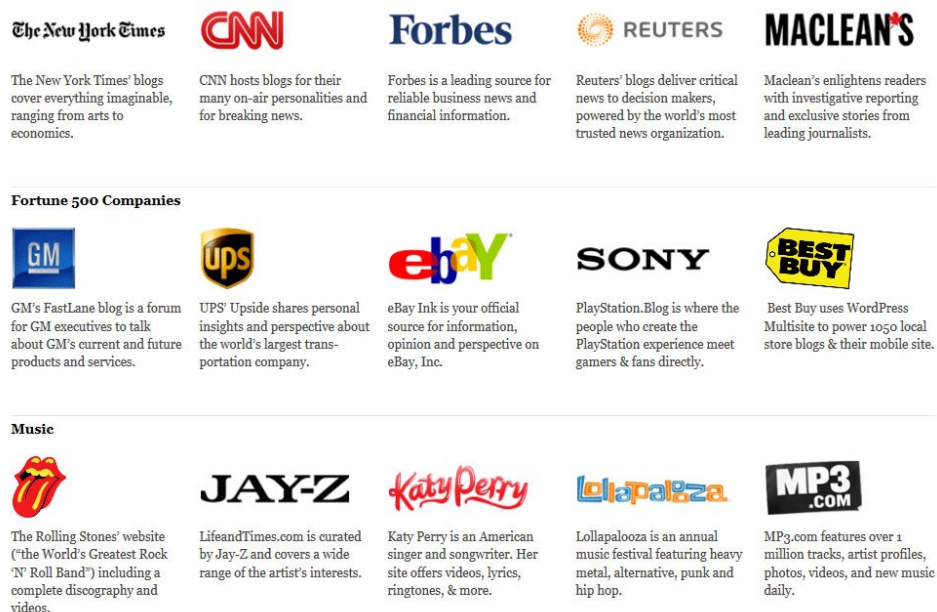
Nevýhody

Díky velkému zájmu o WordPress a velkému množství instalací přitahuje pozornost lidí zabývajících se hledáním mezer v bezpečnosti [7]. Uživatel musí udržovat systém v aktuální verzi včetně všech šablon a pluginů [8]. Uživatel by měl být proto obezřetný už při výběru daných pluginů a šablon a nešetřit tak na bezpečnosti.

Základní práce s WordPressem nevyžaduje znalost programovacích jazyků, v případě volby vlastních úprav nebo vylepšení je nutnosti doplnit si znalosti o základy jazyků *Hypertext Markup Language* (HTML) a *Cascading Style Sheets* (CSS) [8].



Obrázek 1: CMS WordPress [6]



Obrázek 2: Příklady velkých webů využívajících WordPress [6]

1.1.2 JOOMLA

Joomla je další z řady open-source systémů pro správu obsahu. Systém je napsán v jazyce PHP, využívající MySQL databáze pro ukládání dat [18].

Využívá techniky objektově orientovaného jazyka. Jedná se o jeden z nejvíce populárních systémů pro správu obsahu využívající funkce jako page caching či vícejazyčnou podporu s možností rozšíření systému pomocí pluginů a dalších rozšíření [18].

Existuje mnoho důvodů, proč si zvolit právě systém Joomla. Jedná se o open-source systém se znovupoužitelným zdrojovým kódem odpovídajícím požadavkům. S velkým množstvím možností a funkcí může být systém Joomla nasazen na mnoho typů webových stránek [18]. Za jeho úspěchem stojí především jeho rozšiřitelnost a především spousta vývojářů, kteří přinášejí stále nová rozšíření [19].

Výhody

Jednou z výhod je možnost uživatelské autorizace pomocí Lightweight Directory Access Protocol (LDAP), OpenID a Google [5].

Ovládací panel může začátečníky zaskočit svým množstvím funkcí, a tudíž i odradit, avšak na základě následného pochopení struktury a funkcí pak uživatel může plně využít všechny možnosti systému [20].

Za výhodu lze považovat také správu navigace. Joomla obsahuje souhrnný navigační systém, který zvládá obsluhu několika hierarchií [20].

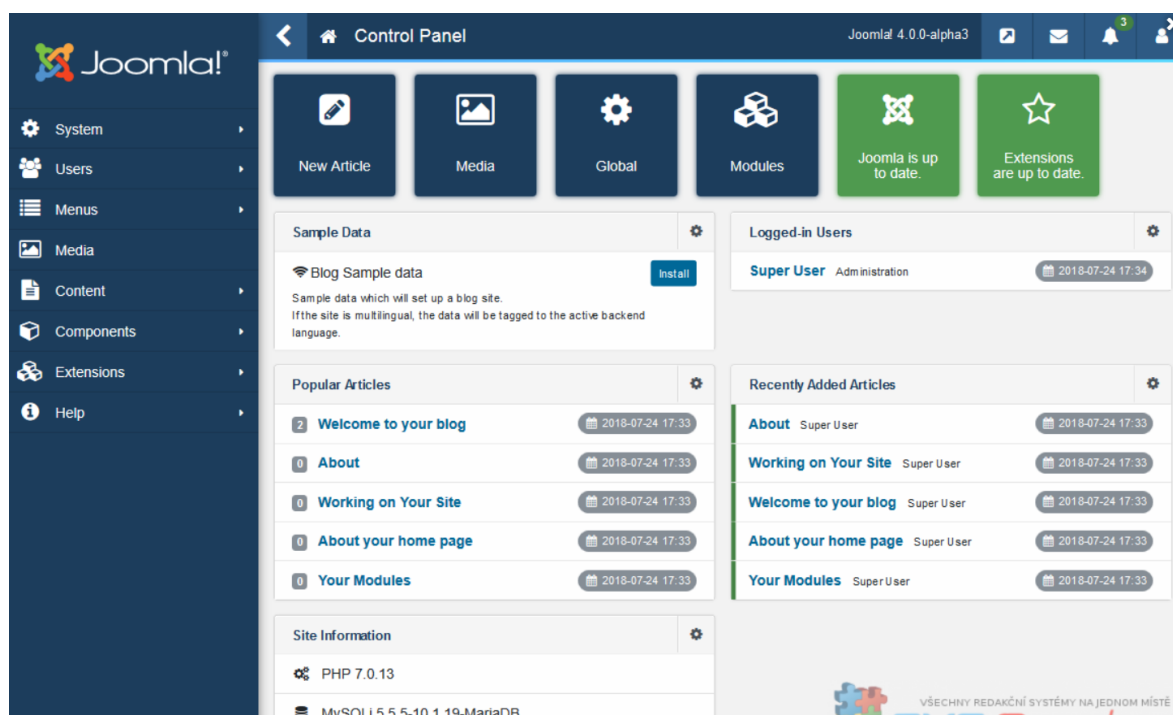
Stejně jako u WordPressu, Joomla disponuje rozsáhlou technickou podporou. Jsou k dispozici velká, aktivní fóra v anglickém jazyce, ale najdeme i uživatelská fóra psaná v češtině. [20]

Nevýhody

Jak již bylo zmíněno v části výhod, uživatelské rozhraní není tak intuitivní jako v případě jiných CMS systémů, ale je stále funkční [5].

Nevýhodou tohoto systému je, že ve srovnání s ostatními systémy (například s WordPressem) Joomla neobsahuje tolik kvalitně zpracovaných témat [5].

Dále některé moduly a pluginy mívají problémy se vzájemnou kompatibilitou. V některých případech se uživatel nevyhne úpravě PHP kódu pro zprovoznění plné funkcionality [20].



Obrázek 3: CMS Joomla 4.0 [21]



Obrázek 4: Stránka Linux.com využívající Joomla [22]

1.1.3 DRUPAL

Poslední z trojice zvolených open-source systémů pro správu obsahu je systém Drupal, který je rovněž naprogramovaný v jazyce PHP a mezi jím podporované databáze patří MySQL, PostgreSQL a SQLite. Další typy databází mohou být použity po přidání rozšiřujících modulů [23].

Zkušení developeri popisují Drupal jako nejsilnější CMS, jeho použití ale patří k těm nejtěžším. Jeho flexibilita z něj dělá druhý nejpoužívanější systém pro správu obsahu na světě, ovšem obliba tohoto systému mezi nováčky je nízká [24].

Největší rozdíl odlišující Drupal od ostatních CMS systémů spočívá v jeho instalaci, neboť je během ní uživatel vyzván ke konfiguraci své stránky [24].

Drupal také přistupuje k tvorbě obsahu více technicky. Webové stránky jsou řízeny takzvaným „blokovým“ systémem, kde každý blok tvoří čisté HTML. Některé klasické prvky jsou předpřipraveny v sadách, ale uživatel zřejmě stráví mnoho času tvorbou vlastních [24].

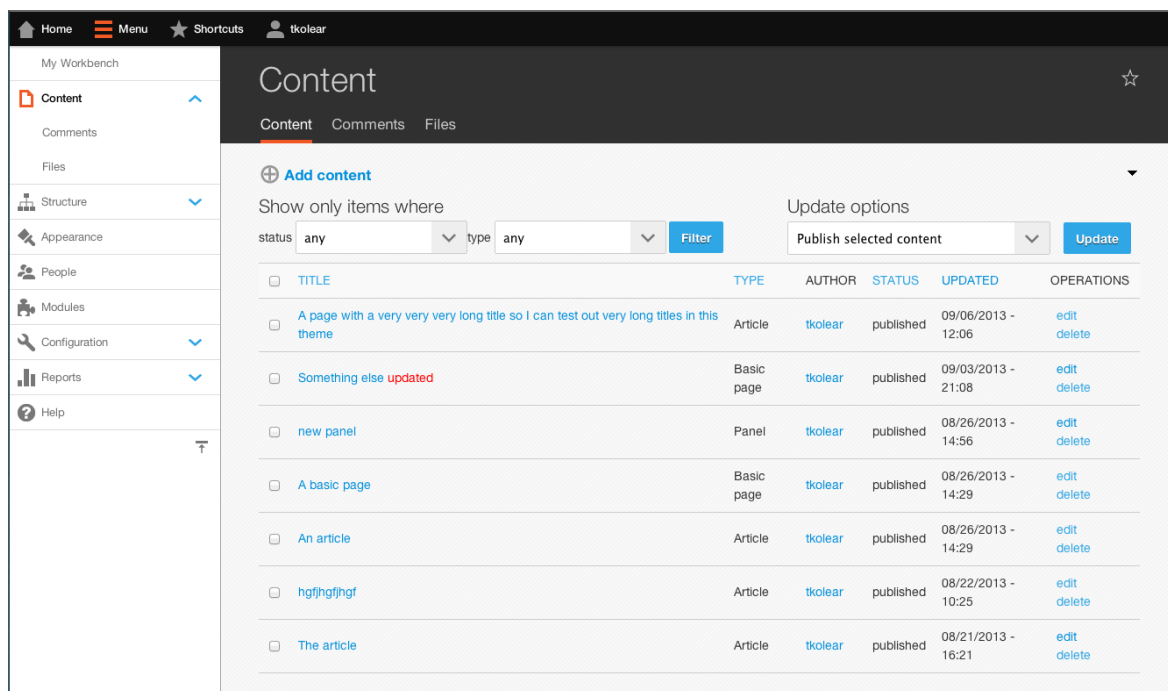
Výhody

Pokud je cílem uživatele vytvořit velké webové stránky s rozsáhlými funkcemi a velkou mírou customizace, Drupal je správná volba. Jeho nabídka funkcionalit je obrovská a je postavená na výkonu [6].

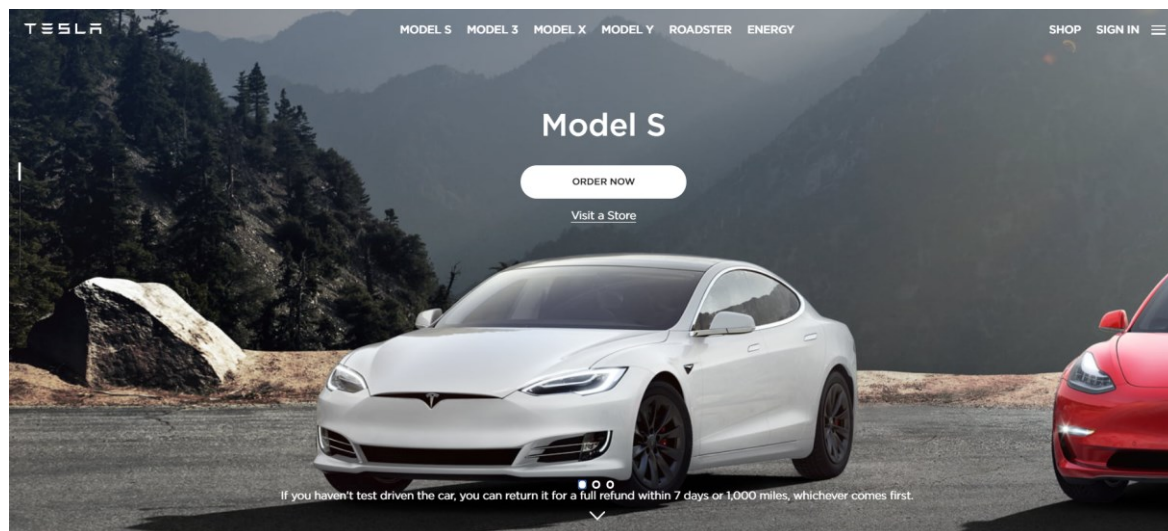
Nevýhody

Drupal není vhodný pro úplné začátečníky s CMS. Při práci s tímto systémem je naopak nutné mít pokročilé znalosti programovacích jazyků HTML, CSS a PHP [25].

Jak instalační rozhraní, tak rozhraní pro úpravy nepůsobí oproti jiným CMS systémům uživatelsky přívětivě, i když jejich funkcionalita není tak rozsáhlá [25].



Obrázek 5: CMS Drupal [26]



Obrázek 6: Stránka Tesla.com využívající Drupal [27]

1.2 KOMERČNÍ ŘEŠENÍ

Některé obce a města svěřují tvorbu svých portálů odborným firmám, které provádí kompletní návrh a realizaci webových stránek a velmi často dodávají i vlastní CMS systém. K vlastním CMS systémům jsou poskytovány příručky, školení, případně garantovaná technická podpora. Tyto služby se liší podle nabídky konkrétních firem.

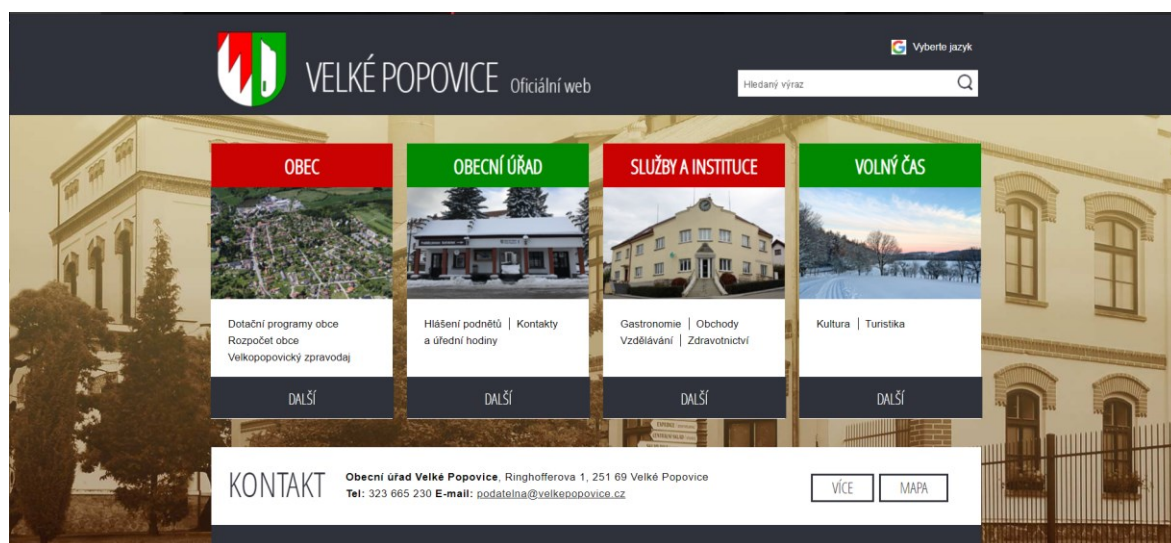
1.2.1 VISMO

Jedná se o redakční systém od firmy Webhouse, která má své velké pole působnosti ve Zlínském kraji. Firma nabízí realizaci webových stránek nejen pro obce a města, ale i komerční subjekty, s vlastním redakčním systémem Vismo Online. Zákazníkům jsou nabízeny různé varianty webových stránek – mohou to být stránky ze vzorkovníku nebo stránky řešené individuálně dle představ zákazníka [9].

Úprava webových stránek probíhá pomocí principu „*what you see is what you get*“ (WYSIWYG), tedy dle anglického překladu „co vidíš, to dostaneš“. K editaci stránky dochází přímo tam, kde se uživatel nachází a není proto potřeba přecházet do administrační sekce [10].

Web je možné rozšířit pomocí široké nabídky modulů – od běžných, jako je anketa nebo fórum, až po moduly pro výběrová řízení nebo například jazykové mutace [10].

Systém se prezentuje jako takzvaný „bezbariérový systém“. Zde je kladen důraz na dostupnost obsahu webu pro osoby se zrakovým či fyzickým postižením, které k tomuto obsahu přistupují pomocí různých kompenzačních pomůcek [10].



Obrázek 7: Webové stránky obce Velké Popovice od firmy Webhouse [11]



Obrázek 8: Webové stránky města Otrokovice od firmy Webhouse [12]

Zhodnocení

Webové stránky měst a obcí od firmy Webhouse působí vizuálně velice hezky. Kvalita se samozřejmě liší u konkrétních měst a obcí dle jejich vizí a smyslu pro detail, které přikládají svým webovým stránkám.

V přehlednosti webových stránek hraje velkou roli už samotná velikost daného města nebo obce (viz. Obrázek č. 11 a Obrázek č. 12). Stránka obce Velké Popovice působí velice jednoduše a přehledně, zatímco stránky města Otrokovice obsahují již na hlavní stránce velké množství odkazů a záložek, což zásadně snižuje orientaci na stránce.

V případě stránek města Otrokovice je však třeba velmi kladně ohodnotit on-line systém pro objednání občanů k různým úkonům na městském úřadu. Podobné systémy by byly vhodné ve více městech České republiky.

WYSIWYG

WYSIWYG, tedy z anglického překladu „*co vidíš, to dostaneš*“, označuje editory, jejichž hlavní výhodou je, že umožňují provádět úpravy v takovém zobrazení, v jakém bude stránka prezentována na webu [13]. Další výhodou je, že stránky lze formátovat bez znalosti jazyka HTML. K nevýhodám WYSIWYG editorů z vlastních zkušeností určitě patří velice špatný, generovaný HTML kód. Proto se WYSIWYG nehodí pro vytváření celých webů, ale spíše tam, kde jsou potřeba pouze různé modifikace textů.

Mezi známé WYSIWYG editory lze zařadit například weby jako Webnode.cz, SimpleSite.com nebo Weebly.com.

1.2.2 GALILEO CONTENT MANAGER

Jedná se o redakční systém od firmy Galileo Corporation. Tato firma se zabývá tvorbou webů pro obce, města a další různé organizace. V jejich správě je přes 2500 webů [14]. Kromě návrhu a realizace webových stránek nabízí firma spoustu služeb v kategorii získání grafických materiálů pro web samotný, jako například 360° panoramatické fotografie nebo virtuální prohlídky [15].

Aktualizace a vytváření struktur stránek probíhá intuitivně, bez potřeby příruček. Systém je často aktualizován a inovován tak, aby korespondoval s aktuálním technickým vývojem [16].

Nechybí zde ani možnost webové stránky rozšiřovat o desítky různých modulů a aplikací, aby odpovídaly představám zákazníka. Systém nabízí možnost multiuživatelské administrace [16].



Obrázek 9: Webové stránky obce Malhotice od firmy Galileo Corporation [17]

Zhodnocení

Webové stránky měst a obcí od Galileo Corporation nepůsobí tak moderně jako stránky od firmy Webhouse. Firma se prezentuje správou více než 2000 stránek, avšak při podrobném procházení referencí lze narazit na značnou skupinu stránek, které působí dojem, že už hodně dlouho nebyly spravovány a využívány. V některých případech stránky dokonce ani neexistují.

Mnoha stránek vzbuzuje dojem, že rozšiřovacích modulů a aplikací je na nich až příliš mnoho. U webových stránek měst a obcí je podle názoru autora práce už tak velký problém s přehledností a orientací na stránce díky velkému množství informací, které jsou

zde umístěny. Lze tedy považovat za zbytečné na stránku umisťovat moduly, které například ukazují graf teplotního vývoje v dané lokalitě za uplynulý týden, které nemají po stránce informativní s obcí nebo městem nic společného.

2 MOŽNOSTI PŘI REALIZACI WEBOVÉ APLIKACE

Webovou aplikaci lze vytvořit na základě využití různých technologií. Frontendovou část tvoří zpravidla jazyky HTML a CSS. Na straně backendu lze na základě kritérií jako rychlost, výkon aplikace, škálovatelnost, podpora nebo cena zvolit jazyk PHP nebo frameworku ASP.NET [60]. U frameworku ASP.NET lze využít návrhový vzor MVC.

2.1 PHP

PHP je programovací jazyk pracující na straně serveru. Jedná se o open-source, což znamená úplně otevřený kód a je zcela zdarma [60]. To může být podstatnou výhodou oproti frameworku ASP.NET, který může být dodáván i s webhostingovým poplatkem [60]. Některé verze frameworku ASP.NET jsou rovněž open-source.

Skript je vložen na webový server, který jeho celý proces následně zpracovává. Uživateli je zobrazen čistý výstup bez registrace, zda se jedná o statickou, HTML stránku nebo stránku vygenerovanou ze strany serveru [61].



Obrázek 10: Princip fungování PHP [61]

2.2 ARCHITEKTURA MVC

V případě pokročilejších webových aplikací je nutné se držet zvolené architektury, která bude tvořit strukturu aplikace [28].

Architektura MVC spočívá v oddělení logiky a výstupu. Aplikace se skládá ze tří částí – Model, View a Controller. O celou logiku aplikace se stará Model. View je v podstatě HTML šablona rozšířena základním PHP kódem a cykly. Propojení mezi Modelem a View zabezpečuje Controller. Dle získaných parametrů rozhoduje o tom, co se bude dále aplikovat [28].

Tato technologie byla autorem práce zvolena, neboť s ní pracoval v rámci předchozích pracovních projektů.

3 POUŽITÉ TECHNOLOGIE

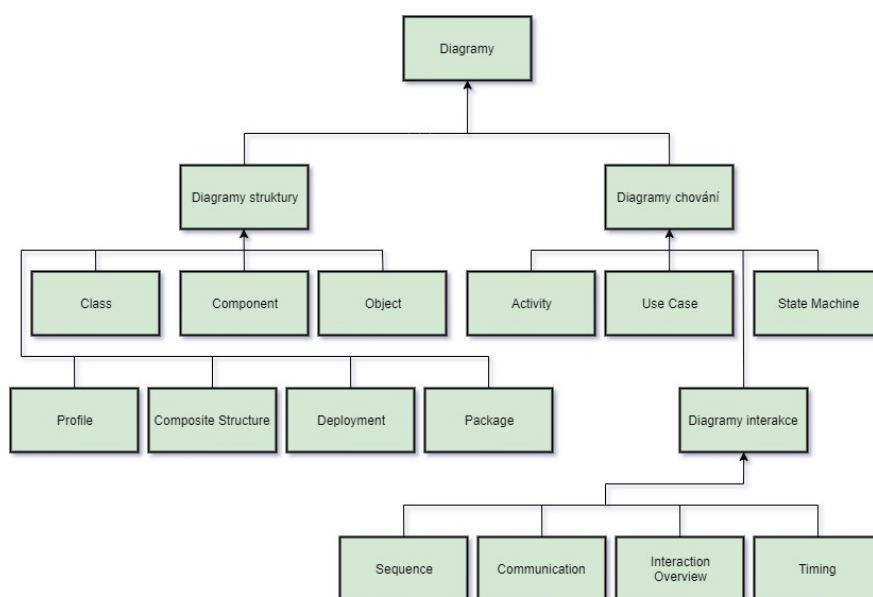
Před samotnou realizací aplikace bylo nutné si podrobně sepsat hlavní funkce aplikace a udělat její kompletní návrh. K tomuto účelu byl využit grafický jazyk UML. Pomocí drátěného modelu byl dále navrhnut základní vzhled a rozvržení prvků aplikace, který byl následně realizován pomocí jazyků HTML a CSS. Backend část byla zpracována pomocí technologie ASP.NET MVC s využitím Entity Framework. Pro práci s databází bylo využito jazyku SQL a pro realizaci některých funkcionalit skriptovacího jazyku JavaScript.

3.1 UML

UML je grafický jazyk, který je využíván k vizualizaci, navrhování a dokumentaci softwarového systému [29].

Díky vzrůstající komplexnosti systémů nelze začít okamžitým psaním kódu bez jakéhokoliv návrhu. Nutnost pracovat ve skupině programátorů s kvalitní týmovou komunikací a neustálé změny ze strany klienta jsou problémy, jež UML umožňuje minimalizovat [30]. Při prvotní analýze požadavků [30] může pomoci dojít k zjištění, co bude nutné programovat, a v pokročilé části projektu pak k rozhodnutí, jakým způsobem toho bude dosaženo.

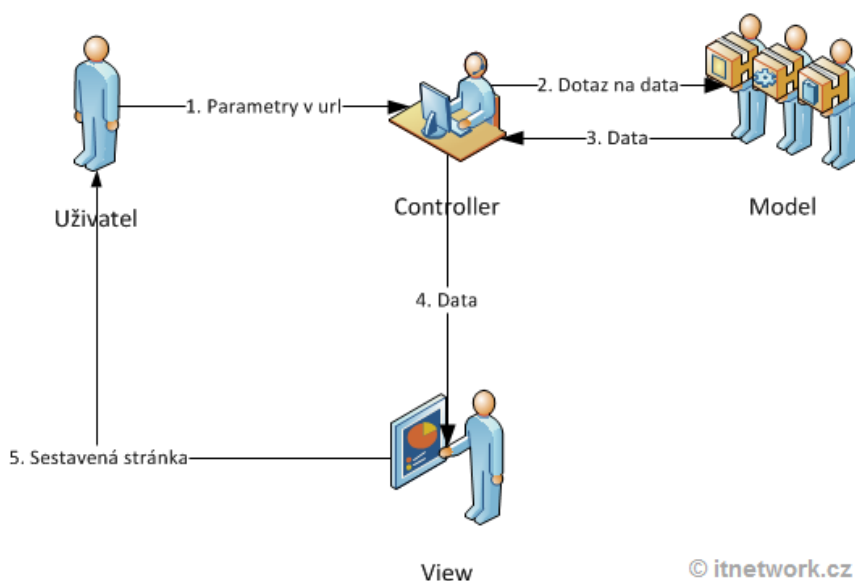
Jedna z největších výhod UML diagramů je jejich abstrakce. Jednotlivé diagramy prezentují specifický pohled na systém z určitého úhlu [30]. Zbytek systému je v danou chvíli zanedbán a je zobrazeno pouze to, co je aktuálně řešeno. Dochází tak ke snížení rizika špatného porozumění zadání – ať už mezi týmem nebo samotným zákazníkem.



Obrázek 11: Ukázka UML složeného ze 14 diagramů

3.2 ASP.NET MVC

Model-View-Controller (zkráceně MVC) je architektonický vzor využívaný především v oblasti webových frameworků. Skládá se ze tří částí: modelu (*Model*), pohledu (*View*) a řadiče (*Controller*). Hlavní funkcionalita tohoto vzoru spočívá v oddělení části, která se stará o logiku aplikace, od části, která má na starost výstup aplikace [31].



Obrázek 12: Architektura MVC [31]

3.2.1 MODEL

Model se stará o veškerou logiku a vše, co k ní patří – od validace až po databázové dotazy. Model může být chápán jako běžná C# třída. Hlavní funkcí této součásti frameworku MVC je zpracování vstupních parametrů a odeslání výstupních dat. Model nezohledňuje, odkud vstupní data přišla, ani to, jak bude vypadat finální zpracování a výpis dat [31].

V případě menších aplikací se model využívá ke koncepčnímu oddělení. V případě, kdy v aplikaci dochází ke čtení datové sady a jejího následného zobrazení, fyzická vrstva spolu se souvisejícími třídami v aplikaci není. Datová sada se v tomto případě stará o povinnost modelového objektu [32].

3.2.2 VIEW

Hlavním úkolem druhé části MVC zvané View je zobrazení výstupu. View je v podstatě HTML šablona, která zahrnuje jednotlivé stránky společně s tagy, díky kterým mohou být do šablony vloženy prvky jako proměnné společně s možností použití podmínek a cyklů. Šablony mohou zároveň obsahovat i jiné šablony, aby nedocházelo ke zbytečné duplicitě [31].

Funkce View lze rozdělit do dvou kategorií – první je funkce zobrazení výstupu, druhou pak funkce šablony. Logika obsažená ve View je minimální [31].

Stejně jako předchozí součást MVC zvaná Model, ani část View neřeší, odkud vstupní data pocházejí, neboť hlavní starostí této komponenty je jejich správné zobrazení.

3.2.3 CONTROLLER

Pomocí komponenty Controller probíhá komunikace mezi uživatelem a zbylými částmi MVC (Model a View). Controller dané komponenty sdružuje a zabezpečuje mezi nimi komunikaci na bázi životního cyklu stránky. Běžně bývá použit jeden Controller k jedné entitě [31]. Controller spravuje požadavky od uživatele přiřazené k danému modelu a v závěru dochází ke zvolení View – tedy pohledu, jehož prostřednictvím bude MVC zobrazeno [32].

3.3 HTML

Hypertext Markup Language (zkráceně HTML) je značkovací jazyk využívaný při tvorbě a popisu obsahu webu nebo dat [34]. Struktura jazyka je poměrně jednoduchá a lehce jí porozumí i úplný začátečník.

HTML jazyk je popsán pomocí takzvaných **tagů**, jejichž vlastnosti jsou popsány pomocí **atributů**. Prostřednictvím tagů lze jednotlivým elementům dávat určitou roli nebo význam, což slouží k rozlišení prvků, jako je tabulka, nadpis nebo prostý text. Tagy jsou prezentovány pomocí ostrých závorek (například `<h1>Nadpis</h1>`). Jedním ze speciálních tagů je například odkaz, díky kterému se lze pohybovat mezi jednotlivými stránkami a tím je mezi sebou propojit [37].

Aktuální verzi HTML je HTML5. V této verzi došlo k vylepšení již obsažených funkcí a přidání několika nových společně s API postavených na skriptech. Verze HTML5 obsa-

huje všechny součásti předchozí verze HTML4 a byla navržena tak, aby zaručila kompatibilitu jak v rámci platform, tak s ohledem na prohlížeče včetně jejich starších verzí. Jedním z dalších kritérií pro vývoj a publikaci HTML5 bylo přívětivé zpracování chyb [35].

K tvorbě webu pomocí jazyku HTML lze využít dva druhy editorů. Prvním je jakýkoliv textový editor, přičemž se autor práce domnívá, že je výrazně vhodnější zvolit takový editor, který přinejmenším zvládá rozlišení HTML syntaxe, validaci nebo funkci napovídání. Druhou možností je využití takzvaných WYSIWYG editorů, které nenutí uživatele učit se základy HTML jazyka, nýbrž umožňují poskládat si stránku podle jeho vlastních představ a následně mu vygenerují požadovaný kód [34].

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>This is a Heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

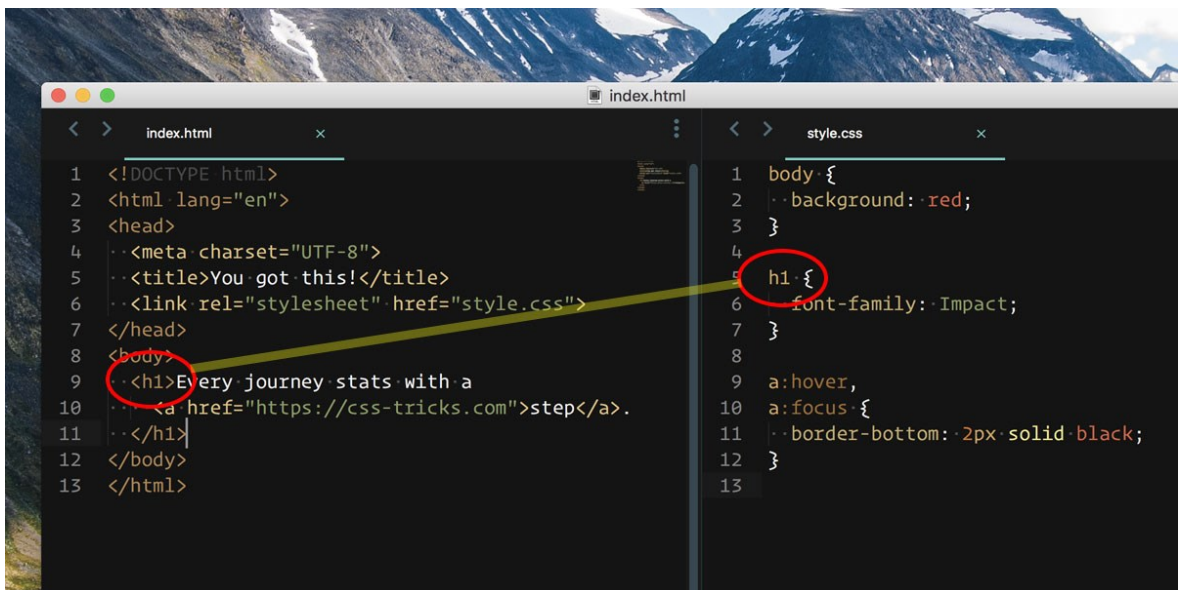
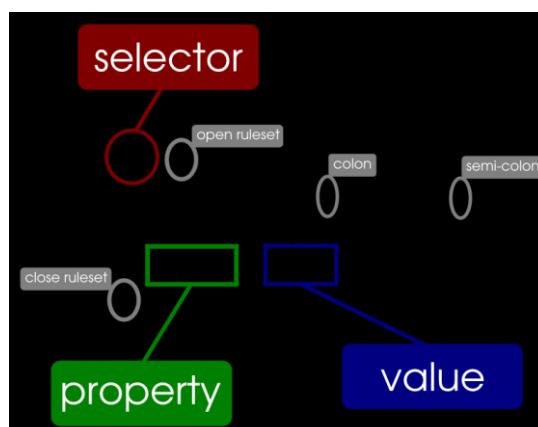
Obrázek 13: Struktura HTML [36]

3.4 CSS

Cascading Style Sheets (zkráceně CSS) je stylovací jazyk využívaný ke stylizaci HTML prvků [35]. Pomocí CSS lze definovat podobu stránky jako celku, například barvu pozadí, font písma nebo formu pro *header* (záhlaví) a *footer* (patičku) stránky. Prostřednictvím CSS je také možno definovat specifické vlastnosti pro každý element stránky pomocí atributů *class* nebo *id*.

Aktuálně používanou verzí jazyku CSS je CSS3, v níž došlo k vylepšení již obsažených funkcí a přidání nových funkcí, které programátorům pomáhají v řešení řady chyb bez nutnosti použití složitých scriptů či přebytečných obrázků. Tato verze také obsahuje řadu podpor pro animace, průhlednost, zakulacené pozadí a mnoho dalších prvků [35].

CSS lze zapsat přímo do elementu pomocí atributu *style* (například `<div style="max-width: 400px; background-color: rgba(255,255,255,0.93);">`), avšak vhodnějším řešením je jednotlivé styly elementů umístit do souboru s příponou *.css* a k jejich identifikaci použít již zmíněné atributy *class* nebo *id*. Použitý CSS soubor je pak nutné nalinkovat do layoutu stránky – například `<link rel="stylesheet" href="@Url.Content("/Content/style.css") type="text/css" />`.

Obrázek 14: Stylizace elementu *h1* [38]

Obrázek 15: Struktura CSS [39]

3.5 SQL

Structured Query Language (zkráceně SQL) je dotazovací jazyk, který je využíván pro manipulaci s relační databází a jejími daty [41].

Relační databáze je databáze tvořená pomocí tabulek, v níž každá tabulka obsahuje informace o jedné položce. Každá tabulka je následně rozdělena pomocí řádků a sloupců, ve kterých jsou obsaženy atributy dané tabulky [42]. Již při vytváření tabulky je nutné každému atributu přiřadit datový typ, jelikož relační databáze je takzvaně typovaná, což znamená, že zvolený datový typ je pevně daný a nelze do něj uložit jiný (například nelze uložit textový řetězec do datového typu *INT*). U některých datových typů lze nastavit i maximální délku – například u datového typu *nvarchar* lze omezit počet znaků pomocí zápisu *nvarchar(150)*. Toto omezení je však nevyhnutelně nutné ošetřit také na straně aplikace, aby byl uživatel včas varován a nedošlo například k jejímu pádu. Při zakládání je taktéž třeba určit, zda danému atributu bude povolena prázdná hodnota *null*, nebo zda bude tato možnost zakázána a tím pádem bez této hodnoty nebude možno danou operaci (například zápis do databáze) dokončit.

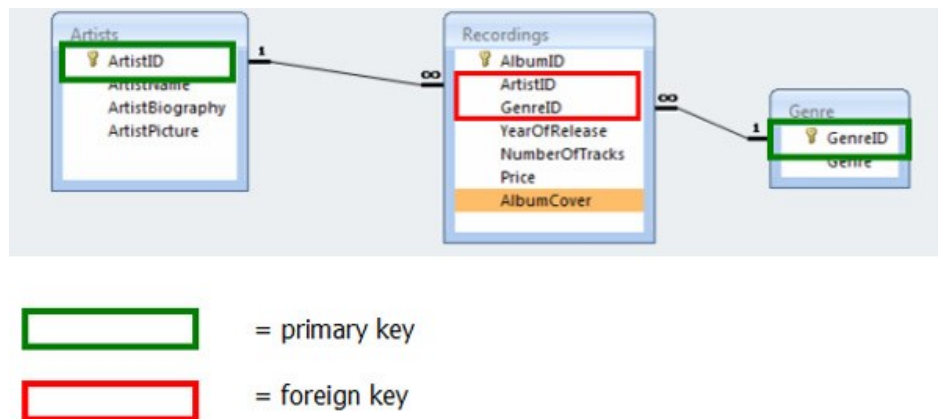
	Column Name	Data Type	Allow Nulls
▶	UserID	int	<input type="checkbox"/>
	Name	nvarchar(150)	<input type="checkbox"/>
	Surname	nvarchar(150)	<input type="checkbox"/>
	Mail	nvarchar(150)	<input type="checkbox"/>
	Login	nvarchar(150)	<input type="checkbox"/>
	Password	nvarchar(150)	<input type="checkbox"/>
	Role	int	<input type="checkbox"/>
	ResetPasswordCode	nvarchar(150)	<input checked="" type="checkbox"/>

Obrázek 16: Struktura tabulky *Users*

Každá tabulka obsahuje svůj jedinečný identifikátor neboli **primární klíč** (*PRIMARY KEY*). Nejčastěji se za účelem tohoto označení používá položka *ID*, které se automaticky generuje pomocí komponenty *Identity*. Jako primární klíč může být použito například rodné číslo.

Druhou možností je využití takzvaného **cizího klíče** (neboli *FOREIGN KEY*) prostřednictvím vztahu definovaného mezi dvěma tabulkami, kde cizí klíč jedné tabulky obsahuje hodnotu primárního klíče tabulky druhé. V aplikaci autora práce je cizího klíče využito například v případě, kdy je třeba v tabulce *dbo.Post* uložit do atributu *CreatedBy* ID uživatele, který tento příspěvek vytvořil.

V případě manipulace s daty lze využít funkci *Stored procedure*. Jedná se o uloženou část SQL kódu, kterou lze opakovaně používat. Procedury podporují použití parametrů, díky čemuž lze kód modifikovat.



Obrázek 17: Primární a cizí klíč [43]

```
USE [BcDatabase]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Patrik Valuch
-- Create date: 2019-06-03
-- Description: Add user
-- =====
ALTER PROCEDURE [dbo].[AddUser]
    @Name      varchar(150),
    @Surname   varchar(150),
    @Mail      varchar(150),
    @Login     varchar(150),
    @Password  varchar(150)
AS
BEGIN
    SET NOCOUNT ON;

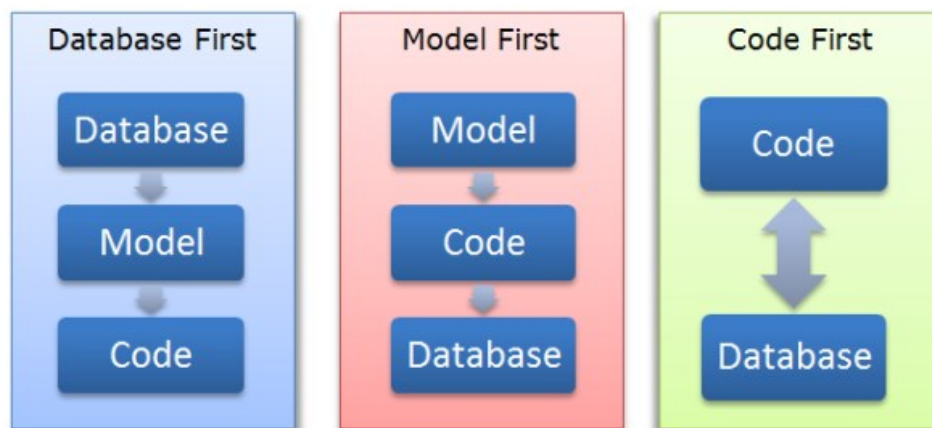
    INSERT INTO [dbo].[Users]
    (
        [Name],
        [Surname],
        [Mail],
        [Login],
        [Password]
    )
    VALUES
    (@Name, @Surname, @Mail, @Login, @Password)
END
```

Obrázek 18: Příklad využití *Stored procedure*

3.6 ADO.NET ENTITY FRAMEWORK

Entity Framework (EF) je open-source objektově-relační mapovací framework pro ADO.NET [44]. Jedná se o součást .NET Framework, ale jeho distribuce probíhá nezávisle vlastním release cyklem. EF je samostatná knihovna vydávaná nezávisle na .NET Framework v podobě pull requests, což v případě nalezení chyby umožňuje vydavatelské společnosti Microsoft zahrnout danou opravu do další připravované verze EF [45].

EF se skládá ze tří hlavních komponent – databázové struktury, konceptuálního modelu a takzvaného C-S mapování. **Databázová struktura** popisuje tabulky, sloupce, datové typy sloupců apod. **Konceptuální model** popisuje konkrétní CLI třídy a jejich vlastnosti. **C-S mapování** pak spojuje tyto dvě předchozí komponenty dohromady. Property třídy je mapována zvoleným způsobem na určitý sloupec v tabulce [45].



Obrázek 19: Tři možné přístupy EF [46]

3.7 JAVASCRIPT

JavaScript (JS) je skriptovací jazyk, který je používán při tvorbě webových stránek nebo aplikací k rozšíření jejich dynamické funkcionality [47].

JS se odlišuje tím, že například oproti jazyku PHP, který je serverový, běží na straně uživatele přímo ve webovém prohlížeči. Díky tomu umožňuje vytvářet dynamické prvky nebo například konkrétní prvky skrýt do doby, než se je uživatel sám rozhodne zobrazit a tím zpřehlednit webovou stránku. JS je podle názoru autora práce velmi vhodným prostředkem k vytvoření HTML editoru, čehož je hojně využíváno v editorech na webových stránkách, které jsou většinou založeny na JS [49].

Pomocí JS lze také snížit počet znovunačtení stránky (tzv. *reload*) po vykonání daných operací aplikace – například pokud návštěvník na webové stránce produktu stiskne tlačítko *Koupit*, není nutné, aby došlo ke znovunačtení stránky, neboť JS lze nastavit ke snížení dostupných kusů bez znovunačtení. JS lze uplatnit také u prvků, které pracují s daty a časem v aplikaci autora této bakalářské práce, konkrétně u elementů pro výběr data a času uživatele. Dalším z možných využití JS je obohacení webových stránek pomocí efektů, například přidáním padajících sněhových vloček v období Vánoc [49].

V případě JS se jedná o jazyk dynamicky typovaný, tudíž jeho překlad probíhá za běhu a jeho následné provedení je dáno zdrojovým kódem. Jazyk obsahuje několik datových typů (*number*, *string*, *boolean*, *object*) a funkcí, jako například *callback* [49].

```
$(".btn-show-rating").click(function () {
    $(this).parent().next().next().show();
});
$(".btn-hide-rating").click(function () {
    $(this).parent().parent().hide();
});
$(function () {
    $(".daterange1").daterangepicker({
        "buttonClasses": "button button-rounded button-mini nomargin",
        "applyClass": "button-color",
        "cancelClass": "button-light",
        locale: {
            format: 'DD/MM/YYYY',
        },
    });
});
```

Obrázek 20: JS pro skrytí a zobrazení části hodnocení akcí a JS pro *datepicker*

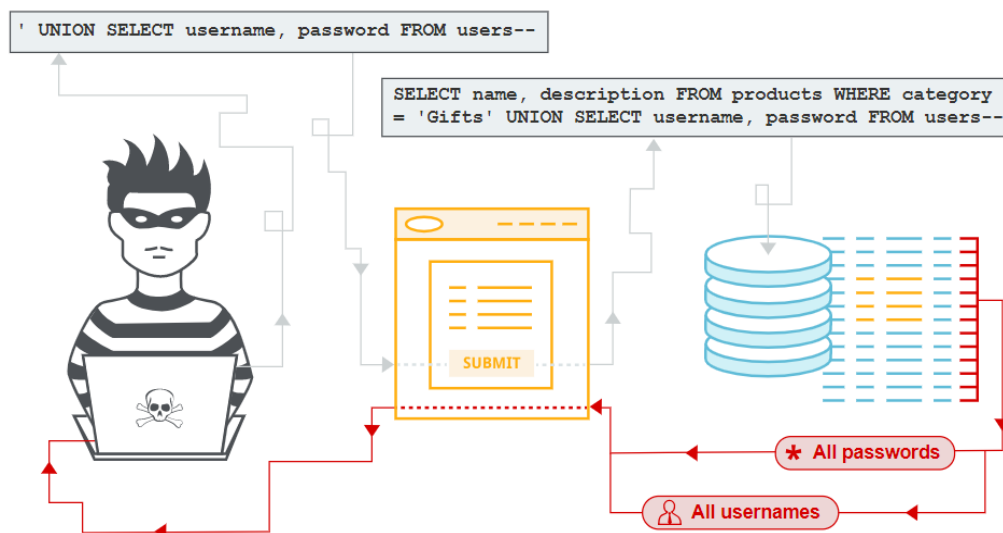
4 RIZIKA WEBOVÉ APLIKACE

Webové aplikace obsahují řadu osobních údajů uživatelů, jež tyto aplikace používají. Cílem každého vývojáře nebo firmy by mělo být, aby jeho aplikace byla dostatečně zabezpečená a nedošlo tak k úniku a následnému zneužití těchto dat ze strany útočníka.

4.1 Injection

K této formě bezpečnostní hrozby dochází při zpracování vstupních dat, mezi něž mohou patřit jak vstupy od uživatele, tak vstupy z importů, pokud nedošlo k jejich důkladné kontrole pomocí validace. Patří zde například *SQL injection*, *OS injection*, *LDAP injection* a další.

Pokud není aplikace proti tomuto druhu útoku ošetřena, může dojít k nedovolenému přístupu k datům v databázi, což potenciálně útočníkovi umožní s databází neomezeně manipulovat či z ní odcizit citlivá data (například kompletní osobní údaje uživatelů aplikace) [50].

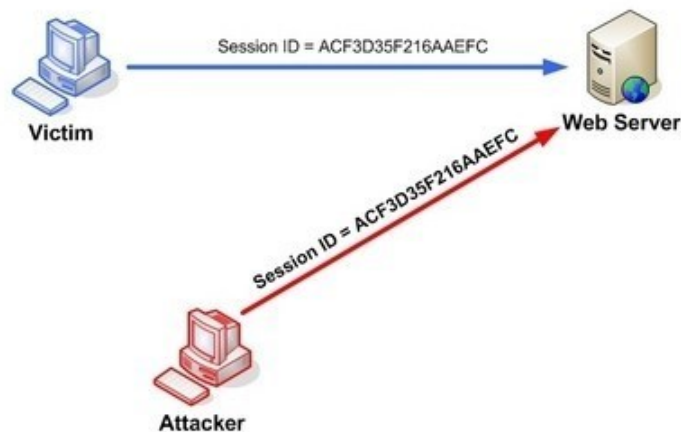


Obrázek 21: Příklad SQL injection [51]

4.2 Broken Authentication and Session Management

K této bezpečnostní hrozbě dochází při správě uživatelů. Útočník se zde snaží napadnout ty části aplikace, které se starají o autentizaci a správu uživatelů s úmyslem přivlastnit

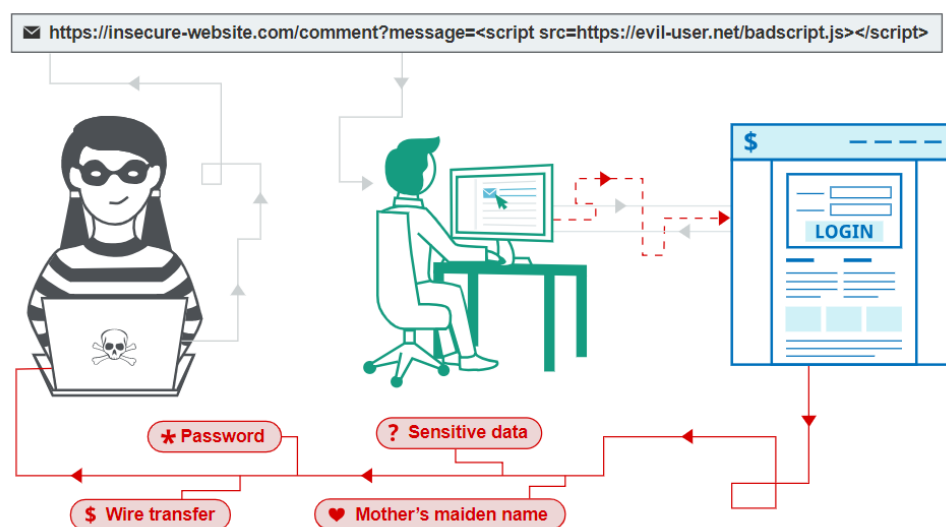
si identitu jiného uživatele, aniž by o tom dotyčný uživatel nebo administrátor věděli. Útočník k tomuto útoku může využít způsoby jako jsou přihlášení, reset hesla nebo funkce zapamatování uživatele. V těchto sekcích se útočník snaží najít bezpečnostní mezeru s cílem pokusit se ukrást identitu jiného uživatele [50].



Obrázek 22: Příklad Broken Authentication and Session Management [52]

4.3 Cross-Site Scripting

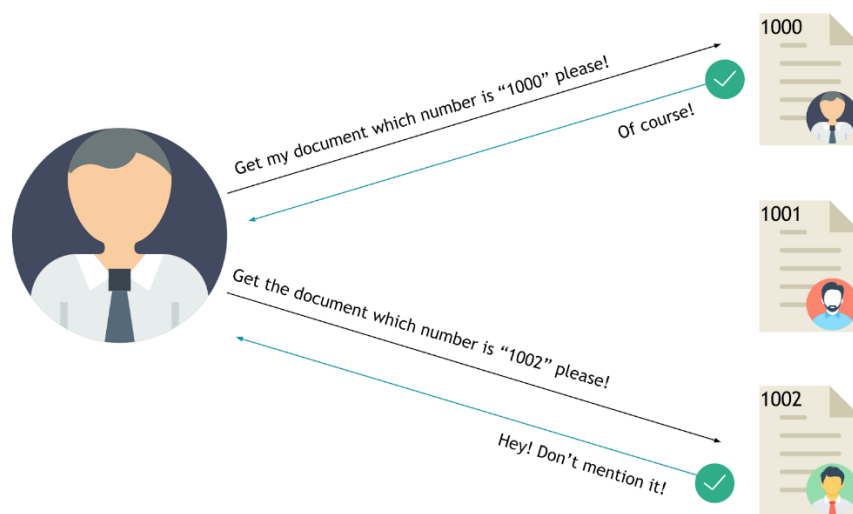
Cross-Site Scripting je bezpečnostní hrozba, k níž dochází v případě, kdy se útočníkovi podaří do stránky umístit vlastní kódy nebo skripty. Prostřednictvím tohoto útoku může útočník narušit zobrazení dané stránky nebo se pokusit o odcizení údajů uživatele. Prevencí proti tomuto typu útoků je důkladné kontrolování veškerých vstupů od uživatelů, což výrazně snižuje riziko této bezpečnostní hrozby [50].



Obrázek 23: Příklad Cross-Site Scripting [53]

4.4 Insecure Direct Object References

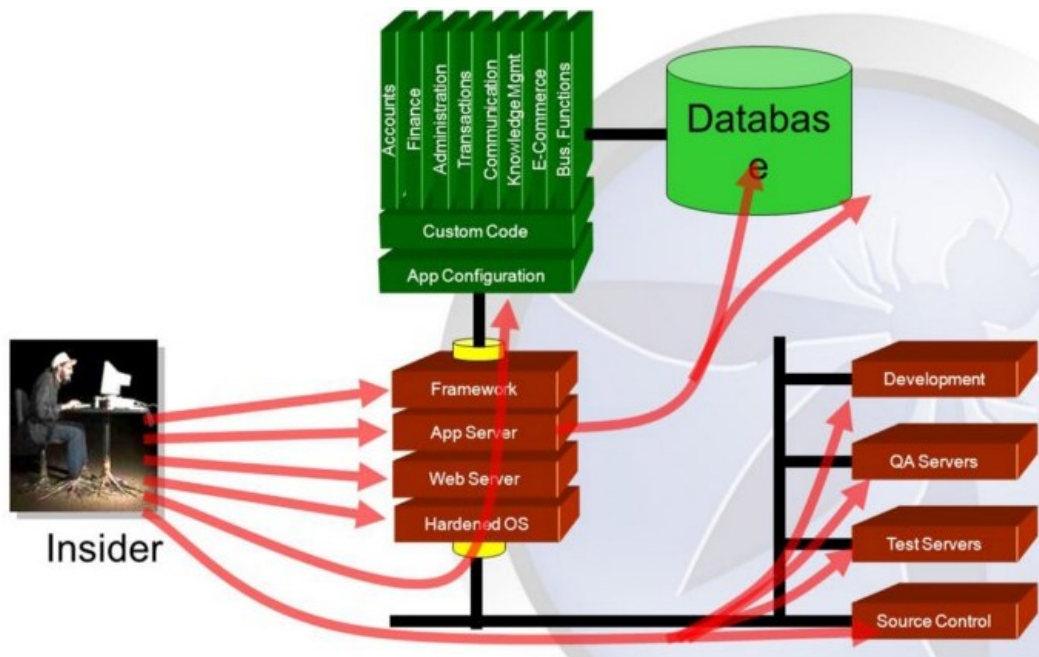
Jedná se o bezpečnostní hrozbu, která je způsobena špatným nastavením nebo ošetřením úrovně oprávnění pro přihlášené uživatele. V aplikaci musí být striktně dáno, kde je daný uživatel oprávněn se pohybovat a ke kterým datům má mít přístup. V aplikaci nesmí dojít k situaci, kdy se *User1* dostane přes *Uniform Resource Locator* (URL) na uživatelský profil uživatele *User2* a dostane tak přístup k jeho osobním údajům [50].



Obrázek 24: Příklad Insecure Direct Object References [54]

4.5 Security Misconfiguration

K této bezpečnostní hrozbě dochází například v důsledku používání zastaralé verze softwaru nebo obecně chybou konfigurace systému. Při vývoji a následném testování aplikace mohou být využity účty s jednoduchými přihlašovacími údaji a jejich existence v plném provozu aplikace může být snadnou cestou pro neoprávněné vniknutí do aplikace. Stejný problém nastává v situaci, kdy nedojde ke změně defaultních přihlašovacích údajů – například nechvalně známou volbou je uživatelské jméno *admin* s heslem *admin* [50].



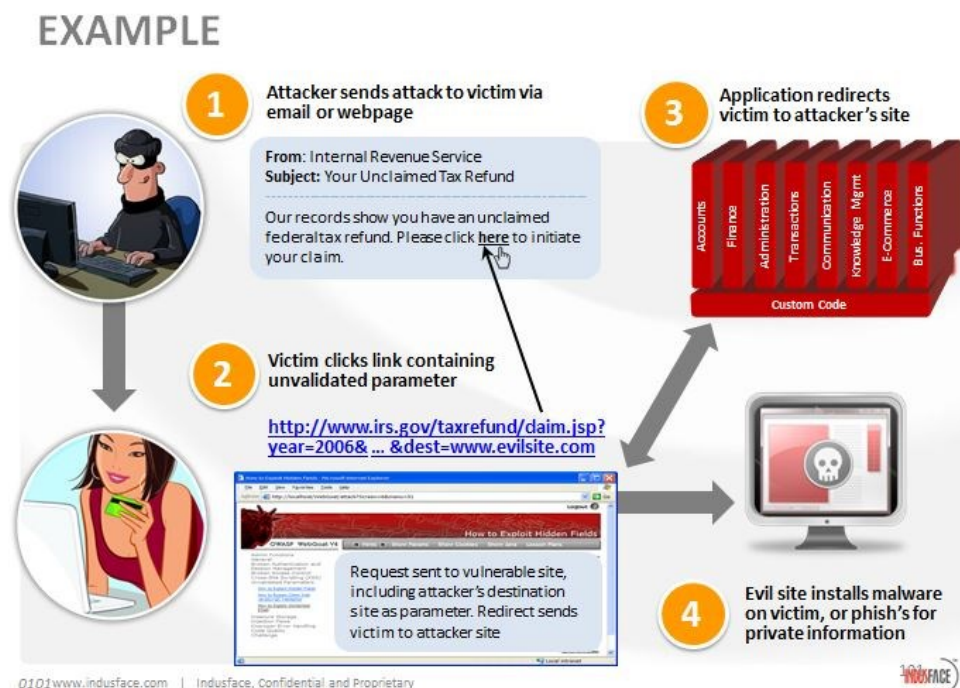
Obrázek 25: Příklad Security Misconfiguration [55]

4.6 Sensitive Data Exposure

Sensitive Data Exposure je forma bezpečnostní hrozby, která se týká osobních údajů uživatelů. Pokud dochází k práci s daty (manipulaci, posílání či uchovávání), je nutné je šifrovat či případně využít šifrovanou komunikaci, aby je útočník nemohl odchytit a zaznamenat.

V případě přihlašování nebo registrace je vhodné použití tzv. **hashování** (rozptylování), protože nelze zaručit, že přihlášení uživatele probíhá ze zabezpečených sítí. Existuje zde riziko, že v důsledku častého používání stejných přihlašovacích údajů ze strany uživatelů budou při prolomení aplikace ukořistěny jejich přihlašovací údaje a ty následně použity pro získání přístupu do jiných účtů.

Bezpečnost aplikace lze ověřit pomocí takzvaných penetračních testů [50].



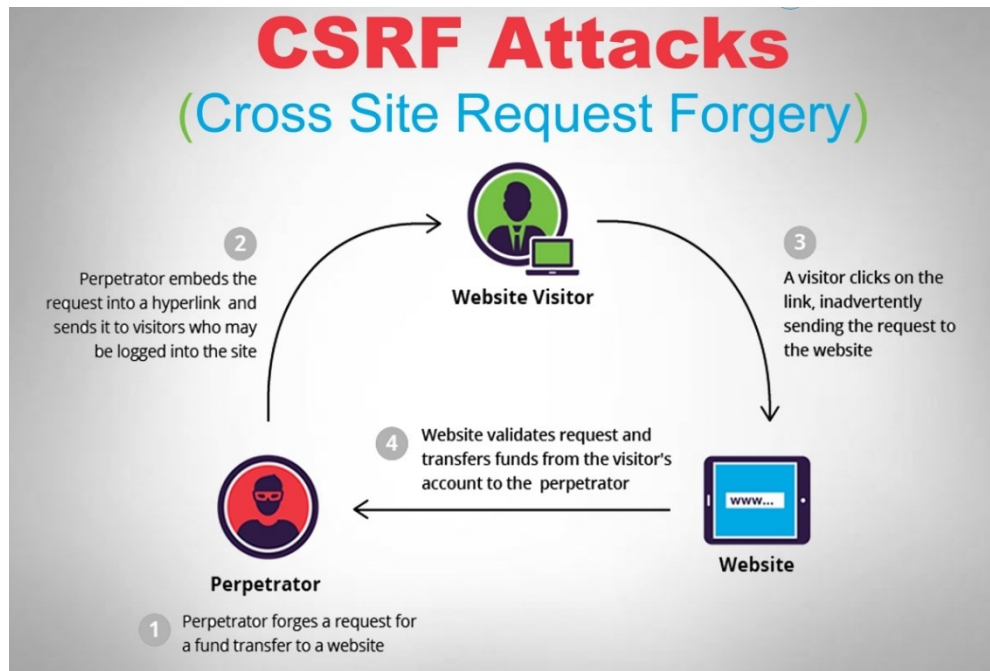
Obrázek 26: Příklad Sensitive Data Exposure [56]

4.7 Missing Function Level Access Control

Jedná se o bezpečnostní hrozbu, která je způsobena špatným ošetřením uživatelských rolí, zejména z hlediska funkcí a metod, které má právo využívat pouze administrátor. Nesmí dojít k situaci, kdy útočník využije neošetřené funkce a podaří se mu v aplikaci provést úpravy (přes URL adresu či jiným způsobem). Například v aplikaci, kde má právo založit nového uživatele pouze administrátor, nesmí dojít k tomu, aby po připojení údajů jako jsou *login*, *password* a *role* k URL adrese na přidání uživatele došlo k založení uživatele ze strany útočníka [50].

4.8 Cross-Site Request Forgery

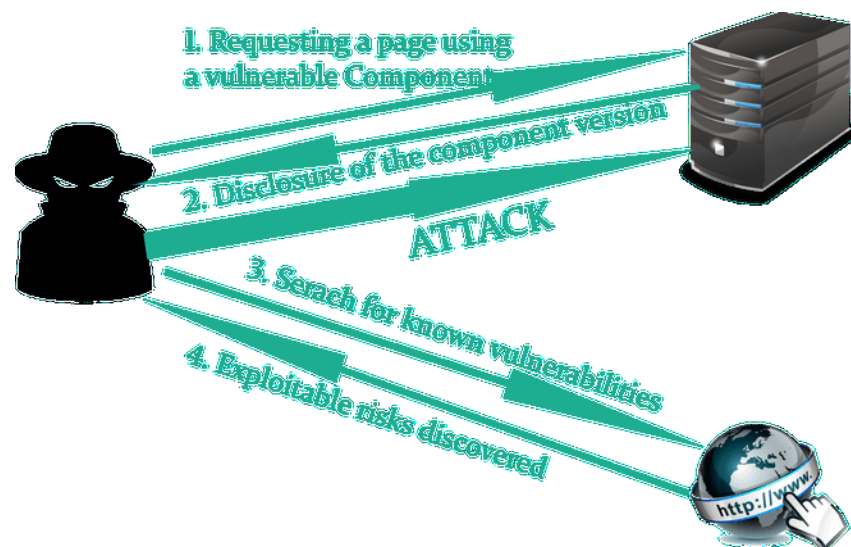
Název *Cross-Site Request Forgery* označuje bezpečnostní hrozbu, během níž se útočník snaží přimět uživatele k vykonání akce, kterou on sám chce. Nejčastěji se pro tyto účely využívají podvodné e-maily, které mají působit dojmem, že byly zaslány autory aplikace nebo administrátorem. Jejich obsahem je útočníkův předem připravený odkaz, který může být uživateli prezentován jako odkaz pro opětovné potvrzení registrace či změnu hesla. Bezpečnostní problém nastává, pokud je uživatel v době kliknutí na daný odkaz přihlášen v aplikaci, neboť útočník tak získá okamžitý přístup k jeho uživatelským datům [50].



Obrázek 27: Příklad Cross-Site Request Forgery [57]

4.9 Using Components with Known Vulnerabilities

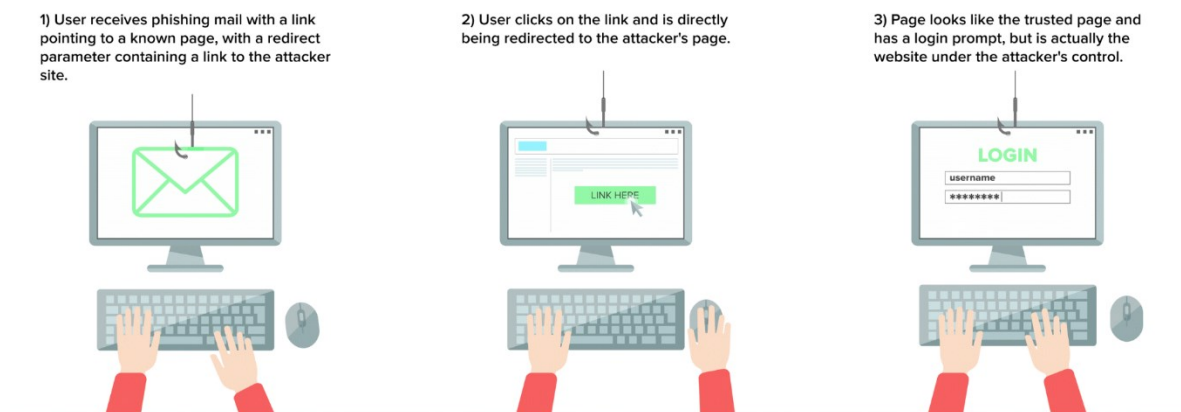
K této bezpečnostní hrozbě dochází již během vývoje aplikace. Proces vývoje totiž lze urychlit a zjednodušit prostřednictvím volně dostupných knihoven, frameworků nebo jiných služeb, které se ovšem v případě, že si autor aplikace důkladně nezkontroluje jejich momentální nedostatky, mohou stát bezpečnostním rizikem. Nedostatky těchto komponent totiž bývají ve většině případů veřejně dostupné a tím pádem dosažitelné i potenciálnímu útočníkovi, který se může pokusit těchto chyb využít [50].



Obrázek 28: Příklad Using Components with Known Vulnerabilities [58]

4.10 Unvalidated Redirects and Forwards

K poslední bezpečnostní hrozbě zmíněné v této části bakalářské práce dochází při přesměrování uživatele, například na jinou stránku nebo sekci aplikace pomocí URL. Útočník zde může využít parametru URL a uživateli podsunout odkaz na vlastní stránku, která se tváří jako požadovaná stránka. Uživatel se tak může například zkusit přihlásit nebo zadat jiné osobní údaje, které se okamžitě ocitají v rukou útočníka [50].



Obrázek 29: Příklad Unvalidated Redirects and Forwards [59]

II. PRAKTICKÁ ČÁST

5 WEBOVÁ APLIKACE PRO STRÁNKY MĚST

Realizace samotné webové aplikace je rozdělena do několika kapitol. V prvním kroku bylo nutné do detailu zmapovat a sepsat všechny základní funkce aplikace, aby bylo možno z těchto poznámek vycházet při dalších návrzích a samotné realizaci.

Z těchto základních funkcí byl vytvořen *Use Case model* společně se scénáři pro každou funkcionalitu, zatímco struktura databáze byla popsána pomocí fyzického modelu. Tento model poskytl základní náhled na pohyb dat v aplikaci a její funkčnost. Následně byla vytvořena databáze dle návrhu, avšak během vývoje byla její podoba několikrát obměněna.

Dalším krokem byl celkový návrh a realizace *frontend* části aplikace. Cílem bylo skloubit jednoduchost, funkčnost a přehlednost, jelikož je aplikace rozšířením pro webové stránky měst nebo obcí, které bývají často zahlcené spoustou informací, a tudíž dost nepřehledné. Pro autora práce bylo samozřejmě taktéž důležité, aby i vzhled aplikace byl poutavý a okamžitě tak zaujmul nové uživatele.

Dalším bodem bylo postupné programování základních funkcí dle počátečního návrhu. V počátku vývoje aplikace bylo využíváno metody *stored procedure* pro komunikaci s databází, ale po několika problémech a po konzultaci s vedoucím práce se autor rozhodl použít EF, který se nakonec ukázal jako lepší řešení. K realizaci některých prvků webu byla využita HTML5 šablona *Canvas*.

Posledním bodem při vývoji aplikace bylo její testování a oprava chyb – jak na straně frontendu, tak backendu.

5.1 ANALÝZA POŽADAVKŮ WEBOVÉ APLIKACE

Analýza požadavků byla autorem práce rozdělena do několika částí. První kategorií tvoří požadavky uživatelské. V aplikaci budou používány dvě uživatelské role, *registrovaný uživatel* a *administrátor*. Pro registrovaného uživatele bude aplikace nabízet možnost přihlášení, se kterou se pojí funkce pro změnu zapomenutého hesla. Neregistrovaný uživatel bude mít možnost se registrovat.

Mezi hlavní požadavky na aplikaci patří rozdělení příspěvků do čtyř kategorií: *aktuality*, *diskuze občanů*, *obecní/městské akce* a *nabídky/poptávky*. Registrovaný uživatel bude moci zakládat příspěvky ve všech kategoriích, ale následné zobrazení příspěvku bude podléhat schválení ze strany administrátora. Registrovaný uživatel bude vždy informován, zda

jeho příspěvek byl či nebyl schválen ke zveřejnění. Příspěvky vytvořené administrátorem budou zobrazeny okamžitě, jelikož se může jednat o oficiální zástupce dané obce či města. Každý z příspěvků bude možno komentovat, přičemž komentáře nebudou podléhat schválení. Pro případ nevhodných komentářů nebo špatné kontroly příspěvku bude aplikace obsahovat možnost nahlášení – jak pro příspěvky, tak jednotlivé komentáře.

Registrovaný uživatel bude mít možnost svůj příspěvek smazat či jej editovat, přičemž po editaci bude nutné příspěvek znovu schválit administrátorem. Administrátor bude moci smazat jakýkoliv příspěvek nebo komentář.

Další požadavky se vázaly k jednotlivým kategoriím příspěvků. Sekce pro obecní akce bude obsahovat možnosti hodnocení pro již proběhlé akce v číselném rozsahu 1 až 5 a akce bude možno filtrovat pomocí datového rozsahu, který si uživatel stanoví. Filtrování bude obsahovat i sekce pro nabídky a poptávky, kde si bude moci uživatel vyfiltrovat příspěvky podle zvolené kategorie.

5.1.1 FUNKČNÍ POŽADAVKY

req Funkční požadavky

Správa příspěvků	Správa komentářů	Správa uživatelů
<input checked="" type="checkbox"/> + R001: Systém musí umožnit vytvoření příspěvku.	<input checked="" type="checkbox"/> + R011: Systém musí umožnit vytvoření komentáře.	<input checked="" type="checkbox"/> + R014: Systém musí umožňovat vytvářet nové uživatele.
<input checked="" type="checkbox"/> + R002: Systém bude podporovat rozdělení příspěvků do kategorií.	<input checked="" type="checkbox"/> + R012: Systém musí umožnit smazání komentáře.	<input checked="" type="checkbox"/> + R015: Systém musí umožňovat změnu hesla uživatele.
<input checked="" type="checkbox"/> + R003: Systém musí umožnit editaci příspěvku.	<input checked="" type="checkbox"/> + R013: Systém musí umožnit nahlášení komentáře.	<input checked="" type="checkbox"/> + R016: Systém bude podporovat rozdělení uživatelů podle rolí.
<input checked="" type="checkbox"/> + R004: Systém musí umožnit smazání příspěvku.		
<input checked="" type="checkbox"/> + R005: Systém musí umožnit nahlášení příspěvku.		
<input checked="" type="checkbox"/> + R006: Systém musí obsahovat schvalování příspěvků.		
<input checked="" type="checkbox"/> + R007: Systém musí umožnit vytvoření hodnocení v sekci událostí.		
<input checked="" type="checkbox"/> + R008: Systém musí umožňovat třídění událostí podle datového rozsahu.		
<input checked="" type="checkbox"/> + R009: Systém bude podporovat rozdělení nabídek/poptávek do kategorií.		
<input checked="" type="checkbox"/> + R010: Systém musí umožňovat třídění nabídek/poptávek podle kategorií.		

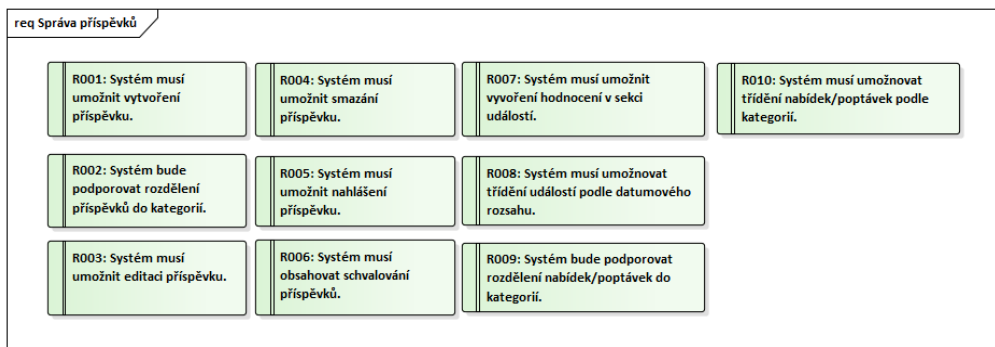
Obrázek 30: Funkční požadavky

R001: Každý registrovaný uživatel bude moci pomocí formuláře vytvářet příspěvek. Parametry formuláře budou záviset kategorii příspěvku.

R002: Každý z příspěvků bude přiřazen do jedné z kategorií. Výchozí kategorie budou Aktuality, Diskuze občanů, Nabídky/poptávky, Obecní akce.

R003: Každý registrovaný uživatel bude moci jím vytvořený příspěvek upravit pomocí formuláře s původními daty příspěvku.

R004: Každý registrovaný uživatel bude moci jím vytvořený příspěvek smazat. Správce bude moci smazat jakýkoliv příspěvek.



Obrázek 31: Správa požadavků

R005: Každý registrovaný uživatel bude moci pomocí tlačítka nahlásit jakýkoliv příspěvek, který se mu bude zdát nevhodný.

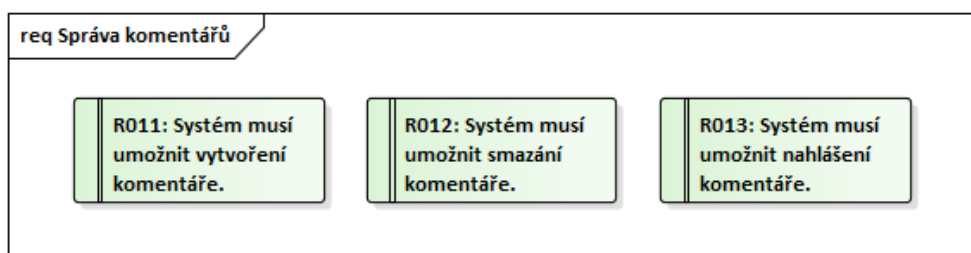
R006: Každý nově vytvořený příspěvek a příspěvky po editaci musejí být schváleny ze strany správce v příslušné sekci.

R007: Každý registrovaný uživatel bude moci hodnotit již proběhlou událost v sekci *Obecní akce*.

R008: Příspěvky v sekci *Obecní akce* lze třídit podle data konání pomocí zadání datového rozsahu.

R009: Každý příspěvek v sekci *Nabídky/poptávky* bude přiřazen do jedné z kategorií. Výchozí nabídka kategorií bude zahrnovat: Nabídky, Poptávky, Služby, Výpůjčky.

R010: Příspěvky v sekci *Nabídky/poptávky* lze třídit podle zvolené kategorie.

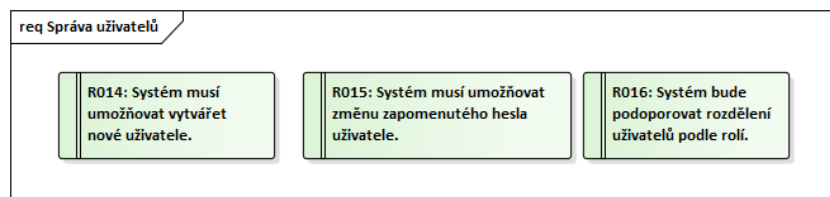


Obrázek 32: Správa komentářů

R011: Každý registrovaný uživatel bude moci pomocí formuláře vytvářet komentář ke zvolenému příspěvku.

R012: Každý registrovaný uživatel bude moci jím vytvořený komentář smazat. Správce bude moci smazat jakýkoliv komentář.

R013: Každý registrovaný uživatel bude moci pomocí tlačítka nahlásit jakýkoliv komentář, který se mu zdá nevhodný.



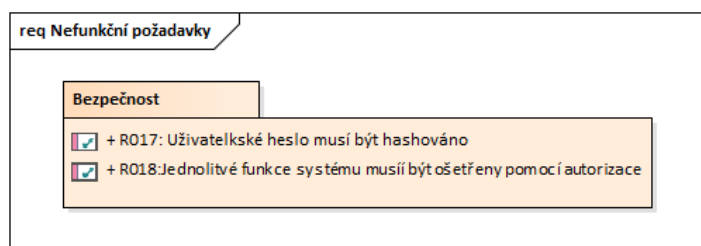
Obrázek 33: Správa uživatelů

R014: Každý neregistrovaný uživatel si bude moci pomocí registračního formuláře vytvořit účet. Registrační formulář bude obsahovat jméno, příjmení, login, heslo a e-mail.

R015: Každý registrovaný uživatel bude moci změnit své zapomenuté uživatelské heslo.

R016: Systém musí obsahovat rozdělení uživatelů do jednotlivých uživatelských rolí. Výchozí role budou: Admin, User.

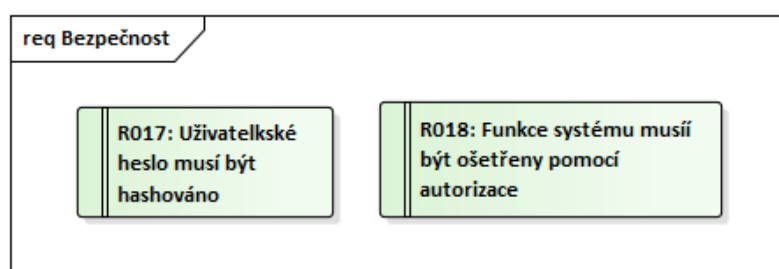
5.1.2 NEFUNKČNÍ POŽADAVKY



Obrázek 34: Nefunkční požadavky

R017: Uživatelské heslo musí být hashováno před jeho uložením do databáze.

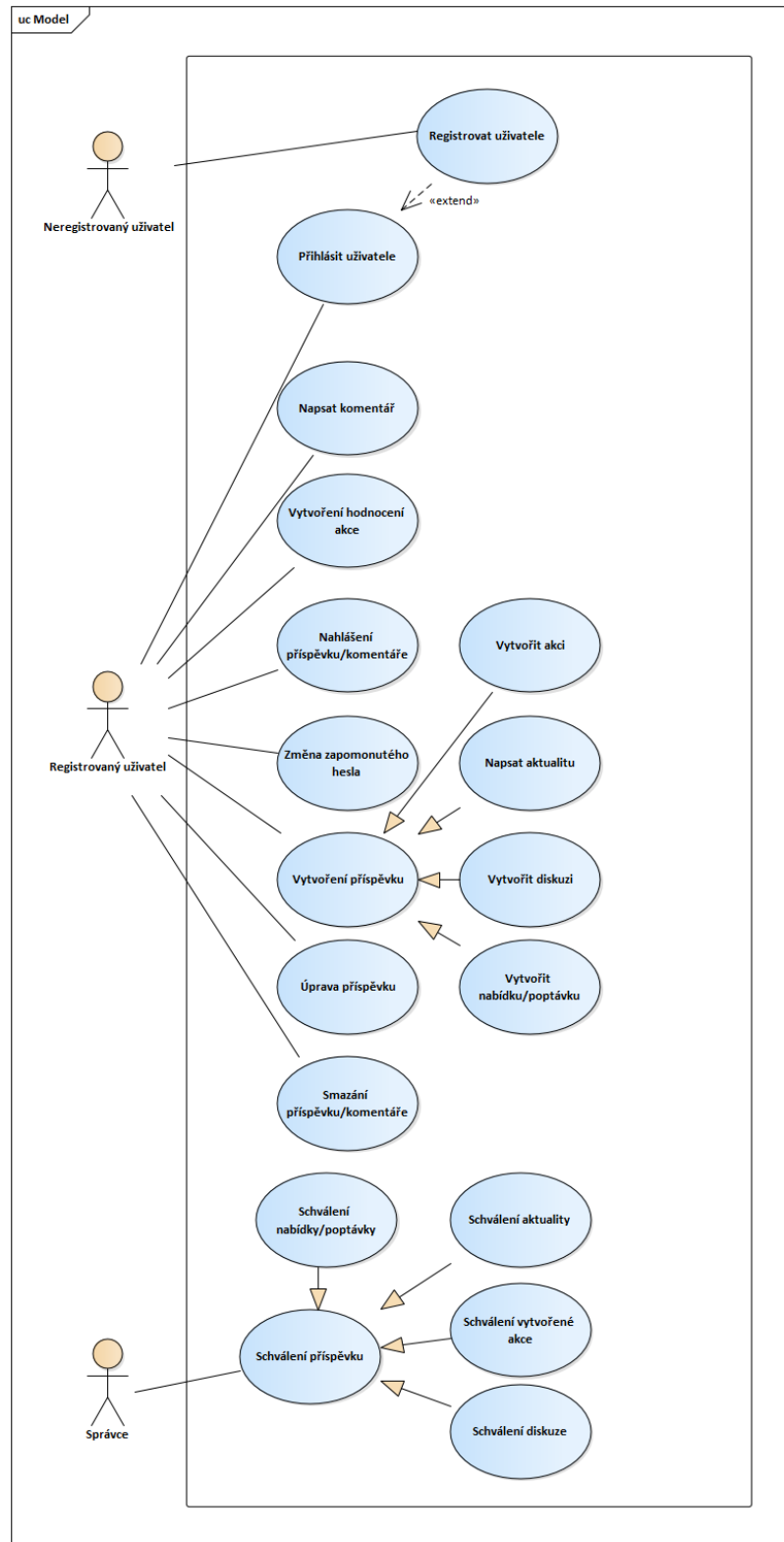
R018: Jednotlivé funkce systému musejí být ošetřeny podle uživatelských rolí, přičemž funkce na úpravu příspěvků musejí být omezeny pouze na autora a správce systému.



Obrázek 35: Bezpečnost

5.1.3 USE CASE MODEL

Use Case diagram popisuje základní funkce webové aplikace. V aplikaci jsou tři základní aktéři: *registrovaný uživatel*, *neregistrovaný uživatel* a *správce*.



Obrázek 36: Use Case model aplikace

Scénář: Registrace uživatele

Tento Use Case popisuje funkci pro registraci nového uživatele do aplikace. Neregistrovaný uživatel musí vyplnit všechny položky v registračním formuláři.

Jméno scénáře	Registrace uživatele
Zúčastnění aktéři	Neregistrovaný uživatel
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když se chce neregistrovaný uživatel zaregistrovat. 2. Uživatel klikne na tlačítko "Registrace". 3. Uživatel vyplní registrační formulář. 4. Uživatel potvrdí registraci. 5. Systém zkontroluje správnost zadaných údajů. 6. Systém uloží uživatele do databáze.
Vedlejší scénáře	<p>5a. Data byla zadána špatně.</p> <p>5b. Údaje jsou již použity pro jiný účet.</p>

Tabulka 1: Scénář – Registrace uživatele

Alternativní scénář: Registrace uživatele – Data byla zadána špatně

Tento alternativní scénář popisuje případ, kdy uživatel špatně vyplní registrační formulář.

Jméno scénáře	Registrace uživatele – Data byla zadána špatně
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 2: Alternativní scénář registrace uživatele

Alternativní scénář: Registrace uživatele – Údaje jsou již použity pro jiný účet

Tento alternativní scénář popisuje případ, kdy uživatel do registračního formuláře zadá data, která již byla použita při registraci jiného účtu.

Jméno scénáře	Registrace uživatele – Údaje jsou již použity pro jiný účet
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o duplicitě zadaných údajů.

Tabulka 3: Alternativní scénář registrace uživatele

Scénář: Přihlášení uživatele

Tento Use Case popisuje funkci pro přihlášení do aplikace. Uživatel je povinen zadat přihlašovací údaje do přihlašovacího formuláře.

Jméno scénáře	Přihlášení uživatele
Zúčastnění aktéři	Registrovaný uživatel, správce
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když se chce registrovaný uživatel nebo správce přihlásit do aplikace. 2. Registrovaný uživatel/správce kliknou na tlačítko "Login". 3. Registrovaný uživatel/správce vyplní přihlašovací jméno a heslo. 4. Systém ověří zadané údaje. 5. Registrovaný uživatel/správce je přihlášen.
Vedlejší scénáře	4a. Registrovaný uživatel/správce zadal špatné údaje

Tabulka 4: Scénář – Přihlášení uživatele

Alternativní scénář: Přihlášení uživatele – Registrovaný uživatel/správce zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy uživatel do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Přihlášení uživatele – Registrovaný uživatel/správce zadal špatné údaje
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 5: Alternativní scénář přihlášení uživatele

Scénář: Vytvoření komentáře

Tento Use Case popisuje funkci pro vytvoření komentáře ke zvolenému příspěvku.

Jméno scénáře	Vytvoření komentáře
Zúčastnění aktéři	Registrovaný uživatel, správce
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když registrovaný uživatel chce přidat komentář pod příspěvek. 2. Registrovaný uživatel se přihlásí. 3. Registrovaný uživatel klikne pod příspěvkem na „Komentovat“. 4. Systém zobrazí formulář pro komentář. 5. Registrovaný uživatel vyplní pole pro text komentáře a potvrdí tlačítkem „Komentovat“. 6. Systém zkontroluje zadané údaje. 7. Systém uloží komentář. 8. Systém zobrazí komentář
Vedlejší scénáře	<p>2a. Registrovaný uživatel zadal špatné údaje.</p> <p>6a. Registrovaný uživatel zanechá pole pro text prázdné.</p> <p>6b. Registrovaný uživatel překročí povolený počet znaků.</p>

Tabulka 6: Scénář – Vytvoření komentáře

Alternativní scénář: Vytvoření komentáře – Registrovaný uživatel zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy uživatel do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Vytvoření komentáře – Registrovaný uživatel zadal špatné údaje
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 7: Alternativní scénář vytvoření komentáře

Alternativní scénář: Vytvoření komentáře – Registrovaný uživatel zanechá pole pro text prázdné

Tento alternativní scénář popisuje případ, kdy uživatel do pole pro text komentáře nezadá žádný znak.

Jméno scénáře	Vytvoření komentáře – Registrovaný uživatel zanechá pole pro text prázdné
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o minimálním počtu znaků pro komentář.

Tabulka 8: Alternativní scénář vytvoření komentáře

Alternativní scénář: Vytvoření komentáře – Registrovaný uživatel překročí povolený počet znaků

Tento alternativní scénář popisuje případ, kdy uživatel překročí povolený počet znaků pro komentář.

Jméno scénáře	Vytvoření komentáře – Registrovaný uživatel překročí povolený počet znaků
Alternativní scénář	1. Systém vyhodí chybovou hlášku a informuje uživatele o maximálním počtu znaků pro komentář.

Tabulka 9: Alternativní scénář vytvoření komentáře

Scénář: Vytvoření hodnocení akce

Tento Use Case popisuje funkci pro vytvoření hodnocení již proběhlé akce uživatelem.

Jméno scénáře	Vytvoření hodnocení akce
Zúčastnění aktéři	Registrovaný uživatel
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když registrovaný uživatel chce přidat hodnocení k proběhlé akci. 2. Registrovaný uživatel se přihlásí. 3. Registrovaný uživatel přejde do sekce "Akce". 4. Registrovaný uživatel pod proběhlou akcí klikne na tlačítko "Hodnotit". 5. Systém zobrazí formulář pro hodnocení. 6. Registrovaný uživatel zadá číselné hodnocení v rozmezí 0-5. 7. Systém zkontroluje zadaná data. 8. Systém uloží hodnocení do databáze.
Vedlejší scénáře	<p>2a. Registrovaný uživatel zadal špatné údaje.</p> <p>7a. Registrovaný uživatel již akci hodnotil.</p>

Tabulka 10: Scénář – Vytvoření hodnocení akce

Alternativní scénář: Vytvoření hodnocení akce – Registrovaný uživatel zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy uživatel do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Vytvoření hodnocení akce – Registrovaný uživatel zadal špatné údaje
Alternativní scénář	1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 11: Alternativní scénář vytvoření hodnocení akce

Alternativní scénář: Vytvoření hodnocení akce – Registrovaný uživatel již akci hodnotil

Tento alternativní scénář popisuje případ, kdy uživatel chce vytvořit hodnocení pro již hodnocenou akci.

Jméno scénáře	Vytvoření hodnocení akce – Registrovaný uživatel již akci hodnotil
Alternativní scénář	1. Systém vyhodí chybovou hlášku a informuje uživatele o duplicitním hodnocení.

Tabulka 12: Alternativní scénář vytvoření hodnocení akce

Scénář: Nahlášení příspěvku/komentáře

Tento Use Case popisuje funkci pro nahlášení nevhodného příspěvku nebo komentáře.

Jméno scénáře	Nahlášení příspěvku/komentáře
Zúčastnění aktéři	Registrovaný uživatel
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když registrovaný uživatel chce nahlásit nevhodný příspěvek nebo komentář. 2. Registrovaný uživatel se přihlásí. 3. Registrovaný uživatel přejde do sekce s příspěvky. 4. Registrovaný uživatel klikne na tlačítko "Nahlásit" u příspěvku nebo komentáře. 5. Systém pošle zprávu správci s requestem kontroly příspěvku/komentáře.
Vedlejší scénáře	2a. Registrovaný uživatel zadal špatné údaje.

Tabulka 13: Scénář – Nahlášení příspěvku/komentáře

Alternativní scénář: Nahlášení příspěvku/komentáře – Registrovaný uživatel zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy uživatel do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Nahlášení příspěvku/komentáře – Registrovaný uživatel zadal špatné údaje
Alternativní scénář	1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 14: Alternativní scénář nahlášení příspěvku/komentáře

Scénář: Změna zapomenutého hesla

Tento Use Case popisuje funkci pro změnu zapomenutého hesla.

Jméno scénáře	Změna zapomenutého hesla
Zúčastnění aktéři	Registrovaný uživatel, správce
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když registrovaný uživatel nebo správce chce změnit zapomenuté heslo. 2. Registrovaný uživatel/správce zvolí "Zapomenuté heslo?". 3. Registrovaný uživatel/správce zadá svůj e-mail. 4. Systém ověří zadaný e-mail v databázi. 5. Systém odešle e-mail s odkazem pro změnu hesla. 6. Registrovaný uživatel/správce zadá nové heslo. 7. Systém ověří nové heslo. 8. Registrovaný uživatel/správce se přihlásí do systému.
Vedlejší scénáře	<p>4a. E-mail není v databázi.</p> <p>7a. Heslo nespĺňuje minimální požadavky.</p>

Tabulka 15: Scénář – Změna zapomenutého hesla

Alternativní scénář: Změna zapomenutého hesla – E-mail není v databázi

Tento alternativní scénář popisuje případ, kdy uživatel do formuláře pro změnu hesla zadá e-mail, který není spjatý s žádným existujícím uživatelským účtem.

Jméno scénáře	Změna zapomenutého hesla – E-mail není v databázi
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o zadání špatné e-mailové adresy.

Tabulka 16: Alternativní scénář změny zapomenutého hesla

Alternativní scénář: Změna zapomenutého hesla – Heslo nespĺňuje minimální požadavky

Tento alternativní scénář popisuje případ, kdy uživatel do formuláře pro nové heslo zadá heslo, které nespĺňuje minimální požadavky ze strany aplikace.

Jméno scénáře	Změna zapomenutého hesla – Heslo nespĺňuje minimální požadavky
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o minimálních požadavcích na heslo.

Tabulka 17: Alternativní scénář změny zapomenutého hesla

Scénář: Vytvoření příspěvku

Tento Use Case popisuje funkci pro vytvoření příspěvku.

Jméno scénáře	Vytvoření příspěvku
Zúčastnění aktéři	Registrovaný uživatel, správce
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když registrovaný uživatel nebo správce chce vytvořit příspěvek. 2. Registrovaný uživatel/správce se přihlásí. 3. Registrovaný uživatel/správce přejde do příslušné sekce, kde chce vytvořit příspěvek. 4. Registrovaný uživatel/správce klikne na příslušné tlačítko pro vytvoření příspěvku. 5. Systém zobrazí formulář pro vytvoření příspěvku. 6. Registrovaný uživatel/správce vyplní daný formulář. 7. Systém zkontroluje správnost zadaných dat. 8. Systém uloží data. 9. Správce schválí příspěvek. 10. Systém zobrazí příspěvek na stránce.
Vedlejší scénáře	<p>2a. Registrovaný uživatel/správce zadal špatně údaje.</p> <p>7a. Data byla zadána špatně nebo vůbec.</p> <p>9a. Správce zamítne schválení.</p>

Tabulka 18: Scénář – Vytvoření příspěvku

Alternativní scénář: Vytvoření příspěvku – Registrovaný uživatel/správce zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy uživatel do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Vytvoření příspěvku – Registrovaný uživatel/správce zadal špatné údaje
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 19: Alternativní scénář vytvoření příspěvku

Alternativní scénář: Vytvoření příspěvku – Data byla zadána špatně nebo vůbec

Tento alternativní scénář popisuje případ, kdy uživatel do formuláře pro vytvoření příspěvku nezadá jeden z povinných parametrů nebo nesplní podmínky formátu pro dané parametry.

Jméno scénáře	Vytvoření příspěvku – Data byla zadána špatně nebo vůbec
Alternativní scénář	1. Systém vyhodí chybovou hlášku a informuje uživatele o chybějícím povinném parametru nebo špatně zadaném formátu parametru.

Tabulka 20: Alternativní scénář vytvoření příspěvku

Alternativní scénář: Vytvoření příspěvku – Správce neschválí příspěvek

Tento alternativní scénář popisuje případ, kdy vytvořený příspěvek není schválen administrátorem ke zveřejnění.

Jméno scénáře	Vytvoření příspěvku – Správce neschválí příspěvek
Alternativní scénář	1. Systém zašle autorovi příspěvku e-mail s informací o zamítnutí příspěvku.

Tabulka 21: Alternativní scénář vytvoření příspěvku

Scénář: Úprava příspěvku

Tento Use Case popisuje funkci pro úpravu již vytvořených příspěvků.

Jméno scénáře	Úprava příspěvku
Zúčastnění aktéři	Registrovaný uživatel, správce
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když registrovaný uživatel nebo správce chce upravit příspěvek. 2. Registrovaný uživatel/správce se přihlásí. 3. Registrovaný uživatel/správce přejde do sekce s příspěvky. 4. Registrovaný uživatel/správce klikne na tlačítko "Upravit" u příspěvku. 5. Systém zobrazí formulář pro úpravu příspěvku. 6. Registrovaný uživatel/správce upraví příspěvek a potvrdí změny. 7. Systém zkontroluje správnost zadaných údajů. 8. Systém uloží data a pošle příspěvek na schválení. 9. Správce schválí příspěvek. 10. Systém zobrazí příspěvek na stránce.
Vedlejší scénáře	<p>2a. Registrovaný uživatel/správce zadal špatné údaje.</p> <p>7a. Data byla zadána špatně nebo vůbec.</p> <p>9a. Správce zamítne schválení.</p>

Tabulka 22: Scénář – Úprava příspěvku

Alternativní scénář: Úprava příspěvku – Registrovaný uživatel/správce zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy uživatel do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Úprava příspěvku – Registrovaný uživatel/správce zadal špatné údaje
Alternativní scénář	1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 23: Alternativní scénář úpravy příspěvku

Alternativní scénář: Úprava příspěvku – Data byla zadána špatně nebo vůbec

Tento alternativní scénář popisuje případ, kdy uživatel do formuláře pro úpravu příspěvku nezadá jeden z povinných parametrů nebo nesplní podmínky formátu pro dané parametry.

Jméno scénáře	Úprava příspěvku – Data byla zadána špatně nebo vůbec
Alternativní scénář	1. Systém vyhodí chybovou hlášku a informuje uživatele o chybějícím povinném parametru nebo špatně zadaném formátu parametru.

Tabulka 24: Alternativní scénář úpravy příspěvku

Alternativní scénář: Úprava příspěvku – Správce neschválí příspěvek

Tento alternativní scénář popisuje případ, kdy upravený příspěvek není schválen administrátorem ke zveřejnění.

Jméno scénáře	Úprava příspěvku – Správce neschválí příspěvek
Alternativní scénář	1. Systém zašle autorovi příspěvku e-mail s informací o zamítnutí příspěvku.

Tabulka 25: Alternativní scénář úpravy příspěvku

Scénář: Smazání příspěvku/komentáře

Tento Use Case popisuje funkci pro smazání příspěvku nebo komentáře.

Jméno scénáře	Smazání příspěvku/komentáře
Zúčastnění aktéři	Registrovaný uživatel, správce
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když registrovaný uživatel nebo správce chce smazat příspěvek. 2. Registrovaný uživatel/správce se přihlásí. 3. Registrovaný uživatel/správce přejde do sekce s příspěvky. 4. Registrovaný uživatel/správce klikne na tlačítko "Smazat" u příspěvku. 5. Systém zobrazí potvrzovací okno. 6. Registrovaný uživatel/správce potvrdí smazání příspěvku. 7. Systém smaže příspěvek.
Vedlejší scénáře	2a. Registrovaný uživatel/správce zadal špatné údaje

Tabulka 26: Scénář – Smazání příspěvku

Alternativní scénář: Smazání příspěvku/komentáře – Registrovaný uživatel/správce zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy uživatel do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Smazání příspěvku/komentáře – Registrovaný uživatel/správce zadal špatné údaje
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje uživatele o chybném zadání údajů.

Tabulka 27: Alternativní scénář smazání příspěvku/komentáře

Scénář: Schválení příspěvku

Tento Use Case popisuje funkci pro schválení všech vytvořených příspěvků od uživatelů.

Jméno scénáře	Schválení příspěvku
Zúčastnění aktéři	Správce
Hlavní scénář	<ol style="list-style-type: none"> 1. Use Case začíná, když správci přijde request na schválení vytvořeného příspěvku od registrovaného uživatele. 2. Správce se přihlásí do systému. 3. Správce přijde do sekce "requestů" od registrovaných uživatelů. 4. Systém správci zobrazí seznam příspěvků čekajících na schválení. 5. Správce rozklikne detail příspěvku. 6. Správce schválí příspěvek. 7. Systém zobrazí příspěvek na stránce.
Vedlejší scénáře	<p>2a. Správce zadal špatné údaje</p> <p>6a. Správce neschválí příspěvek.</p>

Tabulka 28: Scénář – Schválení příspěvku

Alternativní scénář: Schválení příspěvku – Správce zadal špatné údaje

Tento alternativní scénář popisuje případ, kdy správce do přihlašovacího formuláře zadá data, která neodpovídají žádnému z existujících účtů.

Jméno scénáře	Schválení příspěvku – Správce zadal špatné údaje
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém vyhodí chybovou hlášku a informuje správce o chybném zadání údajů.

Tabulka 29: Alternativní scénář schválení příspěvku

Alternativní scénář: Schválení příspěvku – Správce neschválí příspěvek

Tento alternativní scénář popisuje případ, kdy vytvořený nebo upravený příspěvek není schválen administrátorem ke zveřejnění.

Jméno scénáře	Schválení příspěvku – Správce neschválí příspěvek
Alternativní scénář	<ol style="list-style-type: none"> 1. Systém zašle autorovi příspěvku e-mail s informací o zamítnutí příspěvku.

Tabulka 30: Alternativní scénář schválení příspěvku

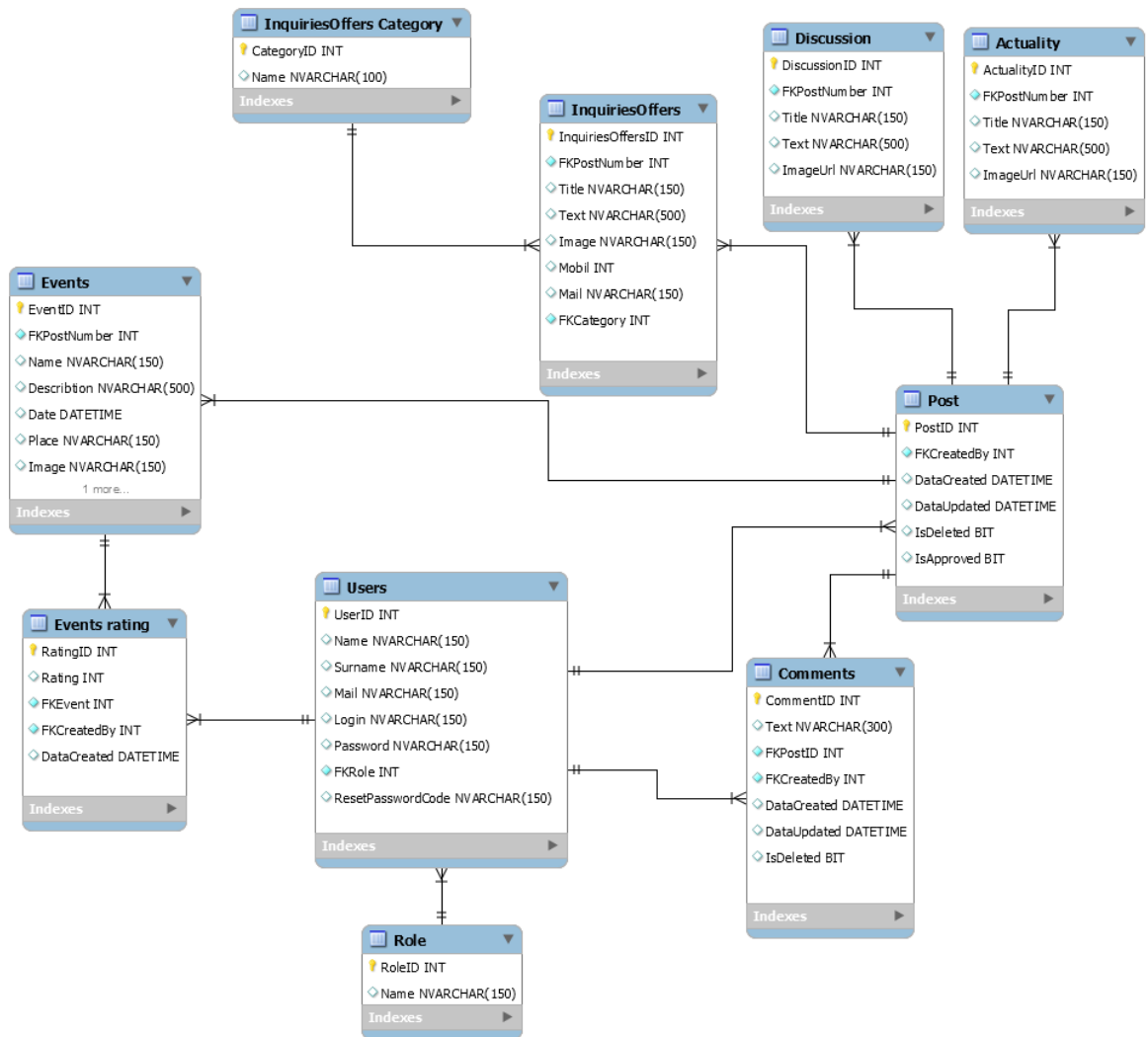
5.2 NÁVRH DATABÁZE

Pro návrh databáze byl zvolen fyzický model, který byl zpracován podle základních požadavků na webovou aplikaci a její funkce.

Tabulka *Users* slouží k ukládání informací o uživateli. Na tuto tabulku se pojí tabulka *Role*, která obsahuje seznam uživatelských rolí. ID uživatelských rolí je obsaženo v tabulce *Users* jako cizí klíč.

Tabulka *Post* sdružuje všechny společné parametry tabulek *Actuality*, *Discussion*, *InquiriesOffers* a *Events*, které obsahují data o jednotlivých příspěvcích rozdělených podle druhu příspěvku. Na tabulku *InquiriesOffers* se navíc pojí tabulka *InquiriesOffers Category*, která obsahuje seznam kategorií pro tyto druhy příspěvku. Tabulka *Events rating* obsahuje hodnocení pro akce uložené v tabulce *Events*. Autoři hodnocení jsou bráni z tabulky *Users* pomocí cizího klíče.

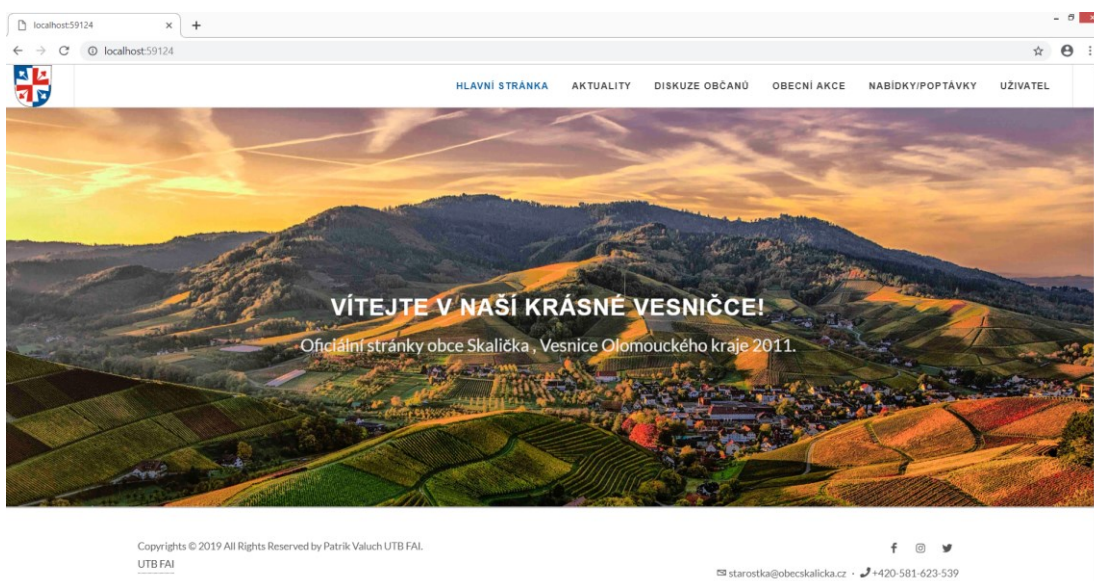
Do tabulky *Comments* jsou ukládány všechny vytvořené komentáře, které jsou k jednotlivým příspěvkům vázány pomocí *PostID* v tabulce *Post*.



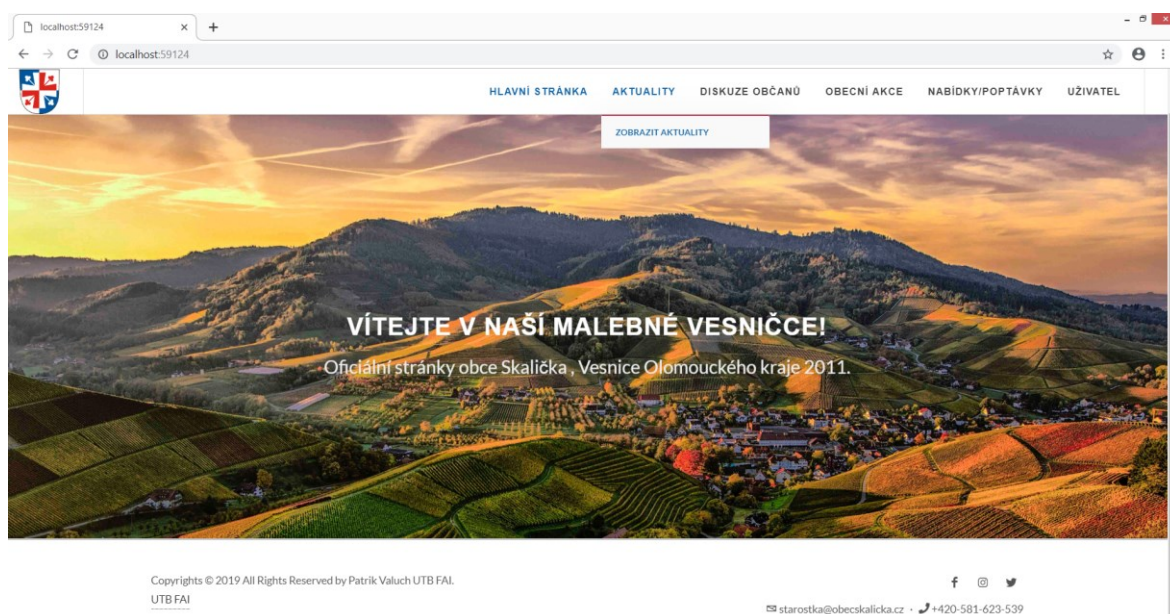
Obrázek 37: Fyzický model databáze aplikace

6 UŽIVATELSKÁ PŘÍRUČKA

Při prvním načtení webové stránky se zobrazí její domovská stránka neboli home page. V levém horním rohu se nachází logo obce, které po kliknutí slouží jako odkaz na hlavní stránku. Hlavička dále obsahuje jednotlivé položky menu, které se rozbíjí po najetí kurzorem myši. Menu obsahuje jednotlivé druhy příspěvků, které lze zobrazit buď kliknutím na samotný název, nebo najetím kurzoru myši a následným kliknutím (například na *Zobrazit aktuality*). Položka *Uživatel* slouží k přihlášení nebo registraci uživatele do aplikace. V zápatí stránky se nachází kontakty na obec a odkazy na její sociální sítě.



Obrázek 38: Homepage aplikace



Obrázek 39: Dropdown menu

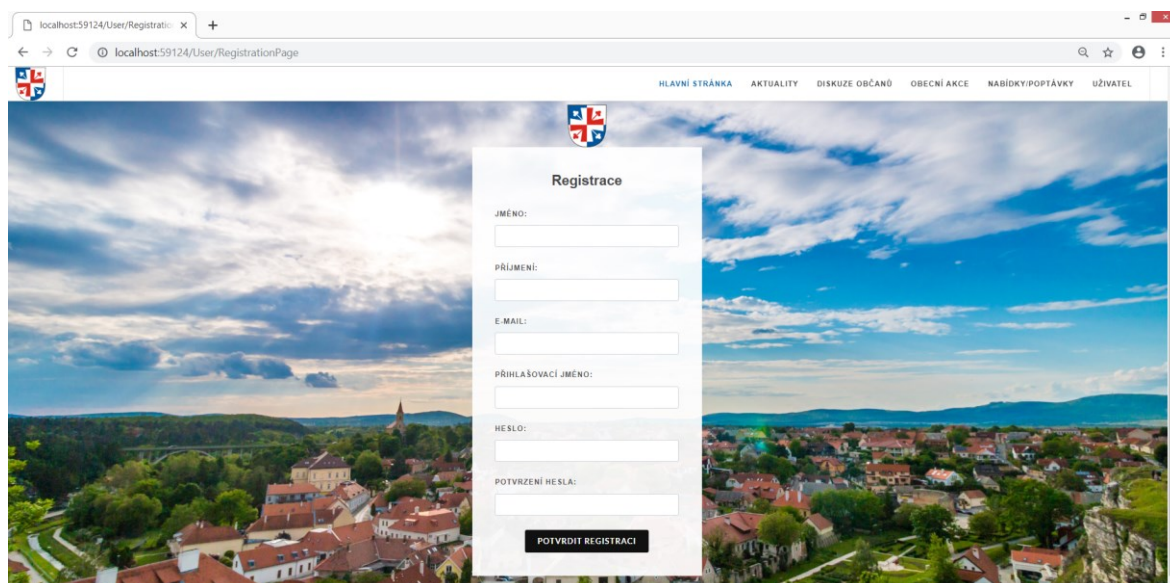
6.1 Z POHLEDU NEREGISTROVANÉHO UŽIVATELE

Neregistrovaný uživatel má v rámci aplikace omezené možnosti – může si sice zobrazit všechny příspěvky spolu s komentáři (v případě událostí i s jejich hodnocením), ale nemá možnost příspěvky vytvářet, komentovat či hodnotit.

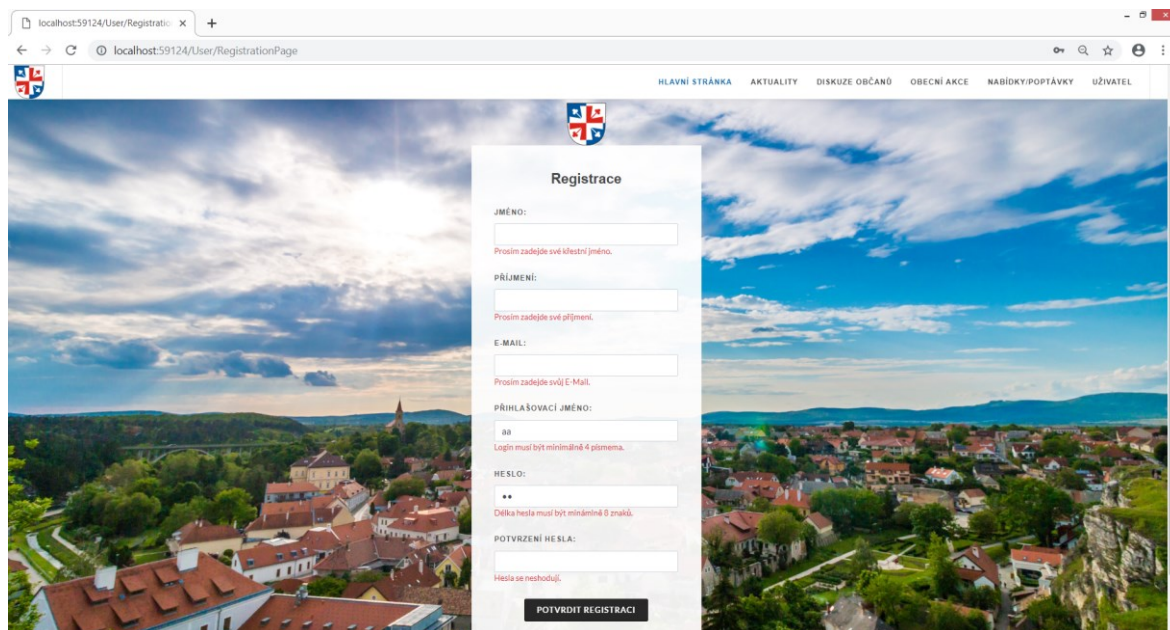
6.1.1 REGISTRACE

Pro vytvoření uživatelského účtu je nutné se nejprve registrovat. Na registrační formulář se uživatel dostane přes položku *Uživatel* v hlavičce stránky s následným výběrem *Registrace*. Uživatel je povinen zadat své jméno, příjmení, e-mail, přihlašovací jméno a heslo, které je nutné pro ověření správnosti zadat ještě jednou.

Při potvrzení registrace probíhá validace zadaných údajů. Žádný z parametrů nemůže být prázdný. U přihlašovacího jména a zadané e-mailové adresy je kontrolováno, zda se zadanými údaji již není registrován jiný uživatel. U e-mailové adresy dochází také ke kontrole správnosti formátu, který odpovídá e-mailovým adresám. Uživatelské jméno musí být minimálně 4 znaky dlouhé, v případě hesla je pak minimální počet znaků nastaven na 8. Po dokončení registrace je uživateli zaslán e-mail s potvrzením úspěšné registrace.



Obrázek 40: Registrace uživatele do aplikace



Obrázek 41: Validace zadaných údajů při potvrzení registrace

6.2 Z POHLEDU REGISTROVANÉHO UŽIVATELE

Registrovaný uživatel má kromě zobrazení obsahu možnost obsah vytvářet. Všechny tyto funkce mu jsou zpřístupněny po úspěšném přihlášení do aplikace.

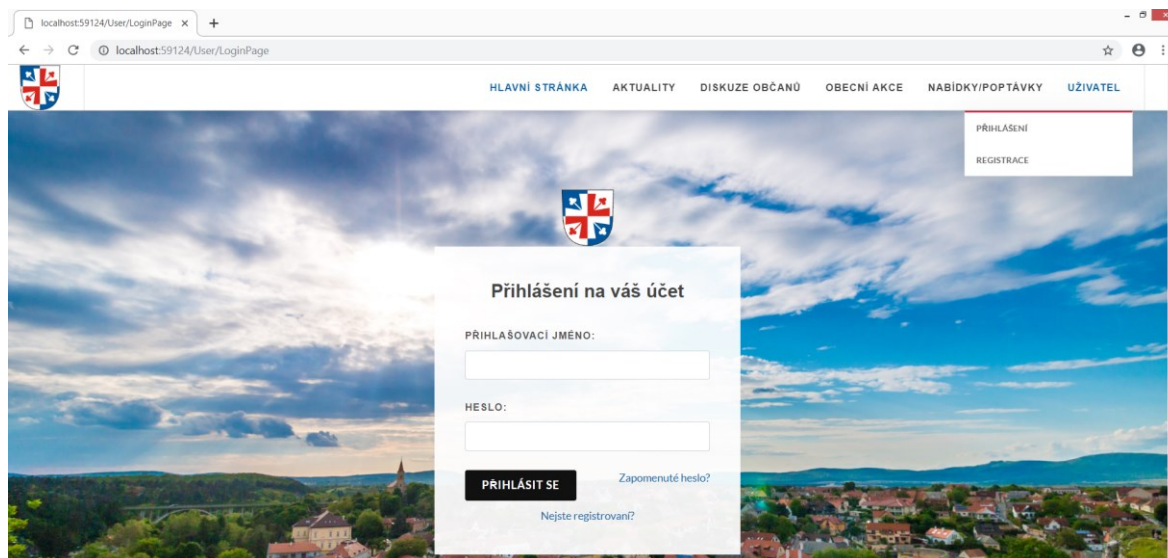
6.2.1 PŘIHLÁŠENÍ DO APLIKACE

Do přihlašovací sekce se uživatel dostane přes položku *Uživatel* v hlavičce stránky s následným výběrem *Přihlášení*. Zde vyplní své přihlašovací údaje a přihlášení potvrdí pomocí tlačítka *Přihlásit se*. Následně systém ověří platnost zadaných údajů. V případě zadání chybných údajů je uživatel upozorněn, že se jedná o špatné údaje nebo neexistujícího uživatele. Po dokončení přihlášení je uživatele přesměrován na domovskou stránku a je informován pomocí notifikace, že byl úspěšně přihlášen.



Obrázek 42: Notifikace úspěšného přihlášení do aplikace

V přihlašovacím formuláři se nachází odkaz na registrační sekci v případě, že by se uživatel dostal do přihlašovací sekce, aniž by byl registrován. Dále se v této části nachází odkaz pro změnu hesla v případě jeho zapomnutí.



Obrázek 43: Přihlašovací formulář do aplikace

6.2.2 ZMĚNA HESLA

Uživatel má možnost si své heslo změnit, ať z důvodu zapomenutí nebo vlastní in-tence. Pro změnu hesla je nutné kliknout na odkaz *Zapomenuté heslo?* v přihlašovacím formuláři (viz. obrázek č. 43). Následně dojde k vyvolání *modal* elementu, do kterého uživatel zadá svou e-mailovou adresu uvedenou při registraci do aplikace.

Obrázek 44: Modal element pro zadání e-mailu

V případě, že daný e-mail není spojen s žádným uživatelským účtem je uživatel upozorněn pomocí notifikace. V opačném případě je informován o odeslání odkazu pro změnu hesla na zadaný e-mail.

S odkazem pro změnu hesla je automaticky vygenerován tzv. *ResetPasswordCode*, který zabezpečuje, že daný odkaz bude použit pouze jednou. Jeho hodnota je vynulována

po úspěšné změně hesla. Ve formuláři pro změnu hesla je opět ověřována minimální délka hesla.



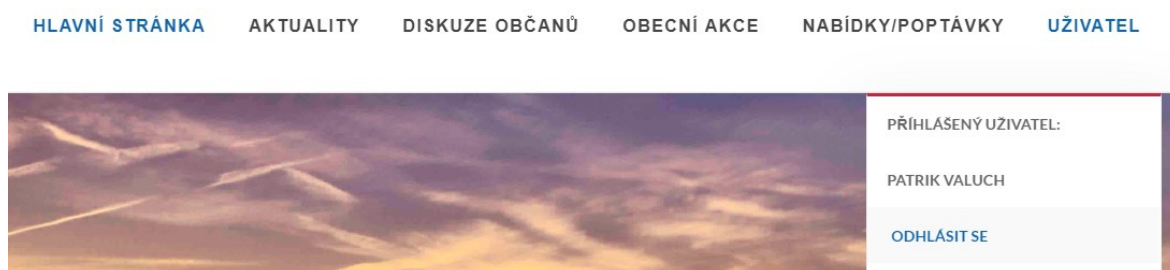
Obrázek 45: E-mail s odkazem pro změnu hesla

ResetPasswordCode
NULL
NULL
NULL
NULL
NULL
NULL
cf9cb2d1-7a97-4e4e-ba6a-88b4636f5e8e

Obrázek 46: Vygenerovaný *ResetPasswordCode*

6.2.3 ODHLÁŠENÍ Z APLIKACE

Odhlášení z aplikace lze provést v dropdown menu pod položkou *Uživatel* v hlavním menu hlavičky. Po kliknutí na *Odhlásit se* je uživatel přesměrován na přihlašovací stránku a informován o úspěšném odhlášení pomocí notifikace.

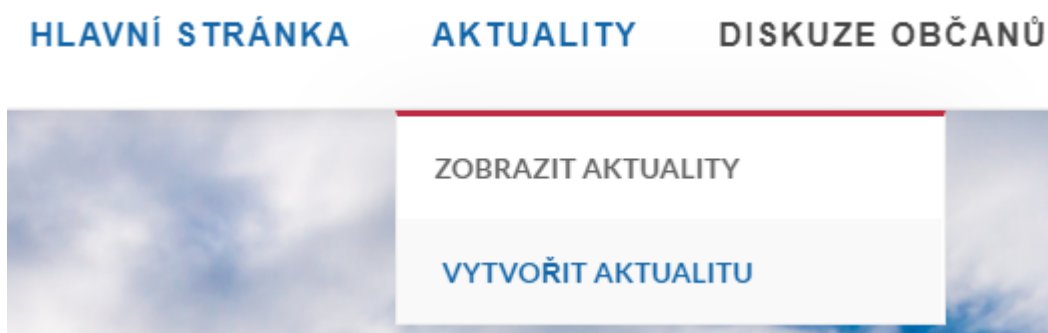


Obrázek 47: Uživatelské menu

6.2.4 VYTVOŘENÍ AKTUALITY A PŘÍSPĚVKU V DISKUZI

Aktuality a Diskuze občanů jsou první dva typy příspěvků v aplikaci. Oba dva druhy mají totožný formulář a postup vytvoření, protože obsahují stejné parametry pro vytvoření.

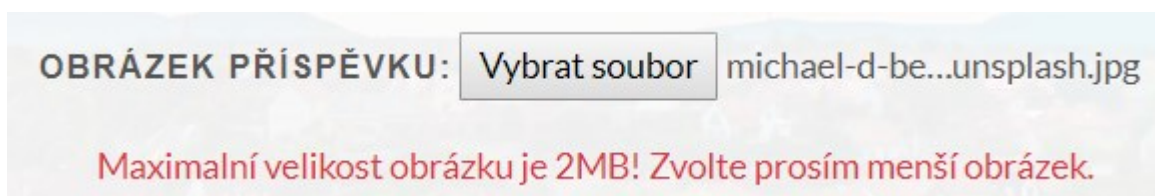
K vytvoření aktuality či příspěvku do diskuze uživatel přistoupí z hlavního menu, kde si zvolí danou kategorii a v dropdown menu zvolí položku vytvoření. Po kliknutí na *Vytvořit aktualitu* dojde k následnému přesměrování na formulář pro vytvoření aktuality. Zde je uživatel povinen vyplnit název příspěvku a zadat text, případně může nahrát obrázek.



Obrázek 48: Vytvoření aktuality

Délka názvu i samotného textu je omezena a v případě jejího přesažení je uživatel informován. Název je omezen na 150 znaků, text na 1000 znaků. Taktéž je stanoven minimální počet znaků, takže ani jedna z položek nemůže zůstat prázdná.

Nahrání obrázku k příspěvku je dobrovolné. V případě, že uživatel obrázek nenahráje, je k příspěvku automaticky přidán defaultní obrázek pro danou sekci. Velikost obrázku je omezena na 2 MB. V případě nahrání obrázku větší velikosti je uživatel o tomto omezení informován a požádán o nahrání menšího obrázku.



Obrázek 49: Validace velikosti nahraného obrázku

Obrázek 51: Formulář pro vytvoření události

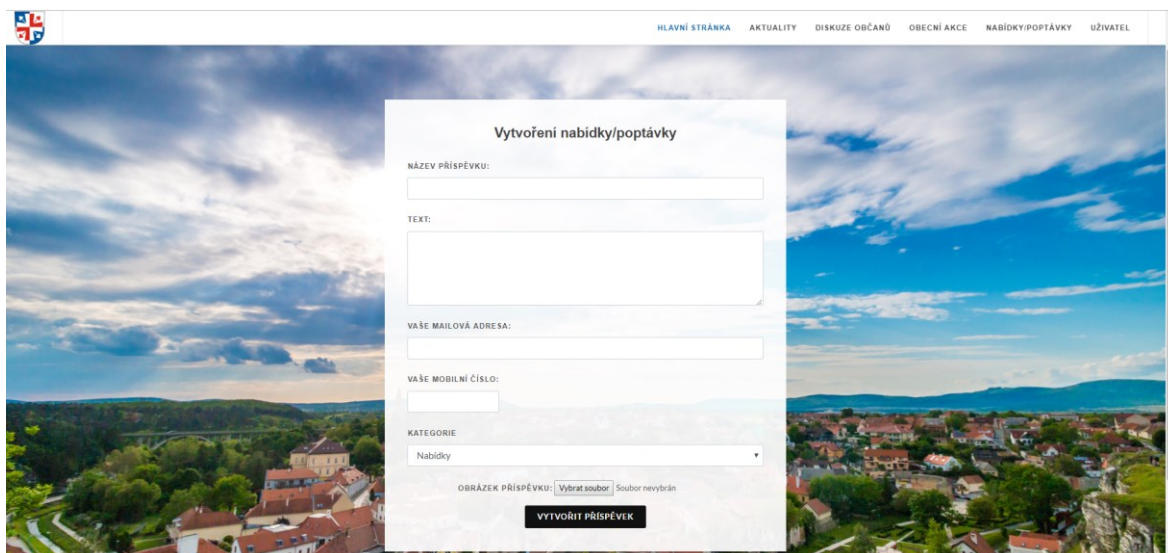
Obrázek 52: *Datepicker* pro zadání data a času události

6.2.6 VYTVOŘENÍ NABÍDKY/POPTÁVKY

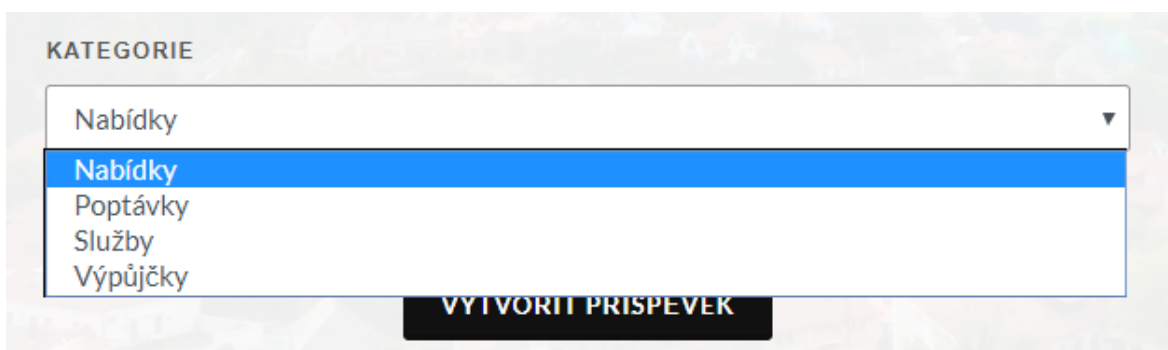
Čtvrtou kategorií příspěvků jsou *Nabídky/poptávky*. K vytvoření příspěvku v této sekci se uživatel dostane rovněž přes dropdown menu *Nabídky/Poptávky* v hlavním menu hlavičky stránky.

Při vytvoření nabídky či poptávky je uživatel povinen zadat název a text příspěvku. Formulář obsahuje kontaktní položky (e-mail a mobilní číslo) sloužící ke kontaktování autora příspěvku. Oba kontaktní parametry jsou povinné a je nutno je zadat pro úspěšné vytvoření příspěvku. Pro nahrání obrázku platí stejná pravidla jako u předešlých typů příspěvků – velikost je omezena na 2 MB a v případě, že obrázek není uživatelem nahrán, je automaticky

použit defaultní obrázek předem určený pro tuto kategorii. Ve formuláři opět dochází ke kontrole počtu znaků a u mobilního čísla a e-mailu je kontrolována správnost formátu. V závěru je uživatel povinen vybrat jednu z kategorií, do které spadá jím vytvořený příspěvek.



Obrázek 53: Formulář pro vytvoření nabídky/poptávky



Obrázek 54: Zvolení kategorie při vytváření nabídky/poptávky

6.2.7 VYTVOŘENÍ KOMENTÁŘE

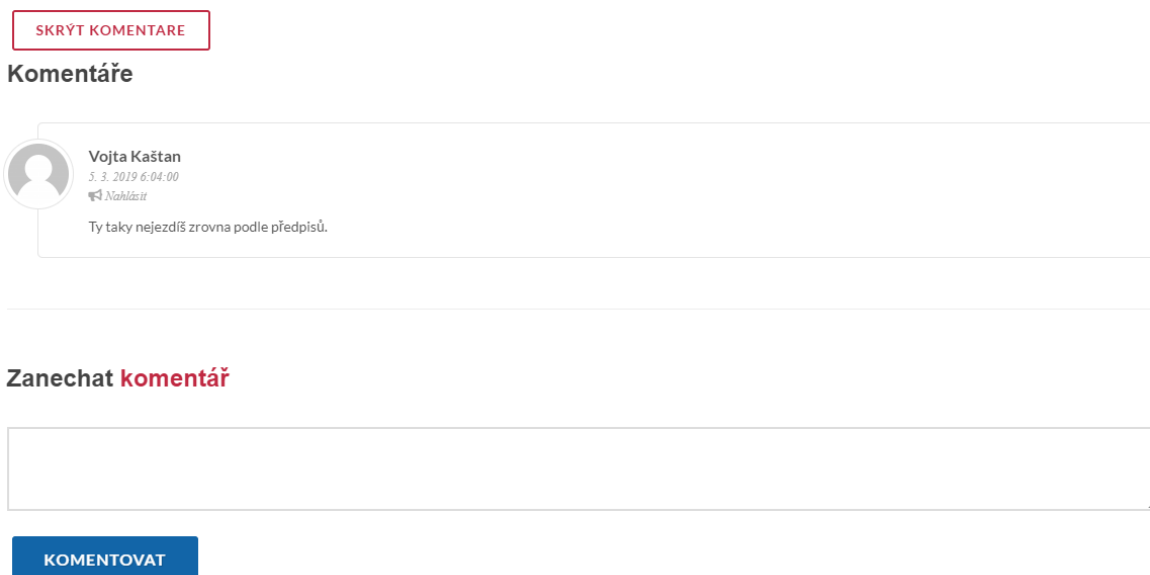
Registrovaný uživatel může k jednotlivým příspěvkům přidávat komentáře. Sekce komentářů se nachází pod každým příspěvkem, ale je defaultně skryta. Uživatel může odkrýt tuto sekci pomocí tlačítka *Zobrazit komentáře* a poté ji opětovně skrýt tlačítkem *Skrýt komentáře*. Délka textu komentáře musí být minimálně 5 znaků a maximální délka komentáře je 300 znaků. Ke komentáři v momentální verzi aplikace nelze nahrát obrázek.

Komentáře nepodléhají schvalování ze strany administrátora a zobrazí se ihned po potvrzení pomocí tlačítka *Komentovat*, přičemž má uživatel možnost svůj komentář kdykoliv smazat. V případě nevhodného komentáře má jakýkoliv uživatel možnost komentář

nahlásit. Pokud se tak stane, administrátor aplikace obdrží e-mail s daným komentářem a výzvou pro jeho kontrolu.



Obrázek 55: Příspěvek s tlačítkem pro zobrazení komentářů



Obrázek 56: Sekce komentářů pod příspěvkem

6.2.8 EDITACE PŘÍSPĚVKU

Uživatel může každý jím vytvořený příspěvek nejen smazat, ale i upravit. Tlačítko *Upravit* se nachází pod názvem příspěvku a lze ho poznat podle ikony klíče. Po kliknutí na toto tlačítko je uživatel dotázán, zda chce opravdu daný příspěvek upravit, a po potvrzení je přesměrován na editační formulář. Pro editaci příspěvku platí stejná validační pravidla jako při vytváření nového příspěvku.

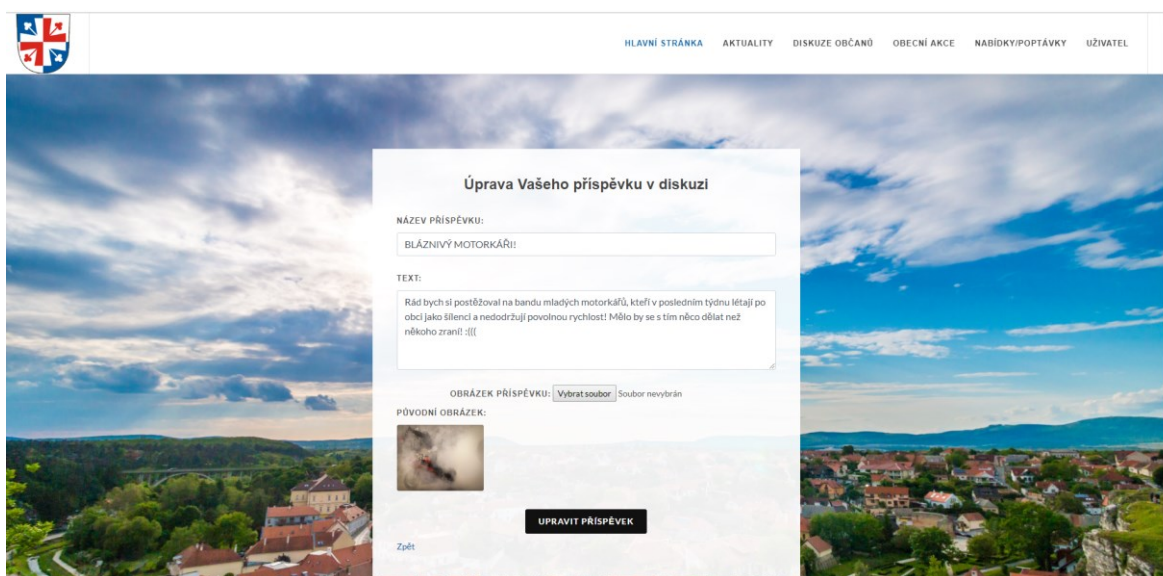
Editační formulář je naplněn původními daty příspěvku, aby uživatel věděl, co příspěvek obsahuje, a je zde zobrazen původní obrázek. Uživatel má možnost nahrát k příspěvku nový obrázek, ovšem pokud této možnosti nevyužije, zůstane u příspěvku obrázek původní.

V levém dolním rohu se nachází tlačítko *Zpět* pro případ, že by se uživatel rozhodl editaci zrušit. Provedené změny uživatel potvrdí pomocí tlačítka *Upravit příspěvek*. Po úpravě příspěvku je nutná opětovná kontrola ze strany administrátora.

BLÁZNIVÝ MOTORKÁŘI!

📅 2. 3. 2019 14:31:20 / 👤 František Oprštlák / 🛠 Upravit 🗑 Smazat

Obrázek 57: Tlačítko pro úpravu a smazání příspěvku



Obrázek 58: Formulář pro úpravu příspěvku v diskuzi

6.2.9 SMAZÁNÍ A NAHLÁŠENÍ PŘÍSPĚVKU

Aplikace obsahuje funkce pro smazání a nahlášení jednotlivých příspěvků. Tlačítka pro jednotlivé funkce jsou obsažena pod názvem příspěvku a jsou prezentována pomocí příslušných ikon – v případě smazání ikona popelnice, pro nahlášení ikona megafonu.

Smazání příspěvku může provést pouze autor daného příspěvku nebo administrátor, který má pravomoc smazat jakýkoliv příspěvek. Smazání je následně nutné potvrdit, aby nedošlo k okamžitému smazání po nechtěném kliknutí na tlačítko.

V případě nevhodného příspěvku může jakýkoliv uživatel pomocí tlačítka *Nahlásit* požádat o jeho kontrolu ze strany administrátora. Po nahlášení je administrátorovi odeslán

e-mail obsahující základní údaje o příspěvku a ID pro jeho rychlou kontrolu. Tuto volbu je nutno potvrdit, aby nedocházelo k chybným nahlášením.

ZTRACENÝ PES

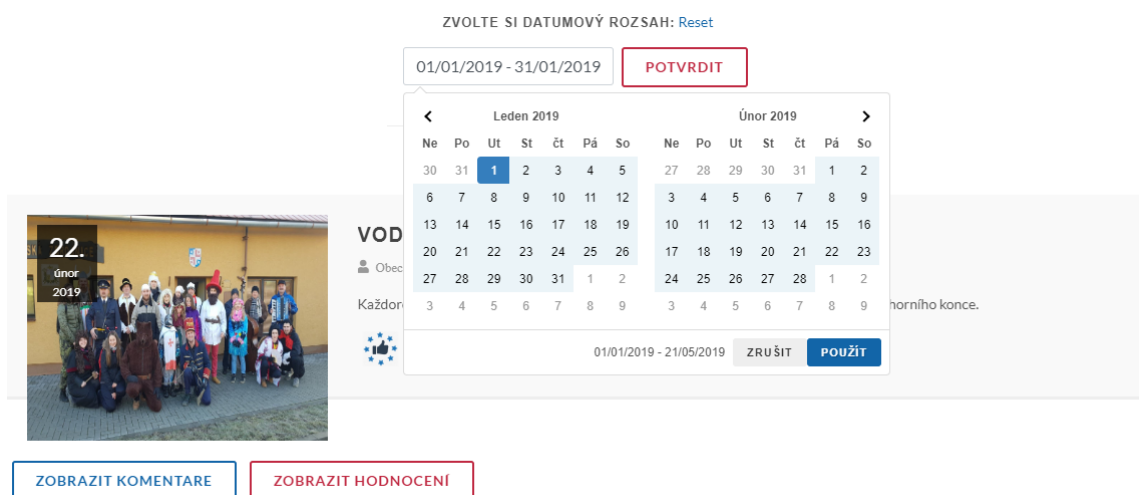
11. 2. 2019 17:46:54 /  Johana Šikulová /  Nahlásit

Obrázek 59: Tlačítko pro nahlášení příspěvku

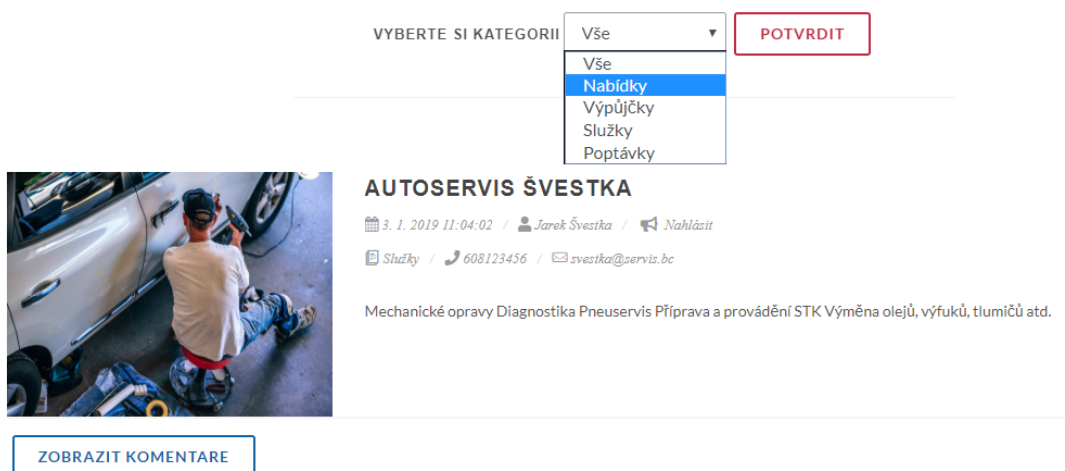
6.2.10 FILTROVÁNÍ PŘÍSPĚVKŮ

Příspěvky v sekci *Obecní akce* a *Nabídky/Poptávky* je možné filtrovat pomocí daných parametrů, zatímco u událostí si uživatel může zvolit datumový rozsah, pro který chce zobrazit akce. Po kliknutí do textboxu s defaultními daty dojde k zobrazení datepickeru, pomocí kterého si uživatel zvolí daný rozsah a volbu potvrdí pomocí tlačítka *Potvrdit*. Nad tímto tlačítkem se nachází tlačítko *Reset* pro zrušení datumového filtru a opětovné zobrazení všech událostí.

Nabídky a poptávky může uživatel vyfiltrovat pomocí daných kategorií, které byly při vytvoření příspěvků v sekci *Nabídky/Poptávky* zvoleny. Pro zpětné zobrazení všech příspěvků se ve filtru nachází také možnost *Vše*.



Obrázek 60: Datepicker pro zvolení datumového rozsahu

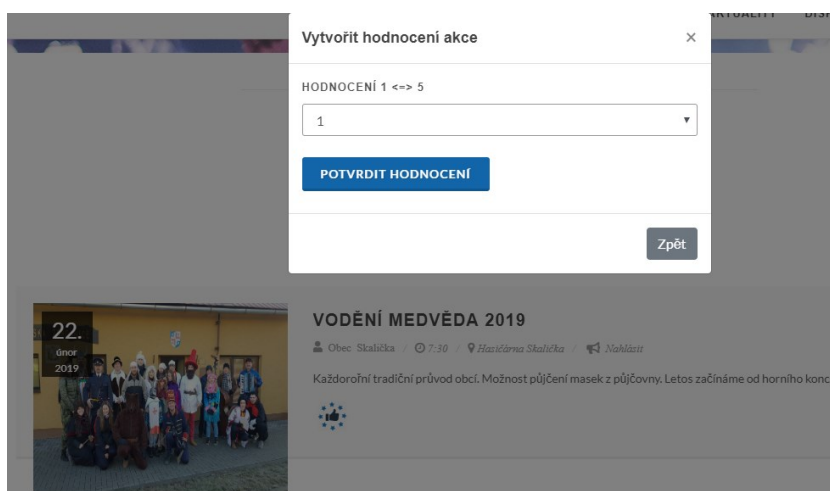


Obrázek 61: Zvolení kategorie ve filtru

6.2.11 VYTVOŘENÍ HODNOCENÍ AKCE

V sekci *Obecní akce* má uživatel možnost akce ohodnotit, přičemž hodnotit lze pouze akce, které již proběhly, a to pouze jednou každým uživatelem. Hodnocení lze zadat pomocí tlačítka s ikonou palce nahoru, které je umístěno pod popisem dané akce. Při stisknutí tohoto tlačítka dojde k vyvolání elementu *Modal* umožňující zadání hodnocení. Uživatel může událost hodnotit číselným hodnocením od 1 do 5, kde se 5 rovná maximálnímu a 1 minimálnímu ohodnocení.

Zadaná hodnocení jsou umístěna v sekci pod událostí, která je stejně jako sekce s komentáři skryta a pro její zobrazení je nutné stisknout tlačítko *Zobrazit hodnocení*. Samotné hodnocení je následně prezentováno pomocí hvězd.



Obrázek 62: Vytvoření hodnocení akce

ZOBRAZIT KOMENTARE ZOBRAZIT HODNOCENÍ

SKRYT HODNOCENÍ

Hodnocení

Jarek Švestka 23. 2. 2019 1:33:46 ★★★★☆	Patrik Valuch 18. 4. 2019 19:14:11 ★★★★☆	František Opršálek 24. 2. 2019 10:12:47 ★★★★☆	Zlata Cibulková 26. 2. 2019 6:19:47 ★★★★★	Vojta Kaštan 23. 4. 2019 23:12:47 ★☆☆☆☆	Petr Panáček 23. 4. 2019 15:13:47 ★★★★☆
Miroslav Trefil 13. 5. 2019 9:03:47 ★★★★☆					

Obrázek 63: Přehled hodnocení od jednotlivých uživatelů

6.2.12 NOTIFIKACE

Notifikace v aplikaci lze rozdělit do dvou kategorií. Zatímco první kategorii tvoří notifikace pomocí e-mailu, do druhé kategorie patří notifikace integrované přímo ve webovém rozhraní.

Prostřednictvím e-mailových zpráv je uživatel informován o úspěšné registraci do aplikace. V případě změny hesla je odkaz pro jeho změnu rovněž zaslán pomocí e-mailové zprávy na e-mail zadaný při registraci uživatele.

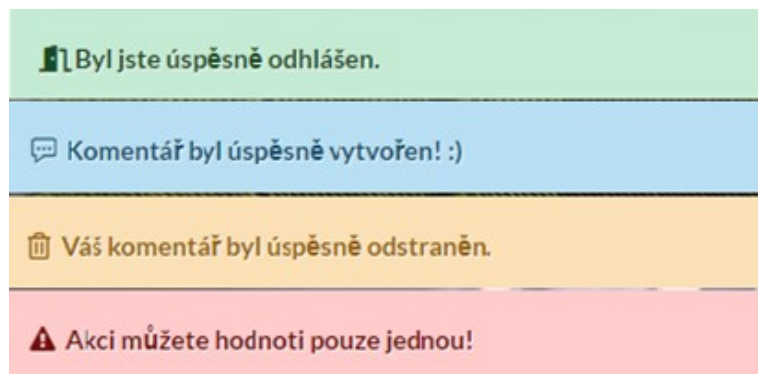
Přímo v aplikaci jsou použity notifikace, které slouží k informování uživatele o úspěšném nebo neúspěšném provedení úkonu, který vykonal. Tyto notifikace slouží jako zpětná vazba ze strany aplikace, aby uživatel věděl o úspěchu jeho požadavku či případné chybě.

Váš účet byl úspěšně vytvořen! ➤

 **Obec Skalička** <obecmail@post.cz>
komu: já ▾

Děkujeme za Vaši registraci na stránkách naší obce! :)

Obrázek 64: Automaticky generovaný e-mail po úspěšné registraci



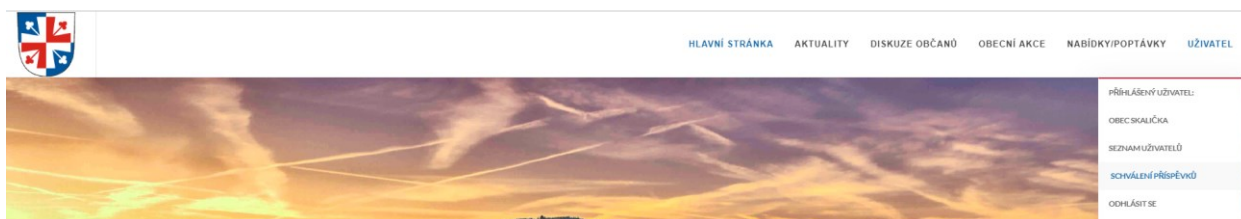
Obrázek 65: Druhy notifikací v aplikaci

6.3 Z POHLEDU ADMINISTRÁTORA

Hlavní funkcí specifickou pro roli administrátora je schvalování vytvořených příspěvků a kontrola či případné smazání nahlášených příspěvků. Administrátorský účet může sloužit jako zastoupení dané obce či města a kromě správy komentářů či příspěvků lze jeho prostřednictvím vykonávat také všechny funkce klasického registrovaného uživatele.





6.3.1 SCHVALOVÁNÍ PŘÍSPĚVKŮ

Hlavní funkcí aplikace je vytváření příspěvků ze strany běžných uživatelů. Tyto příspěvky je ovšem nutno kontrolovat, zda jsou vhodné pro zobrazení na stránkách obce. K tomuto účelu slouží schvalovací sekce pro administrátory. Do této sekce se správce dostane z menu *Uživatel* obsaženým v hlavním menu hlavičky stránky.



Obrázek 66: Uživatelské menu administrátora

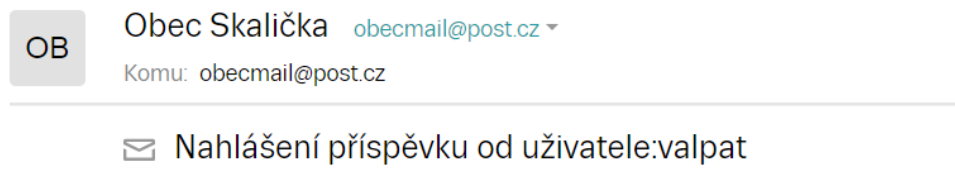
V sekci *Schválení příspěvků* jsou zobrazeny všechny příspěvky ze všech sekcí čekající na schválení. Příspěvky jsou seřazeny podle data vytvoření a obsahují všechny důležité parametry příspěvku, které administrátor potřebuje znát k rozhodnutí o jejich schválení. U každého příspěvku se pak nachází dvě tlačítka, kde jedno slouží ke schválení a jedno k zamítnutí zveřejnění příspěvku. V obou případech je uživatel informován pomocí e-mailu, který obsahuje rozhodnutí administrátora o schválení.

Typ	Uživatelské jméno	Název události	Datum a čas	Místo	Obrázek	Popis	Datum vytvoření	Schválit	Zamítnout
Událost	mirda123	KONCERT KAPELY AD/DC REVIVAL	18. 8. 2019 19:45:00	Sokolovna Skalička		AD/DC Revival konečně ve Skaličce! Vstupné: 150Kč, občerstvení zajištěno (pivo, limo, špekáčky...) Tak dojdě! :)	23. 3. 2019 17:34:45	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Typ	Uživatelské jméno	Název příspěvku		Obrázek	Text	Datum vytvoření	Schválit	Zamítnout	
Aktualita	svetkajarek	SLEVA NA KOLA			Autoservis Světka aktuálně nabízí slevu na kompletní sortiment kol pro Škoda Auto. Více informací přímo v servisu.	1. 4. 2019 7:32:02	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Typ	Uživatelské jméno	Název příspěvku		Obrázek	Text	Datum vytvoření	Schválit	Zamítnout	
Aktualita	mirda123	VAK PŘEROV UPOZORŇUJE			VaK Přerov žádá občany, aby si zkontrolovali své vodoměry a vodovodní přípojky z důvodu malého úniku vody v obci. Děkujeme	15. 4. 2019 8:02:14	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Typ	Uživatelské jméno	Název události	Datum a čas	Místo	Obrázek	Popis	Datum vytvoření	Schválit	Zamítnout
Událost	johanka	Barvám neutěšil Skalička!	11. 7. 2019 11:30:00	Fotbalové hřiště		Den, kdy se staneš součástí akce plné barev, skvělé nálad, přátelství, rodninné pohody a nezapomenutelných zážitků. Běh, ve kterém není hlavním motivem rychlost, ale především zábava. Děti, teenagell i dospělí, mladí i staří, jednotlivci, skupiny či rodiny s dětmi - zúčastnit se může opravdu každý! Vyraž na trasu dlouhou 5km během či chůzí a #barevným Tě v 5 barevných zónách zasype barvami.	17. 4. 2019 19:49:59	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Obrázek 67: Sekce sloužící ke schvalování příspěvků

6.3.2 NOTIFIKACE

Administrátorské účty obsahují stejné notifikace jako účet klasického uživatele. Administrátor je navíc informovaný o nahlášených příspěvcích nebo komentářích pomocí e-mailu. Tento e-mail obsahuje informace o tom, kdo příspěvek nahlásil, název příspěvku, autora a identifikační číslo daného příspěvku. V případě komentáře je odeslán celý text komentáře, identifikační číslo komentáře a identifikační číslo příspěvku, u kterého je obsažen.



Uživatel valpat nahlásil příspěvek 'KOUPÍM DŮM' od uživatele panacek. Identifikační číslo příspěvku = [52].

Je nutná manuální kontrola příspěvku!

Obrázek 68: Automaticky generovaný e-mail při nahlášení příspěvku

7 METODY ZABEZPEČENÍ

Framework ASP.NET dle názoru autora obsahuje velmi obstojnou úroveň zabezpečení. Čtvrtá kapitola této bakalářské práce představila shrnutí 10 nejčastějších rizik pro webové aplikace. Metody popsané v těchto rizicích autor následně nasimuloval a otestoval tak jím vyvíjenou aplikaci.

Části aplikace a funkce, které jsou přístupné pouze pro registrované uživatele, jsou ošetřeny pomocí atributu `[Authorize]`. Funkce pro výpis příspěvků používají atribut `[AllowAnonymous]`, jelikož k nim může přistupovat i neregistrovaný uživatel. Funkce, které má právo užívat pouze administrátor, používají atribut `[SecurityRight("Admin")]`, který pomocí třídy `SecurityRightAttribute` kontroluje roli přihlášeného uživatele. Některé prvky na straně HTML jsou rovněž zobrazovány nebo skryty podle autorizace nebo rolí uživatelů.

Aplikace byla testována na *Cross Site Scripting*. Do komentáře příspěvku se autor práce pokusil umístit text „Na tuto událost nezapomene nikdo! `</div><script>src=http://anonymous.cz/getAllData.js</script>`“. Po submitu (přidání) komentáře aplikace úspěšně rozpoznala hrozbu a komentář nebyl publikován.

Zanechat komentář

Na tuto událost nezapomene nikdo! `</div><script>src=http://anonymous.cz/getAllData.js</script>`

KOMENTOVAT

Obrázek 69: Cross Site Scripting

V aplikaci / došlo k chybě serveru.

Z klienta (`commentText="...ne nikdo! </div><script>src=ht..."`) byla zjištěna potenciálně nebezpečná hodnota `Request.Form`.

Popis: Technologie ASP.NET v počítačích zajišťuje data, která mohou být potenciálně nebezpečná, protože mohou obsahovat značku jazyka HTML nebo skript. Tato data mohou představit pokus o narušení zabezpečení aplikace, například útok pomocí skriptování mezi servery. Pokud je tento typ vstupních dat ve vaší aplikaci přijatelný, můžete do webové stránky zahrnout kód, který je explicitně povoluje. Další informace najdete na webu: <http://go.microsoft.com/fwlink/?LinkID=212874>

Obrázek 70: Chyba po submitu komentáře

Tato aplikace je taktéž ošetřena proti hrozbě *Insecure Direct Object References* – například tím, že editační tlačítko se u příspěvků zobrazuje pouze jeho autorovi. To ovšem

neznamená, že by se útočník hypoteticky nemohl pokusit dostat se na editaci cizího příspěvku pomocí URL adresy. V aplikaci je proto ověřováno ID přihlášeného uživatele s parametrem *CreatedBy*, aby k tomuto ohrožení bezpečnosti aplikace nedocházelo.

V rámci ověřování bezpečnosti byla aplikace taktéž testována na *SQL Injection*. Prvním testem bylo použití „*password:' or 1=1--*“ a dalších podobných triků při přihlašování do aplikace. Všechny zkoušené varianty neprošly a skončily chybovou hláškou oznamující zadání špatných přihlašovacích údajů. Další testy byly například zaměřeny na pokus smazání tabulky pomocí příkazu *DROP TABLE* vloženým do URL adresy, ale ani tyto pokusy neprošly a aplikace tak bezpečnostní hrozbě *SQL Injection* plně odolala.

8 MOŽNÝ ROZVOJ APLIKACE

Aplikace by mohla být vylepšena a rozšířena jak v rovině frontendu, tak backendu. Nejnutnější vylepšení by se ukázala po nasazení aplikace do ostrého provozu, například prostřednictvím pomoci zpětné vazby od jejích uživatelů.

V současné verzi aplikace při provedení některých funkcí dochází k reloadu stránky. Prvním vylepšením by tedy byla větší integrace JS, který dokáže snížit počet reloadů, díky čemuž by stránka na uživatele působila plynuleji.

Jedním z dalších rozšíření by bylo rozšíření uživatelských profilů, například o možnost nahrát vlastní profilový obrázek, vytvořit popis profilu nebo uvést kontaktní údaje. Příspěvky by bylo možné ukládat do sekce *Oblíbené* nebo v případě událostí by bylo možné nastavit upozornění pomocí e-mailu zasláného v den konání akce.

Podstatnou část aplikace tvoří notifikace, které by při budoucím rozvoji mohly být co nejvíce integrovány přímo do rozhraní aplikace bez nutnosti aplikaci opouštět. Notifikace by mohly vypadat a principiálně fungovat jako upozornění na sociální síti *Facebook*. Uživatel by se tak rychle a přehledně dozvěděl například o stavu schvalování jeho příspěvku v prostředí aplikace.

Aplikaci by dále bylo možné rozšířit pomocí různých komponent *Application Programming Interface* (API) – například přihlášením pomocí účtu *Google* nebo využitím *Facebook* API pro sdílení příspěvku na této sociální síti.

Obecně lze konstatovat, že možnosti rozšiřitelnosti aplikace jsou obrovské. Samotný rozvoj a finální stav aplikace by se odvíjel podle představ a požadavků zákazníka – v tomto případě obce či města.

ZÁVĚR

Cílem práce bylo vytvoření funkční aplikace pro obce a města, která bude sloužit pro tvorbu obsahu ze strany běžných občanů, kde kontrolním prvkem bude osoba pověřená obcí či městem.

Základní funkce aplikace, jež byly stanoveny při návrhu práce a aktualizovány po konzultacích s vedoucím práce v průběhu jejího vypracování, byly pokryty. Rovněž obec, která dala písemné povolení pro použití jejího jména, loga a fotek, byla s finálním zpracováním aplikace nadšena a budou s ní vedeny diskuze o případném rozvoji aplikace.

V teoretické části byly splněny první dva body zadání, které obsahovaly rešerši existujících řešení a rozbor použitých technologií. První byla zaměřena na CMS systémy umožňující správu webu, které jsou v dnešní době nejpoužívanější. Autor práce zde věnoval pozornost firmám a jejich řešením webových stránek několika měst a obcí v Olomouckém a Zlínském kraji. Druhá část byla věnována stručnému popisu technologií použitých ke zpracování návrhu a následné realizaci aplikace. Závěr teoretické části této bakalářské práce představuje rozbor nejčastějších rizik pro webovou aplikaci.

Počátek praktické části popisuje průběh návrhu aplikace od zpracování požadavků klienta, vytvoření diagramů a scénářů až po návrh databáze. Zbytek praktické části je koncipován jako uživatelská příručka rozdělená podle rolí uživatelů aplikace, přičemž jsou v této části popsány a znázorněny všechny hlavní funkce navržené aplikace. Závěr praktické části práce je věnován zabezpečení aplikace a testování bezpečnostních rizik zpracovaných v teoretické části.

Přestože byla stanovená funkcionalita aplikace splněna, lze identifikovat řadu oblastí, jež chci v budoucnu na této aplikaci vylepšit. I přesto jsem s její finální verzí spokojen. Tato práce pro mě byla přínosem v oblasti mých programovacích zkušeností, především díky objevení různých, mně dosud neznámých principů řešení a frameworků, které mohu využít ve svých nadcházejících projektech.

SEZNAM POUŽITÉ LITERATURY

- [1] 9 Outstanding Marketing CMS Platforms You Need to Know in 2019 [online]. [cit. 2019-03-11]. Dostupné z: <https://www.bluleadz.com/blog/the-8-best-marketing-cms-platforms-in-2018>
- [2] Jak na obecní web? [online]. [cit. 2019-03-11]. Dostupné z: <https://www.mvcr.cz/clanek/jak-na-obecni-web.aspx>
- [3] O WordPress [online]. [cit. 2019-03-11]. Dostupné z: <http://www.cwordpress.cz/>
- [4] WordPress [online]. [cit. 2019-03-11]. Dostupné z: <https://en.wikipedia.org/wiki/WordPress>
- [5] 10 nejlepších redakčních systémů (CMS) [online]. [cit. 2019-03-11]. Dostupné z: <https://www.interval.cz/clanky/10-nejlepsich-redakcnich-systemu-cms/>
- [6] WordPress vs Joomla vs Drupal (Comparison) [online]. [cit. 2019-03-11]. Dostupné z: <https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>
- [7] WORDPRESS – BEZPEČNOST [online]. [cit. 2019-03-11]. Dostupné z: <https://www.brilo.cz/wordpress-bezpecnost/>
- [8] Výhody a nevýhody WordPress [online]. [cit. 2019-03-11]. Dostupné z: <https://www.marketingmind.cz/vyhody-a-nevyhody-wordpress/>
- [9] Produkty a služby [online]. [cit. 2019-03-14]. Dostupné z: <https://webhouse.cz/produkty-a-sluzby/ms-2274/p1=2274>
- [10] Co vismo umí [online]. [cit. 2019-03-14]. Dostupné z: <https://vismo.cz/>
- [11] Velké Popovice Oficiální web [online]. [cit. 2019-03-14]. Dostupné z: <http://www.velkepopovice.cz/>
- [12] Oficiální stránky města Otrokovice [online]. [cit. 2019-03-14]. Dostupné z: <http://www.otrokovice.cz/>
- [13] Nejlepší WYSIWYG editory webových stránek 2019 [online]. [cit. 2019-03-14]. Dostupné z: <https://www.webhostingcentrum.cz/nejlepsi-wysiwyg-editory-webovych-stranek/>
- [14] Hlavní stránka Galileo Corporation [online]. [cit. 2019-03-14]. Dostupné z: <https://www.igalileo.cz/>
- [15] Naše řešení [online]. [cit. 2019-03-14]. Dostupné z: <https://www.igalileo.cz/nase-reseni/>
- [16] Redakční systém GCM [online]. [cit. 2019-03-14]. Dostupné z: <https://www.igalileo.cz/nase-reseni/nad-produkty/redakcni-system/>

- [17] Reference [online]. [cit. 2019-03-14]. Dostupné z: https://www.igalileo.cz/reference/obec-malhotice.html?k_1=1&kraj=CZ071
- [18] Joomla [online]. [cit. 2019-03-14]. Dostupné z: <https://www.techopedia.com/definition/3276/joomla>
- [19] Co je Joomla!? [online]. [cit. 2019-03-14]. Dostupné z: <https://www.joomlaportal.cz/o-joomla>
- [20] Joomla! [online]. [cit. 2019-03-14]. Dostupné z: <http://jakycmszvolit.cz/joomla.html>
- [21] Joomla [online]. [cit. 2019-03-14]. Dostupné z: <https://www.cms-systemy.cz/redakcni-systemy/joomla/13-clanky/260-joomla-4-na-obzoru>
- [22] Oficiální stránky Linux.com [online]. [cit. 2019-03-14]. Dostupné z: <https://www.linux.com/>
- [23] Drupal [online]. [cit. 2019-03-15]. Dostupné z: <https://cs.wikipedia.org/wiki/Drupal>
- [24] WordPress vs. Joomla vs. Drupal [online]. [cit. 2019-03-15]. Dostupné z: <https://www.webhostingsecretrevealed.net/cs/blog/web-tools/compare-top-3-cms-2017-wordpress-vs-joomla-vs-drupal/>
- [25] Drupal [online]. [cit. 2019-03-15]. Dostupné z: <http://jakycmszvolit.cz/drupal.html>
- [26] Ember [online]. [cit. 2019-03-15]. Dostupné z: <https://www.drupal.org/project/ember>
- [27] Oficiální stránky Tesla.com [online]. [cit. 2019-03-14]. Dostupné z: <https://www.tesla.com/>
- [28] Technologie pro vývoj webových aplikací [online]. [cit. 2019-03-17]. Dostupné z: <https://www.itnetwork.cz/html-css/ostatni/webove-aplikace/technologie-pro-vyvoj-webovych-aplikaci-popis-vybranych-technologii-2>
- [29] Unified Modeling Language [online]. [cit. 2019-04-08]. Dostupné z: https://cs.wikipedia.org/wiki/Unified_Modeling_Language
- [30] Úvod do UML [online]. [cit. 2019-04-08]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy>
- [31] Úvod do MVC architektury v ASP.NET [online]. [cit. 2019-04-08]. Dostupné z: <https://www.itnetwork.cz/csharp/asp-net/mvc/zaklady/asp-dot-net-uvod-do-mvc-architektury>

- [32] ASP.NET MVC Overview [online]. [cit. 2019-04-10]. Dostupné z: <https://docs.microsoft.com/en-us/aspnet/mvc/overview/older-versions-1/overview/asp-net-mvc-overview>
- [33] ASP.net MVC with entity framework and CSS. New York, NY: Springer Science+Business Media, 2016. ISBN 978-1484221365.
- [34] Hypertext Markup Language [online]. [cit. 2019-04-12]. Dostupné z: https://cs.wikipedia.org/wiki/Hypertext_Markup_Language
- [35] GOLDSTEIN, Alexis, Louis LAZARIS a Estelle WEYL. HTML5 a CSS3 pro webové designéry. Brno: Zoner Press, 2011. Encyklopedie webdesignera. ISBN 978-80-7413-166-0.
- [36] HTML5 [online]. [cit. 2019-04-12]. Dostupné z: <https://www.w3schools.com/html/>
- [37] Úvod do HTML [online]. [cit. 2019-04-12]. Dostupné z: <https://www.itnetwork.cz/html-css/webove-stranky/jak-psat-moderni-web-html-tutorial-uvod-do-html>
- [38] How to CSS Selectors Work [online]. [cit. 2019-04-12]. Dostupné z: <https://css-tricks.com/how-css-selectors-work/>
- [39] Basic CSS Crib Sheet [online]. [cit. 2019-04-12]. Dostupné z: https://www.wwp.northeastern.edu/outreach/seminars/intro_2018-04/handouts/basic_css_cribsheet.xhtml
- [40] LAMBERT, Matt. Learning Bootstrap 4. 2nd Revised edition. Birmingham: Packt Publishing, 2016. ISBN 978-1785881008.
- [41] SQL [online]. [cit. 2019-04-12]. Dostupné z: <https://cs.wikipedia.org/wiki/SQL>
- [42] Úvod do MS-SQL a příprava prostředí [online]. [cit. 2019-04-12]. Dostupné z: <https://www.itnetwork.cz/ms-sql/mssql-tutorial-uvod-a-priprava-prostredi>
- [43] Foreign Key [online]. [cit. 2019-04-12]. Dostupné z: https://www.teachict.com/gcse_new/databases/terminology/miniweb/pg8.htm
- [44] Entity Framework [online]. [cit. 2019-04-12]. Dostupné z: https://en.wikipedia.org/wiki/Entity_Framework
- [45] Jemný úvod do ORM a Entity Frameworku [online]. [cit. 2019-04-12]. Dostupné z: <https://www.youtube.com/watch?v=0nM38vBk6LI>
- [46] Code First vs. Database First vs. Model First Approach [online]. [cit. 2019-04-12]. Dostupné z: <https://www.c-sharpcorner.com/blogs/code-first-vs-database-first-vs-model-first-approach1>

- [47] Co je JavaScript? [online]. [cit. 2019-04-13]. Dostupné z: https://developer.mozilla.org/cs/docs/Learn/JavaScript/First_steps/Co_je_JavaScript
- [48] DUCKETT, Jon, Gilles RUPPERT a Jack MOORE. JavaScript & jQuery: interactive front-end web development. Indianapolis, IN: Wiley, 2014. ISBN 9781118531648.
- [49] Úvod do JavaScriptu [online]. [cit. 2019-04-13]. Dostupné z: <https://www.it-network.cz/javascript/zaklady/javascript-tutorial-uvod-do-javascriptu-nepochopeny-jazyk>
- [50] 10 nejčastějších rizik webových aplikací [online]. [cit. 2019-04-14]. Dostupné z: <https://www.zabezpecujeme.net/10-nejcastejsich-rizik-webovych-aplikaci>
- [51] SQL injection [online]. [cit. 2019-04-14]. Dostupné z: <https://portswigger.net/web-security/sql-injection>
- [52] Broken Authentication and Session Management [online]. [cit. 2019-04-14]. Dostupné z: <https://gbhackers.com/broken-authentication-and-session-management/>
- [53] Cross-Site Scripting [online]. [cit. 2019-04-14]. Dostupné z: <https://portswigger.net/web-security/cross-site-scripting>
- [54] How-To: Find IDOR [online]. [cit. 2019-04-14]. Dostupné z: <https://www.bugcrowd.com/how-to-find-idor-insecure-direct-object-reference-vulnerabilities-for-large-bounty-rewards/>
- [55] Web Application Security Risks [online]. [cit. 2019-04-14]. Dostupné z: <https://medium.com/@infosecsanyam/owasp-top-10-2017-web-application-security-risks-31f356491712>
- [56] Sensitive Data Exposure – A Nightmare To All Business Enterprises [online]. [cit. 2019-04-14]. Dostupné z: <https://www.indusface.com/blog/sensitive-data-exposure-nightmare-business-enterprises/>
- [57] Cross-Site Request Forgery (CSRF) [online]. [cit. 2019-04-14]. Dostupné z: <https://medium.com/cross-site-request-forgery-csrf/cross-site-request-forgery-csrf-bc92886dc5b>
- [58] Using Components with known Vulnerabilities – Every Developer Should aware [online]. [cit. 2019-04-14]. Dostupné z: <https://gbhackers.com/9-using-components-known-vulnerabilities/>

- [59] What is an Open Redirection Vulnerability and How to Prevent it? [online]. [cit. 2019-04-14]. Dostupné z: <https://www.netsparker.com/blog/web-security/open-redirection-vulnerability-information-prevention/>
- [60] PHP Vs ASP.NET: How to Choose the Right One? [online]. [cit. 2019-04-27]. Dostupné z: <https://www.pixelcrayons.com/blog/web/php-vs-asp-net-how-to-choose-the-right-one/>
- [61] 1. lekce: Co je to PHP a práce se zdrojovým kódem [online]. [cit. 2019-04-27]. Dostupné z: <https://www.youtube.com/watch?v=WlpuMurAHXU>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

UML	Unified Modeling Language
MVC	Model-View-Controller
HTML	Hypertext Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
CMS	Content Management System
LDAP	Lightweight Directory Access Protocol
WYSIWYG	What You See Is What You Get
EF	Entity Framework
JS	JavaScript
URL	Uniform Resource Locator
API	Application Programming Interface

SEZNAM OBRÁZKŮ

<i>Obrázek 1: CMS WordPress</i>	12
<i>Obrázek 2: Příklady velkých webů využívajících WordPress</i>	13
<i>Obrázek 3: CMS Joomla 4.0</i>	14
<i>Obrázek 4: Stránka Linux.com využívající Joomla</i>	15
<i>Obrázek 5: CMS Drupal</i>	16
<i>Obrázek 6: Stránka Tesla.com využívající Drupal</i>	16
<i>Obrázek 7: Webové stránky obce Velké Popovice od firmy Webhouse</i>	17
<i>Obrázek 8: Webové stránky města Otrokovice od firmy Webhouse</i>	18
<i>Obrázek 9: Webové stránky obce Malhotice od firmy Galileo Corporation</i>	20
<i>Obrázek 10: Princip fungování PHP</i>	22
<i>Obrázek 11: UML složený ze 14 diagramů</i>	23
<i>Obrázek 12: Architektura MVC</i>	24
<i>Obrázek 13: Struktura HTML</i>	26
<i>Obrázek 14: Stylizace elementu h1</i>	27
<i>Obrázek 15: Struktura CSS</i>	27
<i>Obrázek 16: Struktura tabulky Users</i>	28
<i>Obrázek 17: Primární a cizí klíč</i>	29
<i>Obrázek 18: Příklad využití Stored procedure</i>	30
<i>Obrázek 19: Tři možné přístupy EF</i>	31
<i>Obrázek 20: JS pro skrytí a zobrazení části hodnocení akcí a JS pro datepicker</i>	32
<i>Obrázek 21: Příklad SQL injection</i>	33
<i>Obrázek 22: Příklad Broken Authentication and Session Management</i>	34
<i>Obrázek 23: Příklad Cross-Site Scripting</i>	34
<i>Obrázek 24: Příklad Insecure Direct Object References</i>	35

<i>Obrázek 25: Příklad Security Misconfiguration</i>	36
<i>Obrázek 26: Příklad Sensitive Data Exposure</i>	37
<i>Obrázek 27: Příklad Cross-Site Request Forgery</i>	38
<i>Obrázek 28: Příklad Using Components with Known Vulnerabilities</i>	38
<i>Obrázek 29: Příklad Unvalidated Redirects and Forwards</i>	39
<i>Obrázek 30: Funkční požadavky</i>	42
<i>Obrázek 31: Správa požadavků</i>	43
<i>Obrázek 32: Správa komentářů</i>	43
<i>Obrázek 33: Správa uživatelů</i>	44
<i>Obrázek 34: Nefunkční požadavky</i>	44
<i>Obrázek 35: Bezpečnost</i>	44
<i>Obrázek 36: Use Case model aplikace</i>	45
<i>Obrázek 37: Fyzický model databáze aplikace</i>	58
<i>Obrázek 38: Home page aplikace</i>	59
<i>Obrázek 39: Dropdown menu</i>	59
<i>Obrázek 40: Registrace uživatele do aplikace</i>	60
<i>Obrázek 41: Validace zadaných údajů při potvrzení registrace</i>	61
<i>Obrázek 42: Notifikace úspěšného přihlášení do aplikace</i>	61
<i>Obrázek 43: Přihlašovací formulář do aplikace</i>	62
<i>Obrázek 44: Modal element pro zadání e-mailu</i>	62
<i>Obrázek 45: E-mail s odkazem pro změnu hesla</i>	63
<i>Obrázek 46: Vygenerovaný ResetPasswordCode</i>	63
<i>Obrázek 47: Uživatelské menu</i>	63
<i>Obrázek 48: Vytvoření aktuality</i>	64
<i>Obrázek 49: Validace velikosti nahraného obrázku</i>	64

<i>Obrázek 50: Formulář pro vytvoření aktuality s ukázkou validace délky textu</i>	65
<i>Obrázek 51: Formulář pro vytvoření události</i>	66
<i>Obrázek 52: Datepicker pro zadání data a času události</i>	66
<i>Obrázek 53: Formulář pro vytvoření nabídky/poptávky</i>	67
<i>Obrázek 54: Zvolení kategorie při vytváření nabídky/poptávky</i>	67
<i>Obrázek 55: Příspěvek s tlačítkem pro zobrazení komentářů</i>	68
<i>Obrázek 56: Komentářová sekce příspěvku</i>	68
<i>Obrázek 57: Tlačítko pro úpravu a smazání příspěvku</i>	69
<i>Obrázek 58: Formulář pro úpravu příspěvku v diskuzi</i>	69
<i>Obrázek 59: Tlačítko pro nahlášení příspěvku</i>	70
<i>Obrázek 60: Datepicker pro zvolení datumového rozsahu</i>	70
<i>Obrázek 61: Zvolení kategorie ve filtru</i>	71
<i>Obrázek 62: Vytvoření hodnocení akce</i>	71
<i>Obrázek 63: Přehled hodnocení od jednotlivých uživatelů</i>	72
<i>Obrázek 64: Automaticky generovaný e-mail po úspěšné registraci</i>	72
<i>Obrázek 65: Druhy notifikací v aplikaci</i>	73
<i>Obrázek 66: Uživatelské menu administrátora</i>	73
<i>Obrázek 67: Sekce sloužící k schvalování příspěvků</i>	74
<i>Obrázek 68: Automaticky generovaný e-mail při nahlášení příspěvku</i>	74
<i>Obrázek 69: Cross Site Scripting</i>	75
<i>Obrázek 70: Chyba po submitu komentáře</i>	75

SEZNAM TABULEK

Tabulka 1: Scénář – Registrace uživatele	46
Tabulka 2: Alternativní scénář registrace uživatele.....	46
Tabulka 3: Alternativní scénář registrace uživatele.....	46
Tabulka 4: Scénář – Přihlášení uživatele	47
Tabulka 5: Alternativní scénář přihlášení uživatele	47
Tabulka 6: Scénář – Vytvoření komentáře	48
Tabulka 7: Alternativní scénář vytvoření komentáře	48
Tabulka 8: Alternativní scénář vytvoření komentáře	48
Tabulka 9: Alternativní scénář vytvoření komentáře	49
Tabulka 10: Scénář – Vytvoření hodnocení akce	49
Tabulka 11: Alternativní scénář vytvoření hodnocení akce	49
Tabulka 12: Alternativní scénář vytvoření hodnocení akce	50
Tabulka 13: Scénář – Nahlášení příspěvku/komentáře.....	50
Tabulka 14: Alternativní scénář nahlášení příspěvku/komentáře.....	50
Tabulka 15: Scénář – Změna zapomenutého hesla.....	51
Tabulka 16: Alternativní scénář změny zapomenutého hesla	51
Tabulka 17: Alternativní scénář změny zapomenutého hesla	51
Tabulka 18: Scénář – Vytvoření příspěvku	52
Tabulka 19: Alternativní scénář vytvoření příspěvku.....	52
Tabulka 20: Alternativní scénář vytvoření příspěvku.....	53
Tabulka 21: Alternativní scénář vytvoření příspěvku.....	53
Tabulka 22: Scénář – Úprava příspěvku.....	53
Tabulka 23: Alternativní scénář úpravy příspěvku.....	54
Tabulka 24: Alternativní scénář úpravy příspěvku.....	54
Tabulka 25: Alternativní scénář úpravy příspěvku.....	54
Tabulka 26: Scénář – Smazání příspěvku.....	55
Tabulka 27: Alternativní scénář smazání příspěvku/komentáře.....	55
Tabulka 28: Scénář – Schválení příspěvku.....	56
Tabulka 29: Alternativní scénář schválení příspěvku.....	56
Tabulka 30: Alternativní scénář schválení příspěvku.....	56

SEZNAM PŘÍLOH

P I Obsah CD

P II Souhlas s použitím údajů obce Skalička

PŘÍLOHA P I: OBSAH CD

Přiložené CD obsahuje:

- Text
 - Navrh_aplikace_pro_webove_stranky_mest_ValuchPatrik.docx – bakalářská práce ve formátu docx
 - Navrh_aplikace_pro_webove_stranky_mest_ValuchPatrik.pdf - bakalářská práce ve formátu pdf
 - BakalarskaPrace_UzivatelскеUcty_ValuchPatrik.xlsx – Seznam již vytvořených uživatelů v aplikaci s přihlašovacími údaji
- Zdrojové kódy
 - PraktickaCast_ValuchPatrik.zip – všechny zdrojové kódy, export databáze

PŘÍLOHA P II: SOUHLAS S POUŽITÍM ÚDAJŮ OBCE SKALIČKA

SOUHLAS S POUŽITÍM ÚDAJŮ OBCE SKALIČKA

Udělujeme tímto souhlas Patriku Valuchovi narozeného 16.9.1995 k použití údajů obce Skalička ve své bakalářské práci *Návrh aplikace pro webové stránky měst* na Univerzitě Tomáše Bati ve Zlíně, Fakultě aplikované informatiky.

Využity budou tyto údaje:

- Název obce
- Fotografie obce
- Znak obce
- Kontakty na obec
- Příspěvky z webových stránek obce

6. 5. 2019

datum

.....

starostka obce

OBEC SKALIČKA
okres Přerov