

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Tvorba webových aplikací v jazycích Python a PHP

Bc. Jaroslav Tesař

© 2020 ČZU v Praze

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Jaroslav Tesař

Systémové inženýrství a informatika
Informatika

Název práce

Tvorba webových aplikací v jazycích Python a PHP

Název anglicky

Web-application development in languages Python and PHP

Cíle práce

Cílem práce je zhotovit webovou aplikaci v jazycích PHP nebo Python a databázi MySQL, která se bude zabývat redakčním rozhraním mezinárodního odborného časopisu. Aplikace bude umožňovat odesílání informací o článcích k jejich následnému double-blind review. (autoři, titul, titul anglicky, abstrakt, abstrakt anglicky, klíčová slova, klíčová slova anglicky, URL celého textu) Tyto informace budou redakční radě odesílat pouze registrovaní uživatelé a administrátor může jejich registraci potvrdit nebo dočasně odebrat nebo zcela zrušit. Přihlášený uživatel může dostat od admina právo stahovat PDF verzi hotových vydání časopisu. Dále bude v práci obsaženo porovnání výhod a nevýhod tvorby této aplikace ve srovnání s konfigurací hotových komerčních řešení typu Wordpress a OJS.

Metodika

Práce bude obsahovat dvě části, a to teoretickou a praktickou. Teoretická část se bude zabývat webovými aplikacemi a jejich tvorbou obecně, a dále použitý programovací jazyk a jeho vlastnosti. Teoretická část práce bude obsahovat analýzu a syntézu vědeckých a odborných zdrojů, které jsou zaměřeny na webové aplikace a jazyk PHP nebo Python. Základem praktické části bude webová aplikace, psaná v PHP nebo v Pythonu včetně projektové dokumentace. Její realizace bude výsledkem výběru ve výše zmíněných jazycích a bude obsahovat jejich porovnání.

Doporučený rozsah práce

80 – 100 stran

Klíčová slova

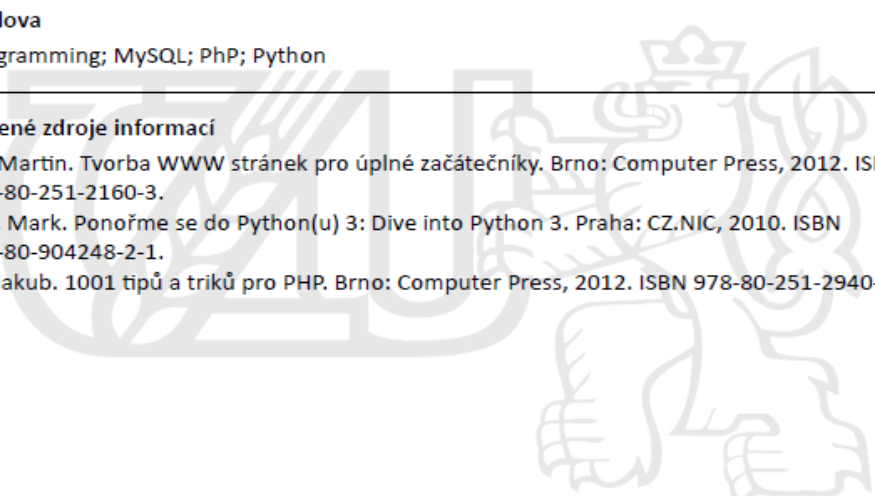
Web programming; MySQL; PHP; Python

Doporučené zdroje informací

DOMES, Martin. Tvorba WWW stránek pro úplné začátečníky. Brno: Computer Press, 2012. ISBN 978-80-251-2160-3.

PILGRIM, Mark. Ponořme se do Python(u) 3: Dive into Python 3. Praha: CZ.NIC, 2010. ISBN 978-80-904248-2-1.

VRÁNA, Jakub. 1001 tipů a triků pro PHP. Brno: Computer Press, 2012. ISBN 978-80-251-2940-1.



Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

doc. Ing. Vojtěch Merunka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 25. 02. 2020

Čestné prohlášení

Prohlašuji, že jsem svou diplomovou práci „Tvorba webových aplikací v jazycích Python a PHP“ vypracoval samostatně, pod vedením vedoucího diplomové práce s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a jsou uvedeny v seznamu použitých zdrojů na konci práce. Jako autor výše uvedené práce také prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 3. 4. 2020

Poděkování

Rád bych poděkoval touto cestou vedoucímu práce Doc. Ing. Vojtěchu Merunkovi, Ph.D. za odborné konzultace a připomínky k mé diplomové práci. Velké poděkování patří také mojí rodině za neustálou podporu v průběhu celého studia.

Tvorba webových aplikací v jazycích Python a PHP

Abstrakt

Cílem práce je zhotovit webovou aplikaci v jazycích PHP nebo Python a databázi MySQL, která se bude zabývat redakčním rozhraním mezinárodního odborného časopisu. Aplikace bude umožňovat odesílání informací o článcích k jejich následnému double-blind review. (autoři, titul, titul anglicky, abstrakt, abstrakt anglicky, klíčová slova, klíčová slova anglicky, URL celého textu) Tyto informace budou redakční radě odesílat pouze registrovaní uživatelé a administrátor může jejich registraci potvrdit nebo dočasně odebrat nebo zcela zrušit. Přihlášený uživatel může dostat od admina právo stahovat PDF verzi hotových vydání časopisu. Dále bude v práci obsaženo porovnání výhod a nevýhod tvorby této aplikace ve srovnání s konfigurací hotových komerčních řešení typu Wordpress a OJS.

Práce bude obsahovat dvě části, a to teoretickou a praktickou. Teoretická část se bude zabývat webovými aplikacemi a jejich tvorbou obecně, a dále použitý programovací jazyk a jeho vlastnosti. Teoretická část práce bude obsahovat analýzu a syntézu vědeckých a odborných zdrojů, které jsou zaměřeny na webové aplikace a jazyk PHP nebo Python. Základem praktické části bude webová aplikace, psaná v PHP nebo v Pythonu včetně projektové dokumentace. Její realizace bude výsledkem výběru ve výše zmíněných jazycích a bude obsahovat jejich porovnání.

Klíčová slova: Tvorba webových stránek; MySQL; PHP; Python

Web-application development in languages Python and PHP

Abstract

The goal of this thesis is to create web application in languages PHP or Python and database MySQL, which will be mainly focused on editorial interface of international technical magazine. Application will be enable sending information about articles for follow-up double-blind review. (authors, title, title in English, abstract, abstract in English, key words, key words in English, URL adress of the whole text) These information will send just by registred users to editorial council and administrator can accept registration or temporarily remove it or delete it. Users which are sign in can obtain right from administrator to download PDF version finished publication of a magazine. Thesis will be contain a comparison advantages and disadvantages of making this application by used languages and configuration of completed commercial solutions like Wordpress and OJS.

Thesis will be contain two parts, teoretical and practical. Theoretical part will be focused on web applications and its creation in general, used programming language and its properties. Theoretical part will be realized by analysis and synthesis scientific and technical sources, which are focused on web applications and languages PHP and Python. The base of practical part will be web application written in PHP of Python including project documentation. Realization of practical part will be a result of choice in programming languages which were mentioned before and will be contain its comparison.

Key words: Web programming; MySQL; PHP; Python

Obsah

1. Úvod.....	11
2. Cíl práce a metodika	13
2.1 Cíl práce.....	13
2.2 Metodika.....	13
3. Teoretická východiska	14
3.1 Webová stránka	14
3.2 Druhy webových stránek	14
3.2.1 Statické webové stránky.....	14
3.2.2 Dynamické webové stránky	15
3.3 Webové aplikace.....	15
3.3.1 Využití webových aplikací	16
3.3.2 Priority při tvorbě webových aplikací.....	17
3.4 Jazyk HTML (Hypertext Markup Language).....	18
3.4.1 Historie HTML	18
3.4.2 Struktura HTML	20
3.5 CSS (Cascading Style Sheets)	21
3.6 PHP	23
3.6.1 Historie PHP	24
3.6.2 Základní syntaxe PHP	26
3.6.3 Kombinace PHP a HTML kódu.....	27
3.6.4 Výhody PHP	28
3.6.5 Nevýhody PHP.....	28
3.7 Python.....	29
3.7.1 Historie Pythonu	29
3.7.2 Základní syntaxe Pythonu.....	30
3.7.3 Výhody Pythonu	32
3.7.4 Nevýhody Pythonu.....	32
3.8 Databáze	33
3.8.1 MySQL	33
3.8.2 PostgreSQL.....	34
3.9 WordPress.....	34

3.9.1 Výhody WordPressu	35
3.9.2 Nevýhody WordPressu	36
3.10 OJS.....	37
3.10.1 Výhody OJS	37
3.10.2 Nevýhody OJS	38
3.11 Metoda funkčních bodů.....	39
3.12 COCOMO model.....	39
4. Vlastní práce	40
4.1 Analýza požadavků.....	40
4.1.1 Požadavky na funkce aplikace	40
4.1.2 Uživatelské role	41
4.2 Návrh funkcionalit aplikace.....	42
4.2.1 Databázový model	42
4.2.2 Diagram případů užití	43
4.2.3 Seznam případů užití.....	44
4.2.4 Drátové modely (wireframy)	45
4.2.4.1 Sekce registrace uživatele	46
4.2.4.2 Sekce přihlášení uživatele	46
4.2.4.3 Sekce pro administrátora	47
4.2.4.5 Sekce vkládání návrhů článků.....	48
4.2.4.6 Odkazy pro stahování hotových verzí časopisu	49
4.2.5 Použitý hardware, software a technologie pro vývoj.....	49
4.3 Implementace.....	50
4.3.1 Nahrání stránek na internet	50
4.3.2 Sekce registrace uživatele	51
4.3.3 Sekce přihlášení uživatele.....	54
4.3.4 Sekce administrace.....	56
4.3.5 Sekce vkládání návrhů článků	60
4.3.6 Odkazy pro stahování hotových verzí časopisu.....	61
4.4 Výsledný výběr mezi PHP a Pythonem.....	63
4.5 Komparace PHP s WordPressem	64
4.6 Komparace PHP s OJS	65
4.7 Odhad potřebného času na projekt pomocí modelu COCOMO.....	65

4.8 Odhad použitých řádků kódu na projekt pomocí metody funkčních bodů.....	66
5. Výsledky a diskuze	68
6. Závěr.....	75
7. Seznam použitých zdrojů.....	77
8. Přílohy	81
8.1 Seznam obrázků.....	81
8.2 Seznam tabulek.....	82

1. Úvod

Tato diplomová práce se zabývá problematikou webové aplikace zahraničního odborného časopisu umožňující registraci a přihlášení uživatelů, vkládáním informací o článcích do databáze pro přihlášené uživatele se schválenou registrací, přidělením práva ke stahování hotových verzí časopisu a schválení, odebrání případně úplně smazání registrace uživatele administrátorem. Jedná se o konkrétně realizaci takzvaného „back endu“ aplikace, neboli realizování přístupu k datům a operaci s nimi.

Následující práce byla vytvořena na základě objednávky mezinárodní knihovny CEEOL z Německa sídlící ve Frankfurtu nad Mohanem. Tato knihovna je hlavním poskytovatelem akademických elektronických časopisů a knih týkajících se civilizace a sociálních věd ve střední a východní Evropě. V rychle se měnící digitální sféře je CEEOL spolehlivý zdroj odborných materiálů, které berou v potaz vědci, nakladatelé a knihovníci. Aktuálně se zhruba 1000 nakladatelů svěřuje do péče této knihovny s kvalitními elektronickými časopisy a knihami. Knihovna pokrývá více než 2000 časopisů, 480 000 článků, 2200 elektronických knih a 2500 dokumentů takzvané šedé literatury. CEEOL nabízí vědcům, výzkumníkům a studentům široké spektrum přístupu k akademickým materiálům, které se konzistentně vyvíjejí. Knihovna nabízí různorodé služby pro podporu institucí a jejich patronů, aby mohla vytvořit co nejvíce kvalitního obsahu a poskytnout k němu co nejjednodušší přístup. Navíc je umožněno nakladatelům získat nové příznivce a podporovat vědecké úspěchy a vědeckou komunitu ve střední a východní Evropě. Na následujícím obrázku se můžete přesvědčit, že časopis Slovjani.info je součástí knihovny CEEOL. Další informace o časopisu, které jsou vedené v knihovně, se v případě zájmu můžete dozvědět na internetové adrese <https://www.ceeol.com/search/journal-detail?id=1890>. (CEEOL2)

The screenshot shows the website for SLOVJANI.info, which is part of the CEEOL library. The header includes the CEEOL logo and navigation links like HOME, SUBJECT AREAS, PUBLISHERS, etc. The main content area features a search bar and a navigation menu. Below the search bar, there are tabs for 'Issues/Articles' and 'Journal Information'. The 'Issues/Articles' tab is active, showing the current issue (2019, Volume 4, Number 2) and a list of articles. The 'Journal Information' tab is also visible. On the right side, there is a section for 'Most popular articles from this journal' with a list of articles and their details.

Obrázek 1 – Časopis Slovjani.info jako součást knihovny CEEOL

Zdroj: (CEEOL)

První část práce popisuje potřebná teoretická východiska nutná k uvedení problematiky a pochopení jednotlivých použitých technologií. Podrobněji popsány jsou zde webové stránky a jejich druhy, webové aplikace, jejich použití a priority při tvorbě, jazyky HTML, CSS, PHP, Python, databázové systémy, MySQL, PostgreSQL a hotová komerční řešení pro tvorbu webových stránek Wordpress a OJS jakožto hotové řešení redakčního systému pro publikování elektronických časopisů. Nakonec je zmíněn model COCOMO a metoda funkčních bodů, které byly použity pro odhad stráveného času na projektu a přibližný počet řádku kódu.

Druhá část práce obsahuje analýzu návrhu jednotlivých požadovaných funkcí, které bylo třeba v aplikaci realizovat. Dále je v této části popsán postup tvorby jednotlivých komponent webové aplikace a komentář k nim. Tyto ukázky obsahují také obrázky některých částí zdrojového kódu, který byl pro programování aplikace použit. Poslední dílčí částí je zde srovnání použitého programovacího jazyka s hotovými komerčními řešeními jako Wordpress a OJS a vyhodnocení odhadů pomocí COCOMO modelu a metody funkčních bodů.

2. Cíl práce a metodika

2.1 Cíl práce

Cílem práce je zhotovit webovou aplikaci v jazycích PHP nebo Python a databázi MySQL, která se bude zabývat redakčním rozhraním mezinárodního odborného časopisu. Aplikace bude umožňovat odesílání informací o článcích k jejich následnému double-blind review. (autoři, titul, titul anglicky, abstrakt, abstrakt anglicky, klíčová slova, klíčová slova anglicky, URL celého textu) Tyto informace budou redakční radě odesílat pouze registrovaní uživatelé a administrátor může jejich registraci potvrdit nebo dočasně odebrat nebo zcela zrušit. Přihlášený uživatel může dostat od admina právo stahovat PDF verzi hotových vydání časopisu. Dále bude v práci obsaženo porovnání výhod a nevýhod tvorby této aplikace ve srovnání s konfigurací hotových komerčních řešení typu Wordpress a OJS.

2.2 Metodika

Práce bude obsahovat dvě části, a to teoretickou a praktickou. Teoretická část se bude zabývat webovými aplikacemi a jejich tvorbou obecně, a dále použitým programovacím jazykem a jeho vlastnostmi. Teoretická část práce bude obsahovat analýzu a syntézu vědeckých a odborných zdrojů, které jsou zaměřeny na webové aplikace a jazyk PHP nebo Python. Základem praktické části bude webová aplikace, psaná v PHP nebo v Pythonu včetně projektové dokumentace. Její realizace bude výsledkem výběru ve výše zmíněných jazycích a bude obsahovat jejich porovnání.

3. Teoretická východiska

3.1 Webová stránka

Webová nebo internetová stránka je určitý dokument s informacemi či daty, který si uživatel může zobrazit prostřednictvím webového prohlížeče na různých zařízeních. Mezi nejpoužívanější přístroje patří osobní počítače, notebooky, mobilní telefony a tablety. Stránky jsou přístupné více uživatelům najednou, jelikož jsou dostupné pomocí internetu. Informace na stránkách obsažené by měly být prezentovány srozumitelným a estetickým způsobem, aby byly čitelné pro návštěvníky webu. Na internetu může být zveřejněno široké spektrum obsahu. Často se jedná pouze o prostý text vzájemně provázaný, který poskytuje určitou informaci o věcech, které uživatele zajímají. Dále může být obsah také multimediální ve formě obrázků, videí nebo zvuků. (DOMES, 2012)

Lidé se mohou na webovou stránku dostat dvěma způsoby. První spočívá v zadání unikátní URL adresy (Uniform Resource Locator) do webového prohlížeče (např. Google Chrome, Mozilla Firefox, Opera atd.). Tato adresa definuje, kde přesně na internetu je webová stránka umístěna a po jejím zadání se návštěvník dostane na požadované místo. Druhou možností je vyhledání stránky pomocí internetových vyhledávačů (např. Google, Bing, Seznam). Zde uživatel také získá adresu hledané stránky, na kterou se dostane jedním kliknutím myši. (Managementmania, 2018)

3.2 Druhy webových stránek

Základní rozdělení bere v potaz, jak se internetová stránka chová navenek. Důležité zde je, jestli stránka mění svůj obsah v čase nebo vypadá stále stejně. Z tohoto pohledu webové stránky budeme definovat statické a dynamické. V současné době společnosti hodně řeší, jestli zvolí statické nebo dynamické stránky, jelikož v nich jsou podstatné rozdíly. (Altaxo)

3.2.1 Statické webové stránky

Tento typ internetových stránek je většinou vytvořený vývojářem v jazycích HTML za použitím CSS stylů včetně textů, obrázků, tabulek a dalších podobných prvků. Nevýhodou statických stránek je, že k jejich vytvoření je nutné se orientovat ve výše zmíněných technologiích. Pokud tyto jazyky člověk nezná, tak je pro něj tvorba velmi obtížná. Bez jejich

znalosti není možná ani aktualizace. Použití tohoto druhu stránek se uplatňuje nejčastěji, když na webu není nutné často měnit obsah nebo zobrazení. Změny obsahu na statických stránkách se dělají přímo v HTML a CSS kódu a je tedy nutné mít určitou znalost těchto jazyků k jejich provedení. Výhodou je, že můžeme přímo kontrolovat kvalitu HTML kódu na rozdíl od redakčních systémů, které kód přímo generují. Tento kód nemusí být například v souladu s optimalizací pro vyhledávače (SEO). Statické webové stránky jsou také velmi úsporné na finance, jelikož je lze provozovat na hostingu zdarma. Není třeba využívat technologie Apache ani databázi s jazykem SQL. Nejčastějším příkladem statických stránek jsou například webové prezentace, které neobsahují příliš velké množství stránek nebo jednostránkové webové vizitky. Každá stránka má svůj vlastní soubor s příponou .html nebo .php. Spravovat statické stránky s velkým množstvím stránek je v dnešní době zbytečné a náročné. (Helpmark, 2012)

3.2.2 Dynamické webové stránky

Dynamické weby na rozdíl od statických mění svůj vzhled a obsah v čase nebo na základě akcí návštěvníka. Tyto stránky jsou ideální pro častou změnu obsahu, jelikož úprava je trochu jednodušší než u statických webů. Navíc jsou také často tvořené pomocí redakčních systémů jako například WordPress, Joomla! nebo Drupal. Při tvorbě pomocí těchto systémů není nutná znalost jazyků HTML, CSS a tak dále jako při použití statických stránek. Je určitě výhodou, protože i v redakčních systémech lze kód většinou upravovat podle svého, ale není podmínkou. Za účelem dynamiky jsou používány známé skriptovací jazyky jako například PHP, JavaScript, Python nebo technologie AJAX. Používají se jak technologie na straně serveru i technologie na straně klienta. Použití těchto jazyků a implementace některých funkcí také dost často vyžaduje mít databázi s podporou SQL a využívat technologie na straně serveru jako Apache. Toto řešení je tak oproti statickým webovým stránkám dražší, jelikož si už tvůrce nevystačí s webhostingem zdarma. Příkladem funkcí dynamických stránek je třeba reakce na pohyb kurzoru na stránce, předkládání obsahu podle konkrétního návštěvníka stránky, provádění různých výpočetních operací, odesílání dat prostřednictvím kontaktních formulářů nebo práce se systémem odesílání souborů. (Helpmark2, 2012)

3.3 Webové aplikace

Webovou aplikací je často myšlen skript, který běží na straně serveru. Skriptem se rozumí zdrojový kód, který zajišťuje jednotlivé funkčnosti programu, aby běžely správně. Ten bývá obvykle propojen s nějakou databází, která uchovává data webové aplikace používaná dále

v programu. Výsledkem skriptu je poté samostatná webová stránka, která je předána internetovému prohlížeči k zobrazení. Hlavní úlohou webových aplikací je zvýšení interakce mezi webovou stránkou a jejím uživatelem. Rozsah se může hodně lišit kvůli různým požadavkům na jejich funkčnost. Jednodušší aplikace obsahuje například jen malé množství řádků kódu. Složitější se naopak mohou dostat klidně i do tisícovek řádků. Rozsáhlejší webové aplikace jsou často navázané na různý další software uvnitř společnosti. Častým příkladem interních firemních programů jsou systémy podporující řízení objednávek nebo různé účetní a fakturační programy. (Adaptic, 2018)

Jako příklad jednoduchých webových aplikací můžeme uvést kontaktní formuláře, diskuze, chat nebo různé slovníky, které jsou dostupné na internetu. Patří sem také katalogy, ceníky nebo bannerové systémy. Ty jsou nejčastěji využívány na webech různých obchodů či obchodníků. Často je můžeme vidět také na různých e-shopech. Slouží především k zobrazování zajímavých slev, nabídek nebo produktů, které zpravidla mají za úkol upoutat stálého či potenciálně nového zákazníka. (Adaptic, 2018)

Obrázek 2 – Formulář pro registraci uživatelů na vlastní stránce

Zdroj: Autor

3.3.1 Využití webových aplikací

Webové aplikace jsou uzpůsobeny k řešení různých problémů nebo úkolů. Mají bohaté využití, které umožňuje pohodlné používání pro návštěvníky nebo jednoduchou správu pro jejich vývojáře. Prvním častým použitím aplikací dostupných na internetu je snadné a rychlé hledání požadovaných informací na stránce s širokým obsahovým spektrem. Tyto typy aplikací

umožňují svým návštěvníkům libovolné prohledávání, organizování a procházení obsahu stránky. Organizování obsahu bývá obvykle realizováno pomocí různých filtrů, které pomáhají zákazníkům určit hledaný produkt. Příkladem takových aplikací může být například dobře známý web Amazon nebo jakýkoliv další stránky nabízející určité produkty nebo služby. (Adobe, 2015)

Dalším časté využití webových aplikací spočívá ve shromažďování, skladování a analýze dat, která byla získána prostřednictvím návštěvníků určitých stránek. V minulosti se údaje z internetových formulářů odesílaly ke zpracování odpovědným zaměstnancům nebo do aplikací určených k jejich zpracování. Dnes může aplikace ukládat data z kontaktních formulářů direktivně do databáze, data určitým způsobem filtrovat nebo vytvořit webovou zprávu pro analýzu dat. Mezi aplikace tohoto typu nejčastěji patří weby týkající se internetového bankovníctví, stránky různých obchodů nebo také ankety a formuláře určené pro zpětnou vazbu od uživatelů jednotlivých internetových stránek. Určité webové aplikace také slouží k tomu, aby ulevily jejich vývojářům od nutnosti ručně aktualizovat často měnící se obsah. Zde se jedná především o žurnalistické stránky a magazíny, které poskytují různé aktuální zprávy a zajímavosti o situacích v rozdílných oborech. Obsah je zde poskytován většinou redaktory. Aplikace fungují tak, že po obdržení nové zprávy nebo novinky od redaktora je informace automaticky do aplikace vložena. Důstojným příkladem těchto aplikací je většina portálů, které poskytují aktuální zprávy, dění ve světě nebo v určitých odvětvích. Konkrétně se jedná například o portál zpráv CNN international nebo ekonomický zpravodaj The Economist. (Adobe, 2015)

3.3.2 Priority při tvorbě webových aplikací

Při vytváření webových aplikací by každý vývojář měl dbát určitých základních pravidel, aby jeho aplikace byla použitelná pro uživatele. Zde se jedná především o bezpečnost, která by měla být na prvním místě u každé aplikace. Hrozby jako ztráta dat, zničení dat nebo krádeže dat třetí osobou by se měly co nejlépe eliminovat. Základním aspektem tvorby aplikace je také využívání dostupných zdrojů. Vývojář může použít již hotový kód k řešení určitých známých problémů nebo úkolů v aplikaci. Pokud se aplikace osvědčí v praxi, tak se dost často tvoří její nástavby a případná další vylepšení. Když se s vylepšeními a nástavbami počítá již při návrhu aplikace, práce se obvykle urychlí a ušetří se také nějaké peníze. Dalším důležitým parametrem je rychlost aplikace jako takové. V případě že je aplikace pomalá, tak jí příliš velký počet lidí používat logicky nebude. Nemluvě o tom, že s takovou aplikací mají problémy také internetové

vyhledávače. Všechny skripty probíhající ve webové aplikaci by tedy měly být optimalizovány na rychlost, aby se činnost aplikace nezpožďovala. Měla by se zde také vyskytovat odolnost aplikace vůči maximální zátěži. Maximální zátěž znamená stav, kdy server obsluhuje velké množství klientů najednou a může se stát, že takzvaně spadne. Aby aplikace byla odolná vůči pádům, je potřeba zvolit vhodné nástroje, správně optimalizovat databázi a výpočty v ní probíhající. V poslední řadě by se také aplikace měla podrobit testování očekávaného fungování. Případné nalezení chyb před nasazením aplikace je možnost nalezené chyby včas opravit a zajistit bezproblémový chod aplikace po nasazení. (Adaptic, 2018)

3.4 Jazyk HTML (Hypertext Markup Language)

Jedná se o technologii, pomocí které se vytváří WWW stránky a části webových aplikací. Slouží k tomu, aby obsah na webových stránkách dával smysl. Jedná se o takzvaný značkovací jazyk, který udává strukturu textu na stránce propojeného pomocí hypertextových odkazů. Označuje část textu, která bude nadpisem, odděluje jednotlivé odstavce textu, speciální a další textové prvky. HTML netvoří samotný vzhled textu, který na stránce lze vidět. Neudává webovému prohlížeči nic o velikosti textu, druhu fontu písma ani nespécifikuje konkrétní barvu písma. (DOMES, 2012)

3.4.1 Historie HTML

V roce 1991 vědec z Velké Británie Timothy John Berners-Lee v Ženevě uvedl první verzi 1.0 značkovacího jazyka, dnes známého jako HTML. Tento jazyk byl vynalezen k formátování webových dokumentů. Jazyk byl uveden společně se světovým projektem World Wide Web. V té době HTML nahradilo SGML. To byl také značkovací jazyk, ale byl dost složitý. Mohli ho používat pouze odborníci. HTML bylo vymyšleno s ideou, že ho bude moci používat každý. Oproti SGML měl jeho nástupní jazyk několik výhod. První pozitivní věcí byla jednoduchost. Tagy v HTML byly navrženy jednotným způsobem ve špičatých závorkách na principu počátečního a koncového tagu. Navíc se stránky psané v HTML zobrazovaly mnohem rychleji, než web realizovaný v původním značkovacím jazyku. První stránku pomocí nového značkovacího jazyka vytvořil Timothy v srpnu v roce 1991. Ona první webová stránka je i v dnešní době dohledatelná v historických archivech. V roce 1993 vyšla druhá verze HTML 1.2, která obsahovala více než 40 tagů. Z toho tři z nich naznačovaly vizuální formátování dokumentu a zbytek se zabýval jeho logickou strukturou. O dva roky později v roce 1995 byla vydána verze 2.0. Změn oproti verzi předchozí nebylo mnoho a při vydávání nových verzí

jazyka se nikam příliš nespěchalo. Dotazy byly implementovány na základě vyhledávání klíčových slov a byly zavedeny různé druhy převádění dat z počítače na server. V téže roce byl značkovací jazyk vylepšen na verzi 3.0. Na rozdíl od předchozích variant přidala více použitelných a zajímavých věcí. Například značky pro vytvoření tabulek, dispozice pro matematické vzorce nebo obtékání textu. V průběhu tvorby HTML 3.0 se vývojáři rozhodli, že by nebylo špatné, aby jazyk obsahoval alespoň nějakou možnost, jak ovlivnit grafický design webové stránky. Ve třetí verzi proto byly implementovány možnosti, jak tento požadavek naplnit. V prosinci roku 1996 bylo uvedeno CSS (zkratka znázorňující cascading style sheets). Jednotlivé styly mohly být připojeny na dokumenty HTML a slouží k úpravě vizuální stránky hojně i v dnešní době. Styly jsou na HTML nezávislé a syntaxe je odlišná. CSS umožnilo nastavení parametrů pro vizuální pohled pro každý tag HTML. Díky tomu mohli vývojáři ovlivnit alespoň trochu vzhled stránek například nastavením fontů, velikosti písma, a tak dále. Pro zvýraznění zázraků CSS vytvořil Dave Shea, který pocházel z Kanady, místo na internetu, kde stovky webových vývojářů mohli upravovat stejný HTML dokument různými CSS styly. V lednu v roce 1997, nedlouho po uvedení CSS stylů, přišla na svět verze HTML 3.2. Byla plně kompatibilní s novými vizuálními styly. Některé inovace, které byly implementovány v třetí verzi, byly v následujícím vydání zrušeny. Vývojáři přidali neobvyklé elementy, které byly podporovány s použitím internetových prohlížečů Internet Explorer 3 a Netscape Navigator. Koncem roku 1997 vývojáři vytvořili verzi 4.0. V této variantě byly upraveny elementy, které byly vydány v předchozích verzích. Mnoho tagů bylo označeno jako zastaralé a nedoporučovalo se je příliš používat. Namísto těchto starších tagů bylo nutné využívat CSS styly. Objevila se zde také podpora pro rámečky, skripty a hlavní procedury pro usazení objektů na stránce. Také byly zavedeny tabulky a tvary, které byly výborně přístupné pro handicapované uživatele. Verze byla vytvořena ve spolupráci s experty na poli mezinárodní scény. Od této doby bylo možné tvořit HTML dokumenty v jakémkoliv světovém jazyce a jednoduše je sdílet po celém světě. (Vertex-academy, 2016)

Po uplynulých dvou letech v prosinci 1999 přišla verze HTML 4.01. Jejím symbolem bylo vyladění objektů, obrázků, formulářů a opravení vzniklých chyb v jazyce. Jednalo se o velice stabilní sestavu, která byla používána webovými vývojáři přes deset let. Na konci října 2014 byla vydána poslední verze HTML 5, která je ve webovém vývoji používána do dneška. Byla určena striktní syntaxe jazyka na rozdíl od verzí předchozích. Vylepšila se podpora technologie používaných multimédií. Do jazyka bylo přidáno 28 nových elementů, které dělaly kód více srozumitelným. Byl kladen velký důraz na podporu skriptů jako například Javascriptu. (Washington)

3.4.2 Struktura HTML

V tomto jazyce jsou používány HTML elementy. Ty jsou reprezentovány takzvanými tagy. Každý tag obsahuje název elementu ve špičatých závorkách. Většina tagů v HTML je definována jako párová. To znamená, že první tag v páru je míněn jako začínající tag a druhý jako koncový. Koncový tag vypadá stejně jako počáteční, akorát obsahuje navíc lomítko před svým názvem. Tagy dávají internetovému prohlížeči informaci, jak má zobrazovat obsah určité stránky. Určují, zda určitý text bude nadpis, tabulka, odstavec, a tak dále. Webové prohlížeče tagy použité v HTML přímo na stránce nezobrazují. Používají je k určení, jak mají zobrazovat určitý dokument. (W3schools, 2017)

Ve většině HTML tagů lze také zvolit velké množství atributů. Atributy určují konkrétní vlastnosti elementů a dávají tak informaci internetovému prohlížeči, jak by se měl určitý element zobrazit nebo chovat. Skládají se z názvu atributu a jeho hodnoty ve tvaru atribut = hodnota. Hodnota všech atributů se píše do uvozovek. Pomocí atributů můžeme určit široké spektrum vzhledu prvků, jejich velikosti nebo hodnoty. Spousta HTML elementů mají atributy výchozí a není třeba je specifikovat. Například když vývojář u elementu odstavce <p> neurčí zarovnání podle svých představ, bude odstavec automaticky zarovnán vlevo. Lze samozřejmě zarovnání pomocí atributů změnit dle požadavku. Zarovnání odstavce do bloku realizujeme následujícím způsobem: <p align = "justify">Text odstavce</p>. (Klikzone, 2011)

Na obrázku níže je vidět základní struktura HTML dokumentů. Obvykle se začíná ještě s určením přesné verze jazyka použitím tagu <!DOCTYPE> pro přesné zobrazení v prohlížeči, ale není to nutné. Dále následuje kořenový element <html>, poté hlavička <head> a titulek <title>. Důležitý je zde tag <body>, který je jako jediný zobrazen na webové stránce. Je proto nutné do něj vkládat informace, které mají být pro uživatele viditelné. Ve struktuře je v tomto tagu vložen nadpis s označením <h1> a odstavce vyjádřené tagem <p></p>.



Obrázek 3 – Struktura HTML dokumentu

Zdroj: (Kumarsher45)

3.5 CSS (Cascading Style Sheets)

CSS vzniklo v roce 1997. Jedná se o kaskádové styly, díky kterým lze měnit grafické vlastnosti elementů v HTML na webové stránce nebo ve webové aplikaci. Lze upravovat například barvu textu, pozadí, zarovnání, velikost písma a mnoho dalších věcí, které si můžete představit. (DOMES, 2012)

Existuje více způsobů zápisů CSS, které se v dnešní době nejčastěji používají. Prvním z nich je zápis CSS stylu přímo do elementu, který chceme tímto způsobem formátovat. Docílíme toho použitím atributu style u požadovaného elementu. Dalším cestou, jak vyjádřit styl, je použití tagu <style> přímo v HTML. Ten definuje, jak a co má být přesně naformátováno. Nazýváme ho takzvaným stylpisem. Posledním způsobem je použití externího souboru s příponou .css, ve kterém jsou umístěny konkrétní styly, které na stránce používáme. Dnes se připojení externího souboru používá nejčastěji, jelikož oddělí obsah stránky od vzhledu stránky a zpřehlední tím výsledný kód. Obvykle se k připojení souboru stylů používá tag <link>, díky kterému na soubor odkazujeme. (Jakpsatweb, 2008)

Na přiložených obrázcích níže je uveden příkladný zápis všech tří výše vysvětlených způsobů použití CSS stylů.

```
<li><a href="index.php" style="background-color: #02447E; color: white;">  
<strong>o časopisu</strong><br>  
<span id="condensed">about the journal</span></a></li>
```

Obrázek 4 – Přímý zápis stylu do elementu

Zdroj: Autor

```
<style>  
.grid-container {  
  display: grid;  
  grid-template-columns: auto auto;  
  background-color: #FFFFFF;  
  padding: 10px;  
}  
  
.grid-item {  
  background-color: rgba(255, 255, 255, 0.8);  
  border: 1px solid rgba(0, 0, 0, 0.8);  
  padding: 20px;  
  font-size: 30px;  
  text-align: left;  
}  
</style>
```

Obrázek 5 – Zápis stylů pomocí tagu style

Zdroj: Autor

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

Obrázek 6 – Odkaz na externí soubor styly.css

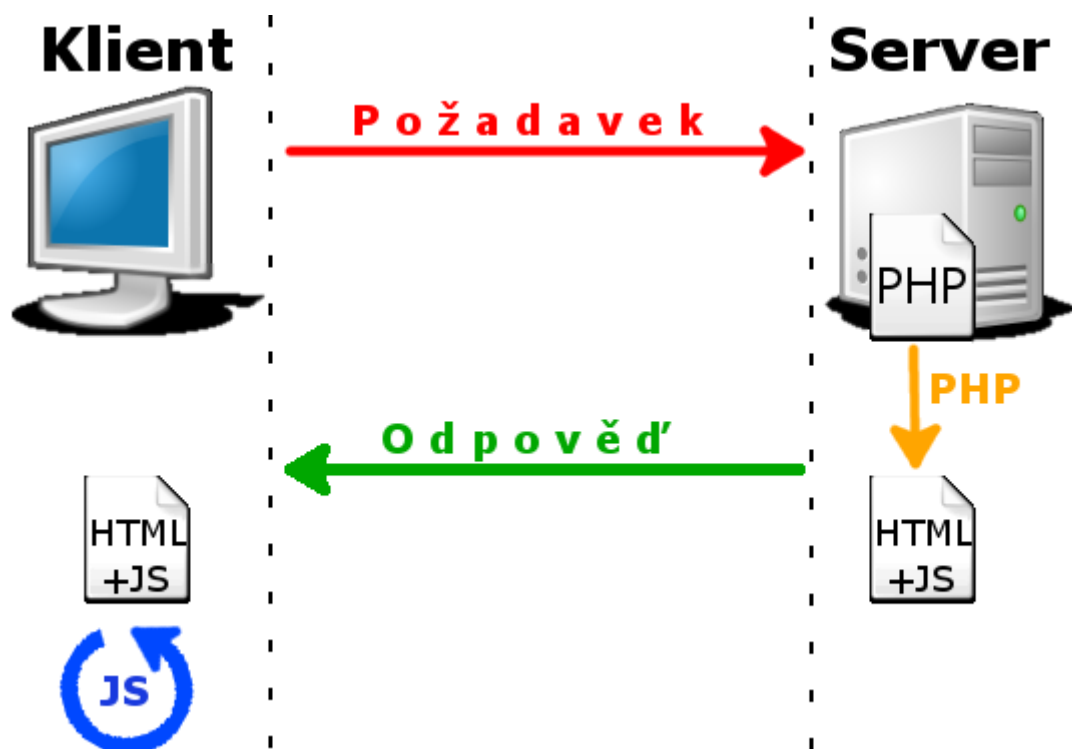
Zdroj: Autor

3.6 PHP

Původní význam zkratky PHP byl Personal Home Page. Nyní tato zkratka znamená Hypertext Preprocessor. PHP bylo založeno v roce 1996 a od doby vzniku jazyk prošel velkými změnami. Jedná se o programovací skriptovací jazyk, který pracuje na straně serveru. Vše spojené s fungováním PHP se odehrává na straně webového serveru, kde jsou uloženy zdrojové kódy stránek. Program napsaný v PHP se nejprve provede na serveru a potom odešle prohlížeči jenom výsledek zadané operace. Například JavaScript, na rozdíl od PHP, vyhodnocuje všechny operace v prohlížeči na straně klienta. Používá se především pro vytváření dynamických internetových stránek. PHP není na pochopení jazyk příliš složitý a v oblasti tvorby webových stránek je široké spektrum situací, kdy je vhodné ho použít. Pomocí tohoto jazyka lze vytvořit například vytvořit knihu návštěv, formulář, anketu, graf a tak dále. PHP také poskytuje možnost propojit vytvořené webové stránky nebo aplikace s databázemi jako například MySQL nebo PostgreSQL. (Tvorba-webu, 2016)

Pomocí jazyka PHP můžeme vytvořit dále například systém komentářů, realizovat registraci a přihlašování uživatelů na stránku a zobrazovat informace na stránce dle toho, kdo stránku navštíví nebo s ohledem na to, s jakými právy je uživatel právě přihlášen. PHP je jazyk pro tvorbu takzvaných skriptů na straně serveru. Skript je druh programu, který popisuje určité chování stránky. Dovoluje přizpůsobit obsah stránky konkrétní situaci a reagovat na chování návštěvníka dle požadavků. Klient je zde webový prohlížeč. Tam se webová stránka zobrazuje. Na druhé straně server je místo, kam jsou veškeré stránky nahrávány a ze serveru jsou stránky načítány na základě požadavku zadaného klientem. Výsledkem práce PHP skriptu je obvykle běžný HTML kód, který se odešle klientovi do webového prohlížeče. (Pehapko, 2018)

Na obrázku níže je zobrazen princip fungování PHP při komunikaci klienta a serveru. Klient odešle svůj požadavek na server. Tam je zpracován pomocí PHP skriptu a ze serveru je odpověď na požadavek klienta odeslána zpět do prohlížeče, kde se návštěvníkovi stránka zobrazí požadované informace.



Obrázek 7 – Princip fungování PHP

Zdroj: (Pehapko, 2018)

3.6.1 Historie PHP

Na počátku zrodu systému figuroval Rasmus Lerdorf. V roce 1994 si Rasmus ve volném čase vytvořil v Perlu jednoduchý systém pro evidenci přístupu k jeho vlastním webovým stránkám. Protože neustálé spouštění interpretu Perlu velice znatelně zatěžovalo server, autor se systémem rozhodl přepsat do jazyka C. (KOSEK, 1999)

Autor sice vytvořil program sice pro svoji vlastní potřebu, ale dovolil ho používat i několika svým známým. Tito lidé si systém velmi oblíbili a zásobovali Lerdorfa novými nápady a připomínkami. Díky této snaze se jim podařilo systém udržet při životě a vývoj na základě jejich poznámek mohl dále pokračovat. Velké oblibě se systém dočkal ve verzi 2.0, která umožňovala velice jednoduchou a intuitivní manipulaci s formuláři. Verze PHP 2.0 byla kvůli tomu označena FI (míněno jako Form Interpreter). (BRÁZA, 2005)

V roce 1998 byla do světa vpuštěna verze PHP 3.0. Na rozdíl od předchozích dostupných verzí programu byla nová konfigurace obohacena především o podporu širokého spektra databázových systémů, možnost použití objektů a cookies. Oproti předchozím variantám byla práce PHP také podstatně zrychlena. V této době se PHP velmi rychle rozšířilo

do světa a začalo se hojně používat na řadách obrovských serverů jako například zpravodajské servery, archivy a tak dále. (BRÁZA, 2005)

Zanedlouho po vydání verze 3.0 začali v zimě roku 1998 Zeev Suravski a Andi Gutmans přepracovávat jádro PHP. Hlavními cíli změny bylo zlepšit výkonnost a komplexnost aplikací a obecně modularitu základního kódu v PHP. Ačkoliv verze 3.0 nabízela spoustu nových funkcí včetně připojování databázi a aplikačních rozhraní třetích stran, tak tyto prvky nevyužívala příliš efektivně. Nový „engine“ označený jako Zend (podle prvních dvou písmen z křestních jmen autorů) byl uveden v polovině roku 1999. Oficiálně bylo PHP 4.0 vydáno v květnu roku 2000 po téměř dvou letech práce. Verze vylepšila výkonost oproti předchozí a obsahovala další klíčové prvky. Jednalo se zejména o práci se sessions, spolupráci s velkým množstvím webových serverů, vylepšené ošetření uživatelských vstupů a v neposlední řadě také nabídla několik nových jazykových konstruktů. (Php, 2017)

V létě 2004 byla po roce testování oficiálně vydána první verze PHP 5. Novější varianta přinesla mnoho změn, které stojí za zmínku. Nutno dodat, že tyto změny neměly vliv na zpětnou kompatibilitu. Došlo ke kompletnímu přepracování systému objektů. Objekty se staly inteligentní součástí jazyka a nabízejí zde možnosti téměř srovnatelné s objektově orientovanými programovacími jazyky. Vylepšily se zde režimy přístupu k objektům, přišla možnost tvorby konstruktorů a destruktorů a volání objektů odkazem. Další důležitou inovací je zavedení jmenných prostorů (takzvaných „namespaces“). (BRÁZA, 2005)

Jako PHP 6 byla označována nakonec nikdy nevydaná verze PHP, která měla přinést podporu Unicode. Aby se předešlo zmatkům s již vydanými články a knihami o PHP 6, bylo rozhodnuto novou verzí označit jako PHP 7. (HUJER, 2015)

Poslední verzí PHP, která doposud zatím byla vydána je PHP 7. Bylo dosaženo dalšího zrychlení v jazyce a ušetření paměťových nároků na aplikace. Zavedena byla také typová kontrola pro skalární datové typy. Také v této konfiguraci se programátorům naskýtá možnost zapsat návratovou hodnotu funkce nebo metody přímo do její definice, což v předchozích verzích nebylo možné. Pokud funkce vrátí jiný typ hodnoty, než je definováno, PHP nám nahlásí chybu. Jádro PHP bylo také upraveno, aby místo fatálních chyb zobrazovalo výjimku. Systém chyb a výjimek byl kompletně přepracován ke větší srozumitelnosti a logice. Vypisování některých chyb bylo definitivně zrušeno. Uveden byl také nový trojcestný porovnávací operátor nazvaný „Spaceship“, v kódu označený `<=>`. Funguje na principu, že pro stejné hodnoty vrátí nulu, pro první menší hodnotu vrátí -1 a pro první větší hodnotu jedna. Přibyly také dvě funkce pro generování náhodných bajtů a náhodných čísel, jelikož v předchozích verzích bylo realizování těchto operací poměrně složité. (HUJER, 2015)

3.6.2 Základní syntaxe PHP

PHP je strukturovaný programovací jazyk. To znamená, že příkazy, které spolu nějakým způsobem souvisí lze sdružovat do bloků, které tvoří strukturu programu a obsahují více než jeden příkaz. Ideálním příkladem použití bloků je ten, když se program nějakým způsobem větví dle určité podmínky. Podmínka se nejprve vyhodnotí. Pokud je splněna, následuje blok příkazů v případě platnosti podmínky. V opačné situaci se provede blok příkazů, když podmínka není platná. Pokud není definováno, co se má stát, když podmínka neplatí, program běží dále s příkazy následujícími. Bloky kódu jsou často nezbytné, protože jeden příkaz často nestačí k obslužení požadované operace. Jsou často používány také například k ohraničení funkce nebo těla cyklu. Aby se daly v PHP bloky dobře rozlišit, tak jsou umístěné do složených závorek. (BRÁZA, 2005)

Jednotlivé části kódu v PHP je nutno uzavírat pomocí identifikátorů k tomu sloužících. Nejjednodušší a nejčastěji využívaný způsob je vidět i na obrázku níže. Za pomoci špičatých závorek se přehledně dozvíme, kde začíná a končí PHP kód. Koncovou značku lze v určité situaci vynechat. Můžeme to udělat tehdy, když soubor končí PHP kódem. V případě neuvedení koncové značky je také výhodou, že pokud se v souboru na konci nachází nějaké zapomenuté bílé značky (například mezery, tabulátory nebo konce řádků), tak se tyto znaky nevypíší. Jednotlivé příkazy v PHP se ukončují středníkem. Ten je sice před koncovou značkou nepovinný, ale jeho vynechávání je dobré se vyhnout. Je tomu tak, protože to poté znesnadňuje přidávání dalších příkazů na konec bloku v PHP kódu. (VRÁNA, 2012)

Na obrázku níže je uveden příklad bloku příkazů v PHP. Poměrně jednoduchý blok realizuje odhlášení uživatele ze stránky a přesměrování na jinou stránku po úspěšném provedení odhlášení.

```
<?php
session_start();
unset($_SESSION["id"]);
unset($_SESSION["jmeno"]);
unset($_SESSION["admin"]);
unset($_SESSION["stahovani"]);
unset($_SESSION["schvaleni_registrace"]);
header("Location:login.php");
?>
```

Obrázek 8 – Příklad bloků příkazů v PHP

Zdroj: Autor

3.6.3 Kombinace PHP a HTML kódu

Kód psaný v PHP není potřeba mít pouze na jednom místě na stránce. Podobně jako například v JavaScriptu lze kód PHP kdykoliv ukončit a pokračovat s ním později. Mezitím lze pracovat v HTML kódu a k PHP se kdykoliv vrátit a vkládat v něm další příkazy. (BRÁZA, 2005)

PHP a HTML kód lze kombinovat, samozřejmě pouze za předpokladu, že jsou jednotlivé části kódu náležitě a správně odděleny. PHP kód je nutné relevantně oddělit počátečním a koncovým tagem od HTML a kódu napsaném v jiných programovacích jazycích. Do PHP kódu můžeme zanořit HTML kód, ale lze také udělat i opak. To znamená, že do HTML kódu lze vložit část kódu napsanou v PHP. (Squelle, 2018)

Na následujícím obrázku můžeme vidět HTML kód, který je součástí kódu v PHP. V každém příkazu echo, který slouží k zobrazení požadovaného výstupu je HTML kódem specifikováno, jak má daná součást vypadat a zobrazit se na výsledné stránce.

```
if ($_SESSION["stahovani"] == 'ano')
{
echo "<table><col><col><tr>\n";
echo "<td class='col-issue'>";
echo "<a href='archive/" . $filename . ".pdf'>";
echo "<img class='img-issue' src='archive/" . $filename . ".jpg'></a><br>";
echo "<a href='archive/" . $filename . ".pdf'>PDF format</a>";
echo "</td>\n<td>";
}
else {}
$koniecTablice = "</td></tr></table>\n\n";
,
```

Obrázek 9 – HTML kód v PHP kódu

Zdroj: Autor

Na dalším obrázku lze vidět opačný případ. Jedná se o PHP kód vložený do HTML kódu.

```
<form action="<?php echo $_SERVER['PHP_SELF']?>" method="post">
<fieldset>
<legend>Přihlášení uživatele</legend>
<p><input name="jmeno" size="20" tabindex="1" type="text" /> <label>Jméno</label></p>
<p><input name="heslo" size="20" tabindex="2" type="password" /> <label>Heslo</label></p>
<p><input name="submit" type="submit" tabindex="3" value=" Přihlásit " /></p>
</fieldset>
</form>
```

Obrázek 10 – PHP kód vložený do kódu HTML

Zdroj: Autor

3.6.4 Výhody PHP

Skriptovací jazyk PHP může nabídnout řadu výhod, proč si ho vybrat a z jakého důvodu je dobré s ním pracovat. Prvním velkým pozitivem je, že je velice snadné se v PHP naučit programovat. Není potřeba, aby měl člověk velké množství zkušeností v oblasti webů a programování předtím, než se pustí do práce s PHP. Syntaxe jazyka se příliš neliší od jazyka C nebo C++ a je tudíž velkému množství lidí blízká. Programovací jazyk nabízí široké spektrum souvisejících technologií, formátů a standardů. Na webu lze dohledat spoustu návodů na vyřešení již známých problémů, se kterými se může člověk jako PHP programátor setkat. Program si také velmi obstojně rozumí se servery Apache, jelikož je to v podstatě sesterský projekt PHP. Nespornou výhodou je také jednoduché připojení k různým databázím. Nejčastěji se jedná o databáze MySQL, PostgreSQL, Oracle, MS SQL a mnoho dalších. Dalším pozitivním faktorem je to, že jazyk PHP je multiplatformní, lze ho provozovat na valné většině webových serverů a operačních systémů, které jsou dnes k dispozici. Posledním pozitivem je to, že PHP je v dnešní době podporováno téměř všemi poskytovateli webhostingu. Velká část z nich podporuje také kombinaci PHP a MySQL zcela zdarma. Tudíž se na tomto řešení dají i ušetřit nějaké finance. (PROCHÁZKA, 2012)

3.6.5 Nevýhody PHP

Ačkoliv je PHP často používaným programovacím jazykem, tak práce s ním také přináší nějaký nevýhody. První slabinou PHP je slabší bezpečnost. Vzhledem k tomu, že jazyk PHP je opensource, tak každý člověk může vidět zdrojový PHP kód, pokud k němu má přístup. Může také nahlédnout do vašich PHP kódů kdokoliv, kdo má přístup k serveru. Pokud jsou ve zdrojovém kódu nějaké chyby nebo bugy, tak lidé mohou odhalit tyto slabiny a zneužít je. Dalším problémem je nevhodnost použití jazyka pro aplikace obrovského rozsahu. Vzhledem k tomu, že programovací jazyk není tak přísně standardizovaný, aplikace obrovského rozsahu by bylo dost těžké nějak rozumně udržovat. Následujícím neduhem jazyka je nepřímá konverze datových typů. Neopatrné programátory může překvapit a vést k neočekávaným chybám. Mírné nesrovnalosti také nastávají při použití polí a hashových tabulek. Nepříjemná věc se také týká hlášení chyb v PHP. Není zcela korektně vyřešeno z pohledu vývojářů. Ačkoliv zkušenější PHP programátoři si jsou schopní tento problém překonat. PHP nikdy nebude mít takový výkon jako například C nebo C++. Jelikož je to skriptovací a interpretovaný jazyk, tak vždy bude o něco pomalejší než aplikace optimalizované v C nebo C++. (Phpbabu, 2017)

3.7 Python

V odborném pojetí je Python objektově orientovaný vysoce postavený programovací jazyk s integrovanou dynamickou sémantikou. Je dobře použitelný pro rychlý vývoj aplikací, protože nabízí dynamické typování a možnosti vytváření dynamických vazeb. Python je relativně jednoduchý na naučení. Vyžaduje ale znalost specifické syntaxe, která je primárně zaměřena na čitelnost zdrojového kódu. Vývojáři by díky tomu měli lépe rozumět kódu napsaném v Pythonu než v jiných programovacích jazycích. To by ve výsledku mělo zredukovat náklady na vývoj a údržbu aplikací psaných v Pythonu. Syntaxe nemá nicméně tolik společného s notoricky známými konstrukty v jazycích C nebo C++. Navíc Python podporuje široké spektrum modulů a balíčků, které umožňuje skládat kód programu dohromady a může být navíc použit vícekrát v různých aplikacích. Pokud se modul nebo balíček jednou naprogramuje může být snadno importován nebo exportován do aplikace. (Pythonforbeginners, 2018)

3.7.1 Historie Pythonu

Python byl nejprve navrhnout holandským počítačovým programátorem Guidem van Rossumem v roce 1991 a vyvinut programovou společností Python. Jazyk Python vznikl s důrazem na čitelnost kódu a jeho syntaxe měla dovolovat vývojářům vyjádřit své myšlenky v pár řádcích kódu. Implementaci Pythonu začal Guide van Rossum už v roce 1989 v centru Wiskunde & Informatica (CWI) nacházející se v Nizozemsku. Původně se jednalo jen o zájmový projekt, kterého se chtěl Rossum účastnit přes zimu. Za předchůdce dnešního Pythonu je brán ABC programovací jazyk, který byl vyvinut s rozhraním Amoeba Operating system. Python byl ovlivněn programovacími jazyky ABC a Modula-3. Inspirací pro název jazyka se stala televizní show vysílaná na programu BBC „Monty Python’s Flying Circus“. (Geekforgeeks)

V roce 1994 byla vydána druhá verze Python 1.0, která obsahovala některá vylepšení. Konkrétně se jednalo o rozšířené zobrazení, filtry a zmenšení. Do další konfigurace s označením 2.0, která byla vydána v polovině října roku 2000 byly přidány systémy kolekcí. Za další důležitou verzi se považuje Python 3.0, který byl uveden na začátku prosince roku 2008. Verze dala do pořádku základní nedostatky jazyka. (Javatpoint, 2016)

Na obrázku níže můžete vidět jednotlivé verze Python, které byly doposud uvedeny.

Python Version	Released Date
Python 1.0	January 1994
Python 1.5	December 31, 1997
Python 1.6	September 5, 2000
Python 2.0	October 16, 2000
Python 2.1	April 17, 2001
Python 2.2	December 21, 2001
Python 2.3	July 29, 2003
Python 2.4	November 30, 2004
Python 2.5	September 19, 2006
Python 2.6	October 1, 2008
Python 2.7	July 3, 2010
Python 3.0	December 3, 2008
Python 3.1	June 27, 2009
Python 3.2	February 20, 2011
Python 3.3	September 29, 2012
Python 3.4	March 16, 2014
Python 3.5	September 13, 2015
Python 3.6	December 23, 2016
Python 3.7	June 27, 2018

Obrázek 11 – Jednotlivé verze jazyka Python

Zdroj: (Javatpoint, 2016)

3.7.2 Základní syntaxe Pythonu

Jazyk Python je trochu podobný ostatním podobným jazykům jako například Java, Perl, nebo C. Na rozdíl od C++ nebo PHP Python neodděluje bloky kódu pomocí závorek. Toto je realizováno pouze pomocí odsazení jednotlivých částí kódu od kraje řádku. Tento systém se může stát velmi zmatečný. Člověk na první pohled nevidí, kdy jednotlivý blok končí, protože

není od ostatních nijak viditelně oddělen. Systém je v syntaxi jazyka striktně dodržován. Například počet mezer v odsazení symbolizuje proměnnou a jiná vzdálenost odsazení vyjadřuje hodnoty dané proměnné. (Tutroialspoint, 2016)

V Pythonu není potřeba u proměnných deklarovat přesný datový typ. U každého přiřazení hodnoty do určité proměnné si Python sám zjistí, jakého datového typu hodnota je a zapamatuje si podstatnou informaci pro další použití. V jazyce je podporováno široké spektrum datových typů, které se hojně používají i v jiných programovacích jazycích. Najdeme zde logickou hodnotu bool (true nebo false), poté čísla (celá, reálná, zlomky), seznamy, N-tice, řetězce, bajty a pole bajtů, množiny nebo slovníky. (PILGRIM, 2010)

Na přiloženém obrázku níže vidíme ukázkou zdrojového kódu napsaném v jazyce Python. V kódu se projevují veškeré vlastnosti, které byly zmíněny. Struktura je dána pomocí odsazování jednotlivých částí od okraje a jednotlivé příkazy nejsou ukončeny středníkem. To může vést k nižší přehlednosti.

```
503         message =
504         if not hasattr(self, '_headers_buffer'):
505             self._headers_buffer = []
506             self._headers_buffer.append((" %s %d %s\r\n" %
507                 (self.protocol_version, code, message)).encode(
508                     'latin-1', 'strict'))
509
510     def send_header(self, keyword, value):
511         """Send a MIME header to the headers buffer."""
512         if self.request_version != 'HTTP/0.9':
513             if not hasattr(self, '_headers_buffer'):
514                 self._headers_buffer = []
515                 self._headers_buffer.append(
516                     ("%s: %s\r\n" % (keyword, value)).encode('latin-1', 'strict'))
517
518             if keyword.lower() == 'connection':
519                 if value.lower() == 'close':
520                     self.close_connection = True
521                 elif value.lower() == 'keep-alive':
522                     self.close_connection = False
523
```

Obrázek 12 – Ukázka kódu v Pythonu

Zdroj: (Live.staticflickr)

3.7.3 Výhody Pythonu

První výhodou jazyka je ten, že podporuje rozsáhlé knihovny, které obsahují mnoho použitelných věcí. Jedná se zejména o operace s řetězci, nástroje pro webové služby a internet, rozhraní operačních systémů a protokoly. Mnoho nejčastěji používaných programátorských úloh jsou již v těchto knihovnách předem zapsané a lze je ihned v Pythonu použít. Python je také dobře integrován s ostatními jazyky. Je vcelku jednoduché vyvíjet webové služby v Pythonu a odkazovat se na komponenty COM a COBRA. To nabízí vývojářům možnosti použití C, C++ nebo Javy v Pythonu. Jazyk také zpracovává XML a ostatní značkovací jazyky. V poslední řadě se Python vyznačuje efektivním programováním. Integrované prvky, frameworky a prvky vylepšení kontroly přispívají k vyšší rychlosti a produktivitě vytvořených aplikací. (Medium, 2017)

3.7.4 Nevýhody Pythonu

Nepříjemností v tomto jazyce je bezpochyby pomalá rychlost spouštění aplikací. Python se řadí mezi interpretované jazyky. To znamená, že pracuje s interpretem, nikoliv s kompilátorem jako například C, C++ nebo Java. V důsledku tohoto faktu je spouštění aplikací pomalejší než v kompilovaných jazycích. Dalším neduhem je zde vyšší spotřeba místa v paměti. Jazyk není příliš vhodný pro vývoj s omezenou paměťovou kapacitou. Jazyk je především používán pro vývoj desktopových aplikací a webových server aplikací. Vzhledem k větší konzumaci paměti a pomalejšímu zpracování aplikace není nejlepší volbou pro mobilní aplikace a pro vývoj počítačových her ve srovnání s ostatními jazyky. Když si vývojář zvolí pracovat v Pythonu a poté bude chtít své rozhodnutí změnit, tak to nebude nejjednodušší. Také v Pythonu není až tak jednoduché rozeznat chyby. Je to způsobeno z důvodu, že jazyk interpretovaný. Pro vývojáře není nic příjemného, když odhalování chyb zabere delší dobu, než je nezbytně nutné. V poslední řadě je v Pythonu problematictější přístup k databázím. Ve srovnání s ostatními databázovými technologiemi vrstva Pythonu přístupu k databázi je lehce zaostalá a primitivní. Je tomu tak z důvodu, že není vhodné použití snadné interakce komplexních složitých dat. (Squareboat, 2019)

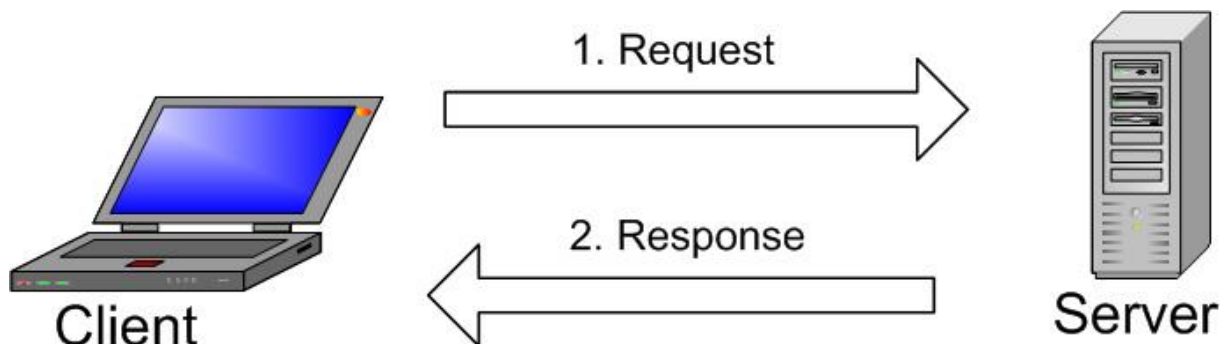
3.8 Databáze

Pod pojmem databáze si představíme nějaký soubor strukturovaných informací nebo také dat, která jsou ukládána ve virtuální podobě do počítačového systému. Databáze je nejčastěji vedena systémem pro řízení báze dat. Často je používána pro něj zkratka v češtině SŘBD. V angličtině je využívána zkratka DBMS, která znamená database management system. Data, která jsou uložena v databázi, jsou téměř vždy uskupována do databázových tabulek. Každá taková tabulka je uspořádána do řádků a sloupců. Tato struktura umožňuje efektivní a rychlé zpracování a tvoření databázových dotazů. Zajištěn je zejména jednoduchý přístup k uloženým datům, jejich správa, řízení, případná aktualizace a organizace shromážděných informací. (Oracle)

První databázové systémy se objevily na počátku šedesátých let minulého století. Databázový systém se skládá z databáze a systému řízení báze dat. Protože databázové systémy pracují každý den s velkým množstvím perzistentních dat, je třeba, aby uměly určitě specifické věci. Systém by měl podporovat definici pro datový model (nejčastěji relační, případně logický). Dále je zapotřebí využít některý jazyk pro manipulaci s daty. Nejčastěji se používá jazyk SQL, případně QBE. Systém by také měl nabízet správu datových transakcí, autorizaci uživatelů a autorizaci operací prováděných nad daty. Samozřejmostí by také měla být robustnost a zotavení po chybách bez ztráty uložených dat. (Sprava-site, 2016)

3.8.1 MySQL

Jedná se o open source relační databázový systém, který je postavený na modelu klient-server. Používá se pro vytváření a upravování tabulek založených na relačním modelu. MySQL vyvinula v roce 1994 švédská společnost MySQL AB. V roce 2008 americká technická firma Sun Microsystems převzala veškerá vlastnická práva tím, že společnost koupila. Nakonec v roce 2010 obrovská firma Oracle koupila Sun Microsystems a od té doby je MySQL vlastnictvím firmy Oracle dodnes. Tento systém je dnes asi nejvíce používaný, když potřebujeme k něčemu využívat databázi. To má několik důvodů. Prvním z nich je velká flexibilita, jednoduchost používání a krátká intuitivní instalace. Další velkou výhodou je rychlost práce s daty. Ani v případě obrovského množství dat se nemusí velké společnosti bát využít MySQL, jelikož jim zajistí dostatečnou rychlost přenosu dat a výbornou dostupnost dat. Navíc je toto řešení vhodné i z hlediska bezpečnosti. Podporuje správu uživatelských účtů a nastavení práv, která udrží data v bezpečí. (Hostinger, 2020)



Obrázek 13 – Princip fungování MySQL

Zdroj: (Hostinger, 2020)

3.8.2 PostgreSQL

PostgreSQL vznikl v roce 1986 jako součást projektu POSTGRES na univerzitě v Kalifornii v Berkley. Byl vyvíjen více než 30 let na různých platformách. Jedná se o multiplatformní objektově relační databázový systém. Ten si získal nejvyšší reputaci na základě své osvědčené architektury, spolehlivosti, datové integrity PostgreSQL disponuje širokým množstvím implementovaných prvků, které pomáhají vývojářům při tvorbě aplikací a administrátorům jsou užitečné k ochraně integrity dat a vybudování stabilního fungujícího prostředí. Některé prvky způsobují mírnou změnu syntaxe oproti standardnímu SQL, ale rozdíly nejsou tak často k vidění. Nejnovější verze 12 byla vydána v říjnu roku 2019 a podporuje 160 ze 179 povinných prvků z jádra SQL:2016. (Postgresql, 2018)

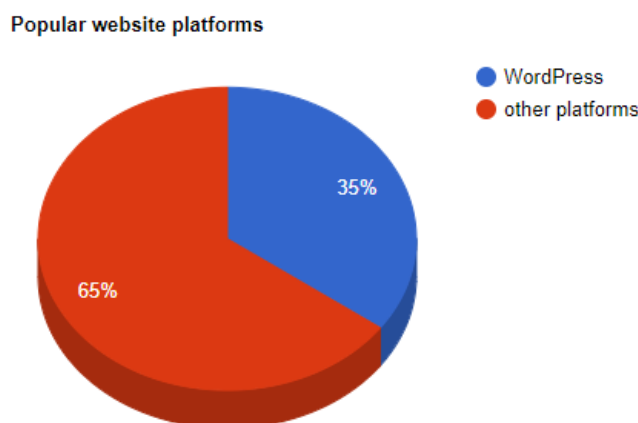
3.9 WordPress

WordPress je jedním z nejpoužívanějších redakčních systémů na světě, který slouží pro tvorbu webových stránek. Pomocí tohoto nástroje může téměř každý vytvořit internetové stránky jakéhokoliv stylu. Z hlediska složitosti se může jednat o jednoduchý blog, ale také například o komplexní stránky pro podnikání, jako třeba online obchod s použitím WooCommerce pluginu, který WordPress podporuje. Jeho použití je také velmi vhodné pro lidi, kteří nikdy nepracovali s kódem nebo profesionálním designem. V redakčním systému je obsaženo široké spektrum šablon a připravených hotových vzhledů stránek, které je možné použít. V tomto směru nemusí autor stránek tušit zcela nic o HTML a nemusí přitom napsat ani řádek kódu. (Wpapprentice.com, 2016)

WordPress byl založen v roce 2001 skupinou zapálených lidí okolo dvou hlavních lidí (Mike Littla a Matta Mullenwega), kteří zprostředkovali strukturovanou, dobře upravitelnou a rozšiřitelnou platformu. Původním účelem byla tvorba a úprava malých osobních webů a blogů pro vlastní potřeby. Nyní však využívá tuto platformu široké spektrum společností, a některé z nich jsou i světoví magnáti v určitých oborech. Na rozvoji se podílí stovky programátorů a nepřímo se na vývoji podílí desetitisíce lidí, jelikož platforma je open source. Dalším pozitivem je to, že tento systém je dostupný zcela zdarma. (Wedesin.cz, 2016)

Redakční systém je populární mezi jednotlivci, velkými podniky a všemi ostatními zapadajícími mezi tyto dvě kategorie. Webové stránky založené na WordPressu využívá například vládní bílý dům v Americe, gigant v oblasti IT společnost Microsoft a také velmi známá hudební skupina The Rolling Stones. (Kinsta.com, 2020)

Na obrázku níže můžete vidět, jak populární WordPress při tvorbě webových stránek je. Na celém internetu na tomto redakčním systému běží přibližně 35 procent všech stránek.



Obrázek 14 – Využití WordPressu na webu celkově

Zdroj: (Themeisle.com, 2020)

3.9.1 Výhody WordPressu

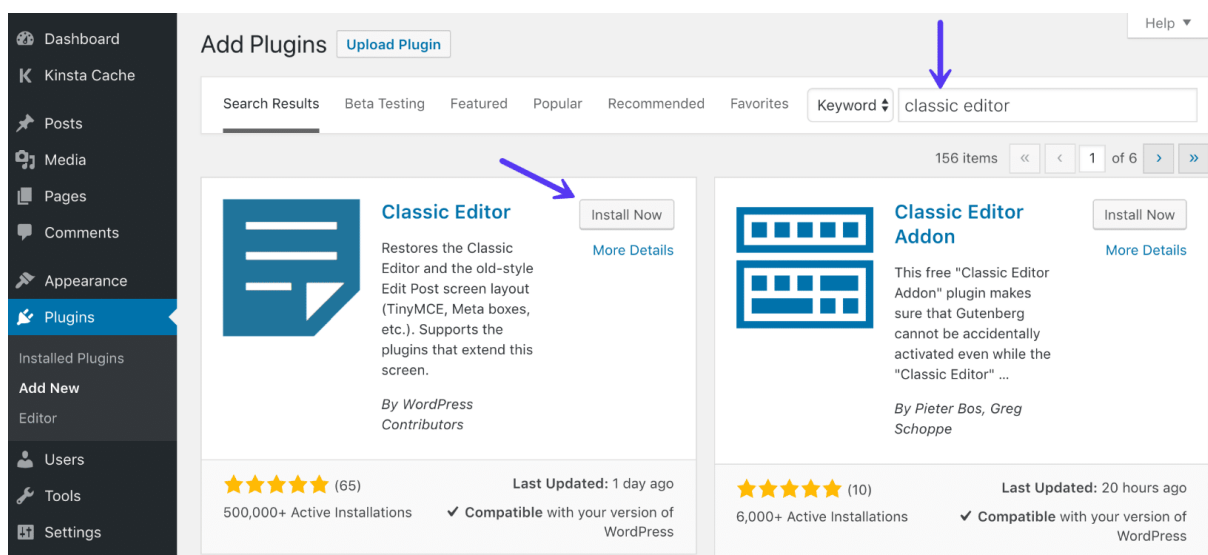
Mezi nesporné výhody patří to, že redakční systém je zcela zdarma a v licenci open source. Na základě tohoto faktu mohou společnosti ušetřit velké množství finančních prostředků. Díky open source licenci se na vývoji podílí velké množství lidí a systém se pravidelně zdokonaluje. Wordpress je také moderní a všestranný. Dají se díky němu tvořit jakékoliv stránky. Je jedno, jestli stránka bude blog, fotografická stránka, slovník nebo elektronický obchod. Systém si

poradí se všemi zmíněnými příklady. Prostředí systému je dobře přístupné a uživatelsky přívětivé. Můžou ho snadno využívat i handicapovaní lidé s vadami zraku nebo jiným postižením. V poslední řadě je ve WordPressu dostupné obrovské množství pluginů, které funkce systému rozšiřují. Tato rozšíření jsou použitelná v širokém spektru případů a díky nim je také tento redakční systém tak populární. (Themeisle.com, 2020)

3.9.2 Nevýhody WordPressu

Ačkoliv má tento redakční systém velké množství pozitivních vlastností, i zde nalezneme několik negativ. Prvním riziko je skryto v bezpečnosti stránek, které na WordPressu běží. Díky své popularitě jsou stránky založené na tomto systému cílem hackerů, jelikož jádro fungování je na každém webu stejné. Tento problém lze řešit správným zabezpečením a pravidelnými zálohami. Dalším problémem je zde spam, také kvůli vysoké popularitě. Další slabinou stránek, vytvořených tímto způsobem, je rychlost přístupu. Tímto nešvarem trpí víceméně všechny redakční systémy. Správným nastavením lze ztrátu rychlosti na webu znatelně snížit. Posledním negativem je to, že stránky vytvořené pomocí WordPressu jsou vcelku náročné v požadavcích na server. (Wedesin.cz, 2016)

Na obrázku níže je zobrazeno rozhraní redakčního systému WordPress.



Obrázek 15 – Ukázka rozhraní WordPressu

Zdroj: (Metropoliscreative, 2019)

3.10 OJS

Open journal systems je redakční systém, který funguje v licenci open source. Primárně je určený pro správu a publikování elektronických časopisů. Svým uživatelům zprostředkovává přehledné a komfortní prostředí s možností automatického nastavení opakovaných redakčních činností, vytvoření efektivní webové prezentace časopisu včetně online publikování plných textů článků s mnoha možnými funkcemi. (Knihovna.cvut, 2019)

Původně byl tento systém vyvinut v roce 2001 skupinou PKP. Prvotním účelem bylo vylepšení přístupu k materiálům sloužícím k výzkumu. Nyní je OJS jedním z nejpoužívanějších open source softwarů ke zveřejňování elektronických verzí časopisů. Využívá ho okolo deseti tisíc časopisů na celém světě. OJS je komplexní nástroj pro spravování pro návrh a editaci článků a publikování celých článků online. Nabízí responzivní čtenářský front-end s možností výběru jednoho ze tří vzhladů. Dále umožňuje flexibilní možnost editace článků s nastavitelnými možnostmi konfigurace. Všechn obsah, který je vložen uživatelem systému, je dostupný a editovatelný online. Jsou také integrovány odborné služby zajišťující publikování jako například Crossref, DOAJ nebo ORCID. Vše je lokálně nainstalováno a pravidelně kontrolováno, a navíc je dostupná také podpora v případě problémů uživatelů. K systému se také dají snadno dohledat různá instruktážní videa a tutoriály. OJS je také dostupné ve třiceti jazycích. Může ho tedy využívat opravdu široké spektrum lidí po celém světě. Skupina PKP také nabízí placenou verzi produktu, která umožňuje mít hosting a instalaci OJS společně s pravidelným zálohováním uložených dat, lepší zajištění bezpečnosti než u neplacené verze včetně upgradů a patchů. Navíc lidé s verzí za peníze docílí toho, že jejich otázky na podporu budou zodpovězeny prioritně. Všechny peníze, které skupina z placeného produktu získá, jsou zpět investovány do jejího vlastního rozvoje. (Pkp.sfu, 2014)

3.10.1 Výhody OJS

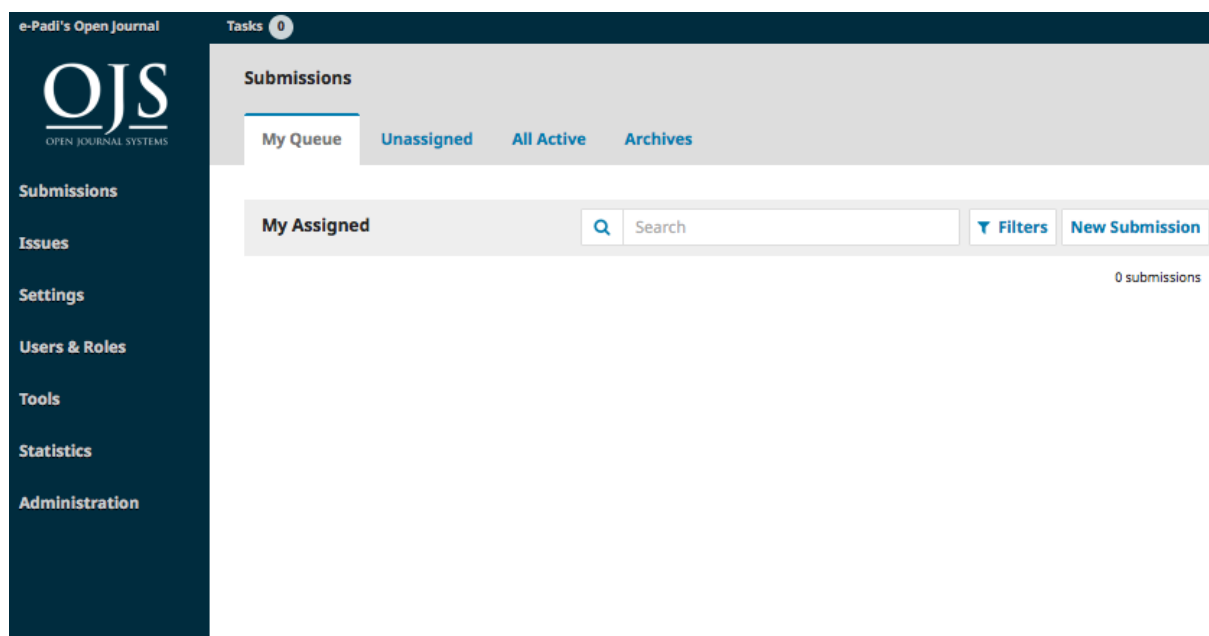
Prvním pozitivem je samozřejmě to, že se jedná o open source software. Není potřeba za něj platit, je dostupný zcela zdarma s možnou editací. Lze sice využít placenou verzi k podpoře společnosti, ale není to nutné. Jednu instalaci Open Journal Systems lze využít pro více časopisů zároveň s možností odlišení designu i funkcionalit u jednotlivých časopisů. Systém umožňuje online přihlašování, zadávání příspěvků a také disponuje přehledným průběhem recenzního řízení. Probíhá také archivace veškerých dokumentů od zadání příspěvku až po jeho opublikování, která bude zpětně dohledatelná a dostupná. Lze také snadno přidělit role a práva všem uživatelům. Velkým pozitivem je, že je umožněno automatizovat jednotlivé redakční

procesy jako například notifikace, odesílání e-mailů, hlídání termínů uzávěrek nebo přidělování recenzentů. Je možné také pracovat s uživatelskými standardy a používat licenci Creative Commons. Celkem jednoduchý je také export dat pro použití v databázi. V poslední řadě lze také nastavit skrytou zkušební verzi pro nového uživatele a komplexně indexovat obsah celého systému. (Knihovna.cvut, 2019)

3.10.2 Nevýhody OJS

Podobně jako ostatní komerční řešení má také OJS některé nevýhody použití. V systému se vyskytují někdy chyby a bugy, které jsou v případě tohoto redakčního systému odstraňovány komunitou používajících uživatelů. Tento postup nemusí vždycky úplně dokonale fungovat. Ačkoliv OJS vyniká v jednoduché administraci vydávání časopisů a publikačních činností, jsou v něm stále používány proprietární textové editory jako například Microsoft Word, Adobe InDesign nebo Adobe Professional. Místo toho je možné využít otevřené nástroje jako OpenOffice.org nebo LaTeX. (Linuxexpres, 2012)

Na obrázku níže můžeme vidět rozhraní Open Journal Systems. Konkrétně se jedná o administrátorské menu.



Obrázek 16 – Ukázka administrátorského rozhraní OJS

Zdroj: (Help.e-padi, 2019)

3.11 Metoda funkčních bodů

Jedná se o metodologii, která slouží k odhadu velikosti kódu, který bude potřeba k realizování určité aplikace. Výsledné vypočtené číslo pak určuje předpokládaný počet řádků zdrojového kódu, který bude pravděpodobně potřeba použít k implementaci určitého zadání. Funkční body jsou zde jednotkou, která slouží k měření. Základní myšlenkou funkčních bodů je, že jsou definovány z pohledu uživatele aplikace. Uživatel je osoba, která jakýmkoliv způsobem komunikuje nebo interaguje s aplikací v nějakém čase. Uživatelské funkční požadavky popisují, co by program měl dělat ve smyslu úkolů a služeb. Do funkčních bodů je započítán počet uživatelských vstupů a transakcí, počet výstupů a transakcí, celkový počet dotazů, celkový počet souborů a počet externích rozhraní, ke kterým se aplikace připojuje (databáze, webový server). Všechny tyto věci jsou zohledněny ve výpočtu počtu funkčních bodů. Poté co získáme výsledný počet funkčních bodů, ho vynásobíme průměrnou délkou kódu na jeden funkční bod v použitém jazyce. Tím získáme celkový odhad délky aplikace nebo jejích funkcionalit v řádcích kódu. (Functionpoints)

3.12 COCOMO model

Tato metodologie vychází z anglické zkratky Constructive Cost Model. Jedná se o regresní model, který na základě počtu tisíců řádků kódu odhaduje počet času potřebného k vyhotovení prací na projektu. Model byl navrhnut v roce 1970 Barry Boehmem a dnes je brán za jeden z nejlépe dokumentovaných modelů. Klíčovými faktory, které definují kvalitu SW produktů, jsou zde úsilí (effort) a časový plán (schedule). Úsilí nám ukazuje množství práce, která je nutná pro dokončení zadání. Výsledek je reprezentován v takzvaných člověkoměsících, kde jedna tato jednotka je 140 hodin. Další podobnou jednotkou je člověkoden (man-day), který čítá osm hodin. Časový plán vychází z úsilí a udává čas, za který bude projekt dokončen při vynaloženém úsilí. Podle definice Boehma jsou projekty rozděleny do třech kategorií. Pro každou jednotlivou kategorii jsou ve vzorcích použity jiné koeficienty. První kategorií je organický projekt, kde je malý vývojový team a řeší se většinou již v minulosti známé problémy a úkoly. Dalším druhem projektu je řadový. U něj je tým developerů rozsáhlejší, mají větší zkušenosti s kódováním a řeší méně známý problém nebo úkol. Posledním typem projektu je tzv. Embedded projekt. U něj je vyžadován nejvyšší stupeň znalosti všech použitých technologií, velmi široký vývojový tým a řešení komplexních a složitých úkolů. Existuje více druhů modelů COCOMO, každý je jinak složitý a bere v úvahu různé aspekty. (Geekforgeeks2)

4. Vlastní práce

Tato kapitola se věnuje jednotlivým krokům při tvorbě aplikace. Na začátku je popsána analýza požadavků na veškeré funkcionality, které bylo třeba v aplikaci realizovat. Dále se zde dozvíte o návrhu a implementaci jednotlivých funkcí. Nakonec bude okomentován výběr mezi jazyky PHP a Python pro tvorbu aplikace a srovnání vybraného jazyka s hotovými komerčními řešeními jako WordPress a OJS. Na konci je také proveden odhad doby strávené na práci pomocí modelu COCOMO při odhadované délce 1500 řádků kódu a za použití metody funkčních bodů odhad celkového možného počtu řádků kódu, které bude třeba použít. Front end aplikace byl zčásti již hotový a byl předán vedoucím práce s cílem doplnit aplikaci o konkrétní požadované funkcionality specifikované v cíli práce, zadání a provést potřebné úpravy. Stránky, které odkazují na webovou aplikaci, jsou dostupné z adresy <http://tesar.slovjani.info/>.

4.1 Analýza požadavků

Důležitou částí tvorby nové aplikace nebo jejích dílčích částí je analýza požadavků a funkcí, které by aplikace měla umět zprostředkovat. Je nezbytná pro určení všech podkladů, které jsou poté nutné k následné realizaci. Z takové úvodní analýzy pramení následný návrh a implementace. Hlavní myšlenkou analýzy je rozdělení větších úkolů (můžeme taky říci problému) na dílčí úkoly.

4.1.1 Požadavky na funkce aplikace

Tato část se bude věnovat požadavkům na funkčnosti webové aplikace. Přehled byl zpracován autorem do následující tabulky.

Tabulka 1 – Požadavky na jednotlivé funkcionality aplikace

Název požadavku	Popis požadavku
Stránka registrace	Obsahuje formulář, který slouží pro registraci nového uživatele do aplikace. Vložené údaje do tohoto formuláře jsou odeslány do databáze, kde se uloží.
Registrace uživatele	Uživatel si může vytvořit účet k přihlašování do aplikace.
Stránka přihlášení	Zobrazuje formulář, pomocí kterého se již zaregistrovaní uživatelé mohou přihlásit do aplikace.

Přihlášení uživatele	Zaregistrovaný uživatel se může přihlásit do aplikace pomocí formuláře. Po zadání údajů se ověří v databázi, že uživatel zadal správné údaje.
Stránka vkládání návrhů článků	Dostupný formulář pro přidání návrhu článku do databáze. Stránka je dostupná pouze pro uživatele, kteří mají schválenou registraci administrátorem.
Vložení článku	V případě přihlášení uživatele s potřebnými právy bude možno odesílat návrhy článků do databáze přes formulář.
Stránka administrace	Bude obsahovat výpis registrací z databáze do tabulky a formulář pro úpravu práv a registrací v databázi. Dostupná pouze administrátorovi. Upravovat nebo mazat registrace se bude podle uživatelského ID.
Schválení registrace	Administrátor může registraci ostatních uživatelů schválit.
Dočasné odebrání registrace	Administrátor může registraci ostatních uživatelů označit jako dočasně odebranou.
Smazání registrace	Administrátor může registraci ostatních uživatelů nenávratně smazat.
Přidělení práva stahování hotových verzí časopisu	Administrátor může přidělit právo uživateli stahovat hotové verze časopisu v PDF. Přihlášenému uživateli s tímto právem se v záložkách zobrazí odkaz pro stažení.

4.1.2 Uživatelské role

Přístup k jednotlivým sekcím v aplikaci bude řízen na základě práv. Z tohoto důvodu byly vytvořeny uživatelské role, které tyto práva zohledňují. Celý přehled je zobrazen v následující autorem vytvořené tabulce.

Tabulka 2 – Uživatelské role figurující v aplikaci

Uživatelská role	Popis role
Administrátor	Disponuje přístupem do všech sekcí, které aplikace obsahuje. V administrátorské sekci může přidělovat práva možnosti stahování hotových verzí časopisu ostatním uživatelům a spravovat jejich registrace. Ty může schválit, dočasně odebrat nebo úplně smazat z databáze. Může stahovat veškerá hotová vydání časopisu v PDF v záložkách: O časopisu, Archiv a ve výsledcích hledání v archivu.
Nepřihlášený uživatel	Je pouhým návštěvníkem webu. Má přístup do sekcí k registraci, přihlášení, informacích o časopisu, k seznamu redaktorů, instrukcí autorům článků a čtenářům a do archivu, kde může využít vyhledávání v hotových verzích časopisu. Nemůže stahovat hotové verze časopisu.

Přihlášený uživatel se schválenou registrací	Má přístup do všech sekcí stránky, kromě sekce administrační. V sekci vkládání článků může odeslat návrh článku do budoucího čísla časopisu do databáze.
Přihlášený uživatel s neschválenou nebo odebranou registrací	Má přístup do všech sekcí stránky, kromě sekce administrační a sekce vkládání návrhů článků. Z důvodu stavu jeho registrace musí vyčkat na schválení registrace administrátorem, aby mu bylo vkládání návrhů článků umožněno.
Přihlášený uživatel s právem stahování hotových vydání	Má přístup do všech sekcí stránky na základě jeho stavu schválení registrace. Může stahovat veškerá hotová vydání časopisu v PDF v záložkách: O časopisu, Archiv a ve výsledcích hledání v archivu.
Přihlášený uživatel bez práva stahování hotových vydání	Má přístup do všech sekcí stránky na základě jeho stavu schválení registrace. Nemůže stahovat veškerá hotová vydání časopisu v PDF v záložkách: O časopisu, Archiv a ve výsledcích hledání v archivu.

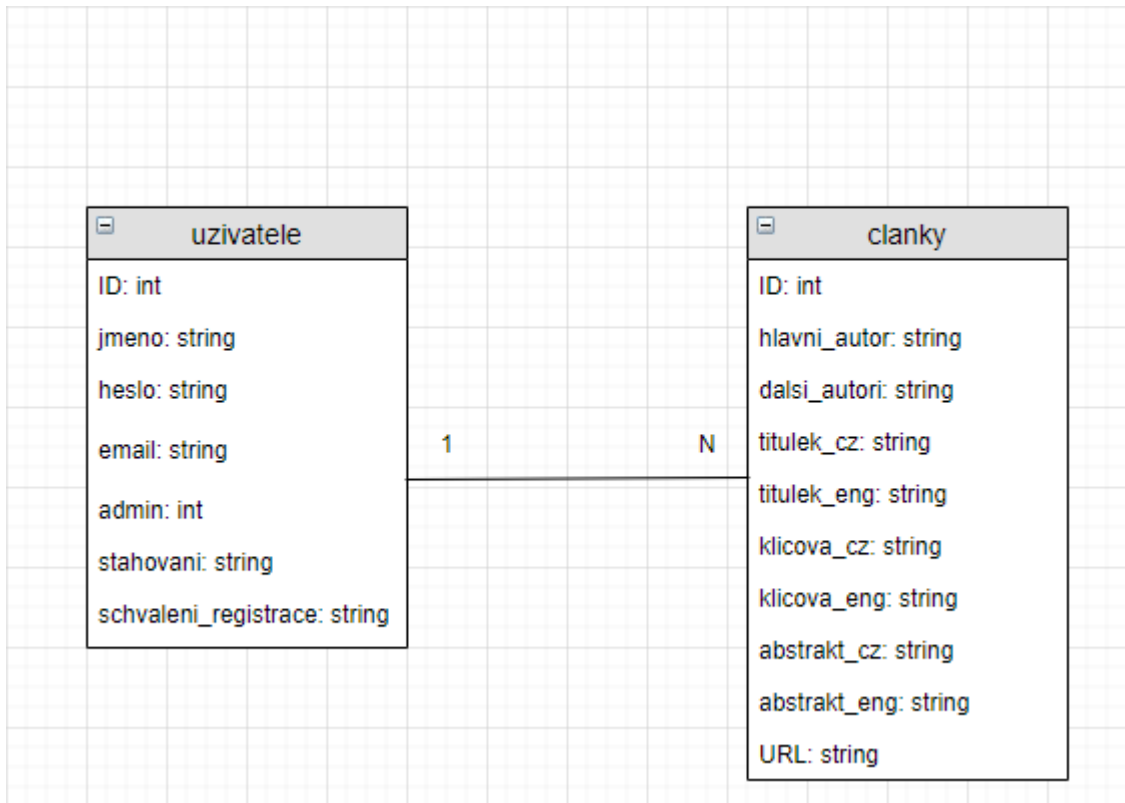
4.2 Návrh funkcionalit aplikace

Po vytvořené analýze požadavků na jednotlivé funkcionality aplikace a stanovení uživatelských rolí se přešlo k samotnému návrhu jednotlivých sekcí, které bylo nutné ve webové aplikaci realizovat. Konkrétně se jednalo o sekce registrace, přihlášení, administrace, vkládání článků a dostupnost stahování hotových verzí časopisu pouze pro lidi s určitým právem. V této kapitole bude probírán návrh databázového systému, Use Case diagram (diagram případů užití), seznam případů užití a jednotlivé wireframy pro veškeré nově navrhované sekce, které bylo třeba v aplikaci zrealizovat.

4.2.1 Databázový model

Databáze jsem navrhl dle požadavků na nové funkčnosti aplikace. Výsledný návrh je zde zohledněn v ER diagramu (Entity Relationship diagram). Nacházejí se v něm dvě třídy, které jsou v databázi reprezentovány tabulkami. Třída „uzivatele“ reprezentuje uživatele, kteří se do systému zaregistrují. O každém uživateli budou v databázi uloženy následující informace: ID uživatele, přihlašovací jméno, heslo pro přihlášení (nebude ukládáno v zadané podobě), e-mail uživatele, právo administrátora, právo stahování hotových verzí časopisu a stav schválení uživateli registrace. Dále zde máme třídu „clanky“, které popisuje návrhy na články, které uživatelé s příslušnými právy odeslali do databáze. O všech člancích bude v databázi uloženo ID článku, jméno hlavního autora článku, jména dalších autorů článku, titulek v češtině, titulek v angličtině, klíčová slova v češtině, klíčová slova v angličtině, abstrakt v češtině, abstrakt

v angličtině a URL odkaz na celý článek. Diagram byl vytvořen v online nástroji draw.io. Je dostupný na adrese <https://www.draw.io/>.

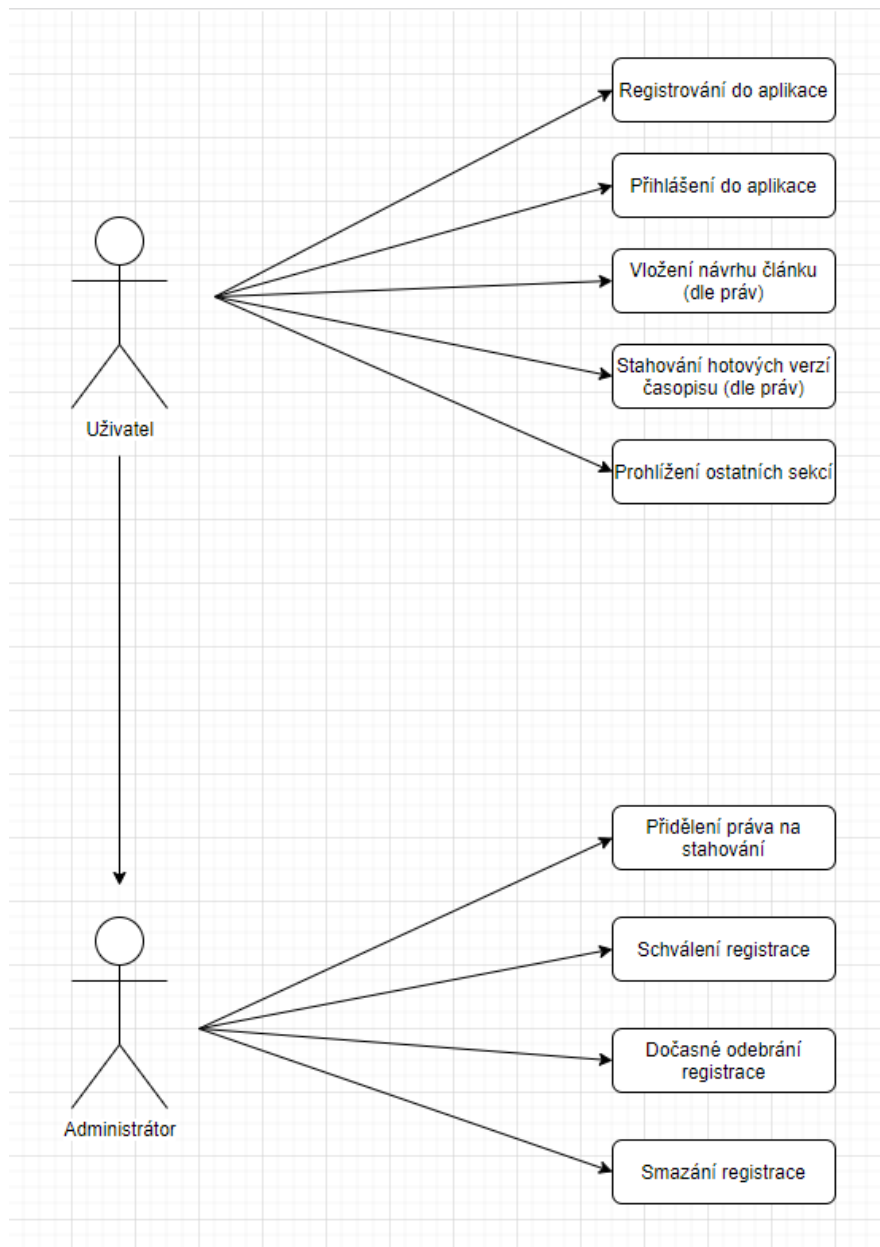


Obrázek 17 – Datový model systému (ER diagram)

Zdroj: Autor

4.2.2 Diagram případů užití

Na základě výsledků z úvodní analýzy byl vytvořen diagram případů užití neboli Use Case diagram. Tento diagram zobrazuje aktéry označené symbolem panáčka, kteří s aplikací přijdou do styku a můžou s ní nějakým způsobem manipulovat nebo v ní provádět určité akce. Každá taková operace je definována jako Use Case. Ty jsou v diagramu znázorněny obdélníky s názvem. Šipkami jsou spojené jednotlivé případy užití s aktérem, kterému náleží. Diagram byl vytvořen v online nástroji draw.io. Je dostupný na adrese <https://www.draw.io/>.



Obrázek 18 – Diagram případů užití

Zdroj: Autor

4.2.3 Seznam případů užití

V následující tabulce jsou detailněji popsány jednotlivé případy užití. Každý Use Case má přidělené své číslo označení, název, popis a aktéry.

Tabulka 3 – Seznam případů užití

Označení	Název	Popis	Akteři
1.	Registrace	Umožňuje zaregistrovat nového uživatele aplikace.	Všichni uživatelé
2.	Přihlášení	Zaregistrovaný uživatel se může přihlásit.	Registrovaní uživatelé
3.	Vložení návrhu článku	Uživatel se schválenou registrací, může vložit návrh článků do databáze.	Přihlášení uživatelé se schválenou registrací
4.	Stahování hotových verzí časopisu	Uživatel s právem stahovat může prostřednictvím odkazu zobrazit a poté stáhnout hotovou verzi časopisu.	Přihlášení uživatelé s právem stahovat
5.	Prohlížení ostatních sekcí nevyžadujících žádná práva	Umožňuje prohlížení sekcí, které nevyžadují žádná uživatelská práva ani nutnost být přihlášen	Všichni uživatelé
6.	Přidělení práva na stahování	Dovoluje adminovi přidělovat práva pro stahování	Administrátor
7.	Schválení registrace	Umožňuje adminovi schvalovat registrace uživatelů	Administrátor
8.	Dočasné odebrání registrace	Admin má možnost označit registraci jako dočasně odebranou	Administrátor
9.	Úplné smazání registrace	Admin může úplně smazat registraci uživatele	Administrátor

4.2.4 Drátové modely (wireframy)

Další částí návrhu jednotlivých sekcí tvoří jednotlivé drátové modely realizovaných částí aplikace. Pro drátové modely je nejčastěji používána anglická zkratka wireframe. Ten určuje umístění jednotlivých funkčních prvků na navrhované stránce. Náčrtek pomocí drátového modelu neobsahuje žádné grafické specifikace jako barvy nebo konkrétní obrázky. Konstrukce návrhu je koncipována pouze prostřednictvím čar, textu a dalších negrafických prvků. Veškeré wireframy v této práci byly vytvořeny autorem pomocí online nástroje Mockingbird. Ten je dostupný na internetové adrese <https://gomockingbird.com/home>. Nejprve byla navrhována sekce registrace uživatele, poté přihlášení uživatele, následovala sekce pro administrátora, a nakonec sekce pro vkládání návrhů článků.

4.2.4.1 Sekce registrace uživatele

Představa, jak by mohla vypadat registrace nových uživatelů webové aplikace, byla vcelku jednoduchá. Do lišty záložek se přidá nová záložka s názvem. Poté bude obsahovat registrační formulář s několika poli, která budou nutná vyplnit k úspěšnému dokončení registrace. Formulář bude nějakým způsobem ohraničen a bude také obsahovat tlačítko, které potřebnou operaci po stisknutí provede.

The wireframe shows a web page layout for user registration. At the top is a header bar labeled "Hlavička" containing a row of navigation buttons: "Admin", "Register", "Login", "Insert a.", "About j.", "Editors", "Review", "Authors", "Readers", and "Archive". Below the header is a user status bar: "Welcome, you are login as 'user'". Click here to [Logout]. The main content area is titled "Registrace uživatele" and "Zde se můžete zaregistrovat". It features a registration form with four input fields: "Přihlašovací jméno", "E-mail", "Heslo", and "Heslo znovu", followed by a "Zaregistrovat" button. The footer is labeled "Zápatí".

Obrázek 19 – Wireframe registrace uživatele

Zdroj: Autor

4.2.4.2 Sekce přihlášení uživatele

Další položkou návrhu byla realizována funkcionality pro přihlášení registrovaných uživatelů do aplikace. Principiálně bude designově podobná jako registrační sekce, také má svou vlastní záložku v navigační liště. Její součástí je také formulář sloužící tentokrát k přihlášení zaregistrovaných uživatelů. Ten disponuje pouze dvěma textovými poli pro vyplnění přihlašovacího jména a hesla. Tlačítko zde bude ověřovat správnost údajů a v případě shody uživatele přihlásí.

Hlavička

[Admin](#)
[Register](#)
[Login](#)
[Insert a.](#)
[About j.](#)
[Editors](#)
[Review](#)
[Authors](#)
[Readers](#)
[Archive](#)

Welcome, you are login as "user". Click here to [Logout](#)

Zde se můžete přihlásit

Přihlašovací jméno
 Heslo

Zápatí

Obrázek 20 – Wireframe přihlášení uživatele

Zdroj: Autor

4.2.4.3 Sekce pro administrátora

Sekce pro administrátora je už celkem odlišná od ostatních funkcionalit. Obsahuje sice také formulář, ale tím podobnost s přihlášením a registrací uživatele končí. Myšlenkou návrhu bylo vytvořit přehledné prostředí pro administrátora, kde bude moci upravovat práva uživatelů a měnit stav jejich registrací v databázi. Samozřejmě bude realizace jako nová záložka v navigaci. Dále bude obsahovat výpis uživatelských registrací z databáze včetně jejich práv. Forma výpisu dat z databáze z hlediska přehlednosti by mohla být vcelku přehledná v podobě tabulky. Admin by si mohl pohodlně dohledat uživatele, u kterých chce provést změnu a na základě zobrazených údajů je realizovat. Nutností bude také formulář, který bude několik polí. Jedno bude muset nějakým způsobem identifikovat konkrétního uživatele, u kterého budou provedeny změny a další pole přiřadí jednotlivé údaje, které se budou měnit. Obsahem budou také dvě tlačítka. První z nich slouží k úpravě dat záznamů o registraci. Druhé tlačítko realizuje úplné odstranění záznamu o registraci uživatele z databáze.

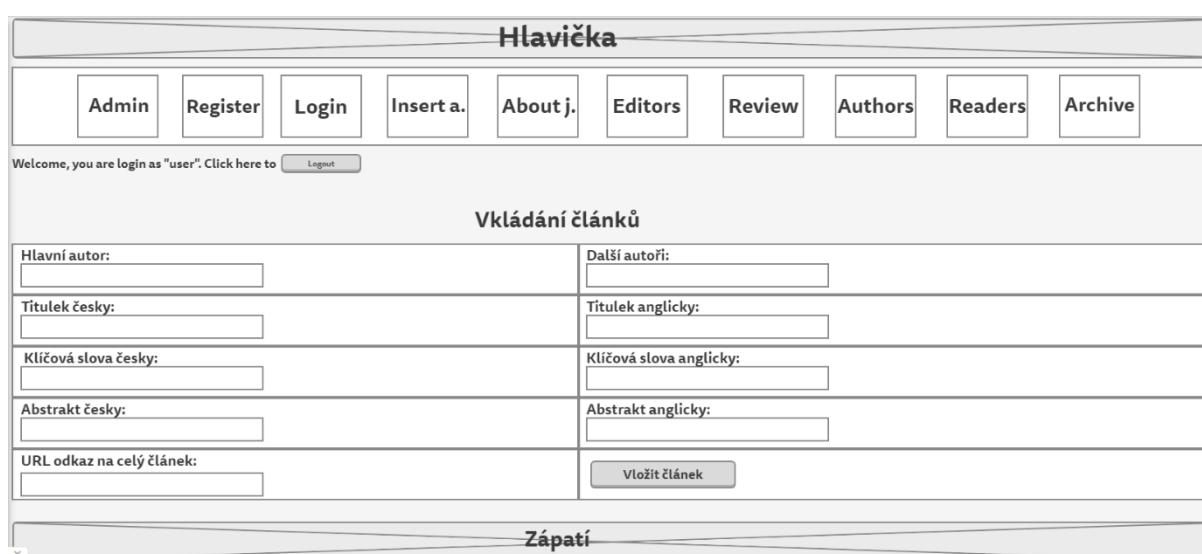


Obrázek 21 – Wireframe administrace

Zdroj: Autor

4.2.4.5 Sekce vkládání návrhů článků

Nakonec bylo třeba realizovat návrh sekce pro vkládání návrhů článků do databáze. Opět bude součástí navigačního menu na vrchu stránky jako samostatná záložka. Dále zde bude nutný formulář, který bude obsahovat jednotlivé pole pro specifikaci a jednoznačnému určení článku. Samozřejmostí je zde také tlačítko, které všechna data zadaná uživatelem do formuláře vyplní, zpracuje a odešle do databáze.

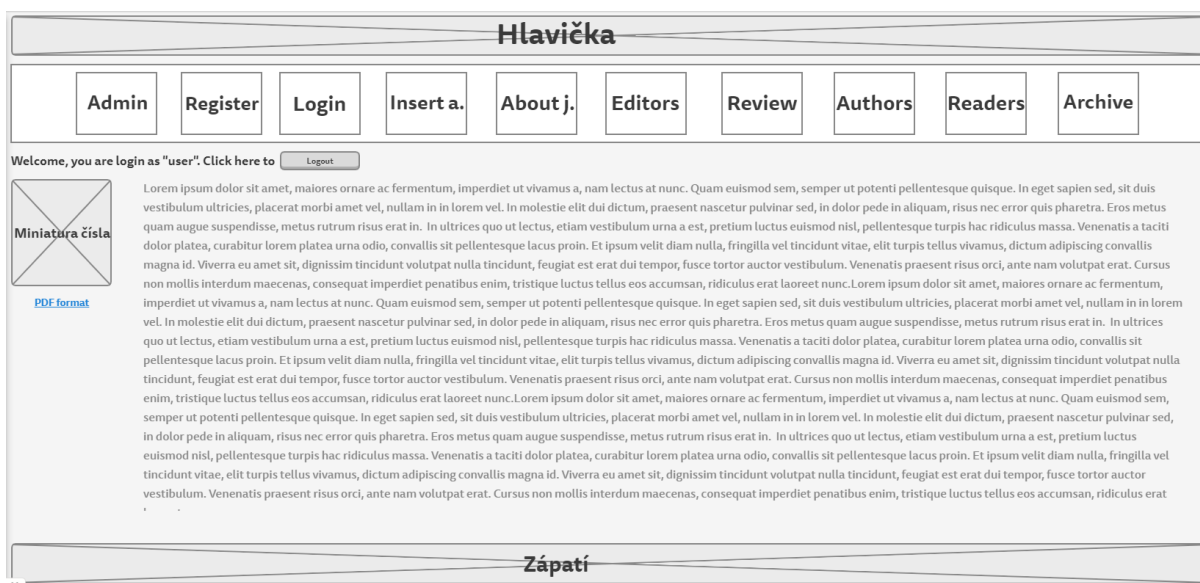


Obrázek 22 – Wireframe vkládání návrhů článků

Zdroj: Autor

4.2.4.6 Odkazy pro stahování hotových verzí časopisu

Poslední funkcionalitou jsou dostupné odkazy pro stažení hotových verzí časopisu pro uživatele s právem k jejich používání. To se dotýká tří aplikačních sekcí. První z nich je sekce o časopisu, kde je zobrazeno nejnovější vydání časopisu. V sekci archivu jsou dostupná veškerá hotová čísla časopisu. Odkaz bude dostupný pro každé hotové vydání časopisu. Nakonec je v archivu dostupná možnost vyhledání hotových čísel časopisu prostřednictvím formuláře pomocí zadaných parametrů. Po zobrazení výsledků bude odkaz dostupný u každého nalezeného výsledku.



Obrázek 23 – Wireframe odkazu pro stahování

Zdroj: Autor

4.2.5 Použitý hardware, software a technologie pro vývoj

Vývoj byl realizován v domácím počítačovém prostředí na stolním počítači s operačním systémem Windows 10. Detailnější specifikace stroje je následující: grafická karta – Radeon XFX RS Radeon RX 570 8GB TripleX edition, procesor – Intel Core i5 9400F (6 jader 2,9GHz, až 4.1 GHz v turbo režimu), operační paměť 16GB RAM.

Software použit k vývoji byl NetBeans ve verzi 8.2. Dále byl využit multiplatformní freeware FTP klient FileZilla v opensource licenci ve verzi 3.47.2. Webová aplikace byla testována ve webovém prohlížeči Google Chrome.

Technologie, které byly využity v práci, jsou následující: skriptovací jazyk PHP v nejnovější verzi, HTML 5, dále kaskádových stylů CSS a jako relační databázový systém bylo použito MySQL.

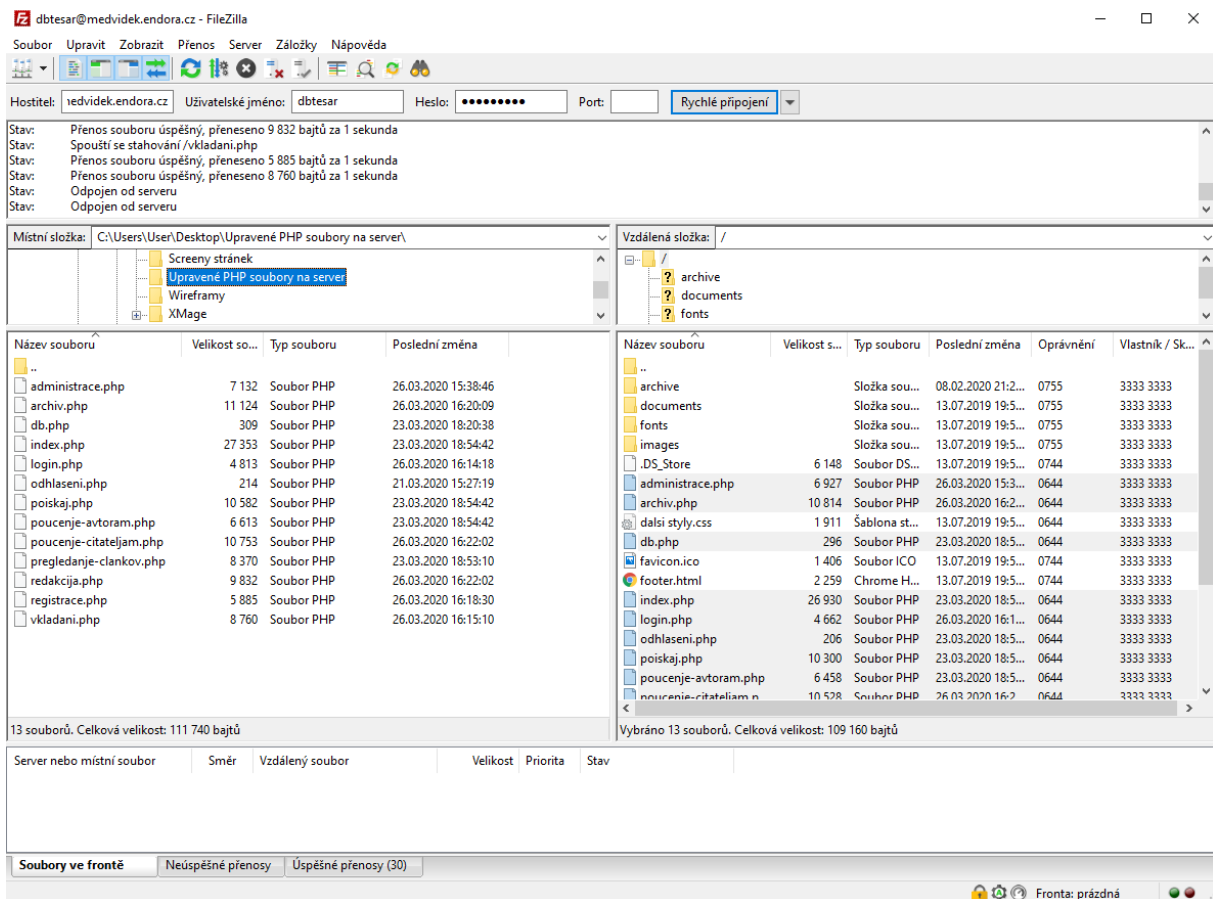
4.3 Implementace

Po vytvoření návrhu jednotlivých funkcionalit, které bylo třeba v aplikaci realizovat, můžeme přejít na implementaci. U každé vytvořené sekce se můžete dočíst komentář ohledně implementace a řešení jednotlivých problémů. Pokusím se také zařadit relevantní ukázky určitých zajímavějších částí zdrojového kódu. Na začátku bude zmíněn způsob nahrávání stránek na vzdálený server.

4.3.1 Nahrání stránek na internet

Na úvod chci zmínit způsob nahrávání webových stránek (aplikací) na internet. Když už je webová aplikace kompletně hotova nebo při testování jednotlivých funkcí, ji nahrajeme na internet. K tomu je zapotřebí mít pronajatou doménu stránek a webový server, kam můžeme stránky nahrát. V této práci je využit hosting zdarma na serveru www.endora.cz, který tudíž nutnost registrace domény a pronajmutí serveru odpadá. Pro méně rozsáhlé projekty je způsob zajištění potřebných věcí naprosto dostačující.

Server nabízí možnost nahrání souborů přímo na svých stránkách ručně, ale dle mého názoru je mnohem lepší použít nějaký program pro nahrávání souborů do vzdáleného úložiště. Je tomu tak z důvodu, že na stránkách je nutné vkládat soubory po jednom a to může dlouho trvat. Já jsem se rozhodl použít program FileZilla, což je FTP klient zdarma nabízející potřebné možnosti. Dále by šel použít například Total Commander, který dost často defaultně nainstalovaný na počítačích nebo noteboocích. Funguje na principu, že na jedné straně má uživatel svůj adresář se soubory a na druhé straně po přihlášení vidí adresář serveru, na který stránky nahrává. Na obrázku můžete vidět ukázku toho, jak program FileZilla vypadá.



Obrázek 24 – Ukázka prostředí programu FileZilla

Zdroj: Autor

4.3.2 Sekce registrace uživatele

Nejprve bylo třeba zrealizovat sekci, ve které se noví uživatelé aplikace budou moci do systému zaregistrovat. Návrh byl vytvořen na základě formuláře, do kterého zadá člověk požadované údaje. Dle těchto shromážděných dat je nutné, aby se registrační údaje zadané uživatelem ve formuláři zapsaly do databáze. Přihlašovací jméno a e-mail můžeme ukládat do databáze přímo v podobě, která byla zadána. Heslo bychom však z bezpečnostního hlediska do databáze měli ukládat nějakým způsobem zašifrované. Kdyby se náhodou stalo, že se do databáze někdo neoprávněný dostane, tak nezíská přístupové údaje uživatelů a nemůže je zneužít. Budeme tedy potřebovat naše stránky připojit v tomto okamžiku k databázi. Připojení k databázi je realizováno na následujícím obrázku. Byl vytvořen soubor s názvem db.php, který se zavolá pokaždé, když je připojení do databáze potřeba. Kód realizuje reporting chyb při připojení, definuje databázový server, jméno databáze, uživatelské jméno a heslo pro přihlášení do databáze a poté se připojí.

```

<?php
error_reporting(E_ALL);
define('DBHOST', 'localhost');
define('DBNAME', 'dbtesar');
define('DBUSER', 'dbtesar');
define('DBPASS', 'dbtesarXl');
mysql_connect(DBHOST, DBUSER, DBPASS) or die(mysql_error());
mysql_select_db(DBNAME) or die(mysql_error());
session_start();
?>

```

Obrázek 25 – Kód realizující připojení do databáze

Zdroj: Autor

Formulář obsahuje čtyři pole (Přihlašovací jméno, E-mail, Heslo a Heslo znovu). Na konci formuláře je dostupné tlačítko Registrovat, které celou operaci provede. Když uživatel nevyplní některá pole nebo zadá dvě neshodná hesla, objeví se varovná hláška nad formulářem. Bezpečnost ukládání hesla je zajištěna tím, že se vezme heslo zadané uživatelem a před uložením do databáze se převede hashovací funkcí SHA512. Znamená to, že do databáze bude uloženo heslo zašifrované touto funkcí o určité délce. Dále se ještě před uložením do databáze zkontroluje, že přihlašovací jméno se neshoduje s jiným již uloženým v databázi. V kladném případě vyskočí upozornění, že zvolené jméno již existuje a je potřeba zvolit jiné. Když je vyplněn formulář a pole s chybovými zprávami je prázdné, tak se registrační údaje, které uživatel zadal, uloží do databáze. Heslo je zašifrované, ostatní údaje se uloží přesně v té podobě, jakou zadal uživatel. Dále se ještě uloží do databáze defaultně nastavená práva uživatele. Lze to vyčíst z příkazu insert na konci kódu na obrázku. Vloží se ID uživatele, jméno, zašifrované heslo, e-mail, právo admina (nastavená nula), právo na stahování (nastaveno ne), a schválení registrace (nastaveno ne). Při zadávání údajů ve formuláři je ještě ošetřeno, že uživatel nezadal na začátek nebo konec řetězce nežádoucí mezery a podobně pomocí trim. Dále jsou speciální znaky v řetězcích převedeny na HTML entity díky použití htmlspecialchars. Pole e-mail je také validováno v HTML díky použití typu pole jako e-mail a musí obsahovat zavináč. Na obrázku přikládám zdrojový kód, který je proveden po kliknutí na tlačítko registrovat ve formuláři. Chybové hlášky jsou vypisovány nad formulářem červeně a ve formě odrážek. Pokud registrace proběhla úspěšně je uživatel přesměrován na sekci registrace a dozví se, že registrace proběhla úspěšně. Toto je realizováno pomocí metody GET po úspěšném ukončení registračního procesu.

```

if(isset($_POST['register']))
{
    $message = array();

    $_jmeno = trim(htmlspecialchars($_POST['jmeno']));
    $_email = trim(htmlspecialchars($_POST['email']));
    $_heslo = trim(htmlspecialchars($_POST['heslo']));
    $_ukladaneheslo = hash(sha512, $_heslo);

    if(empty($_jmeno))
    {
        $message[] = 'Pole jméno je prázdné.';
    }

    if(empty($_email))
    {
        $message[] = 'Pole e-mail je prázdné.';
    }

    if(empty($_heslo))
    {
        $message[] = 'Pole heslo je prázdné.';
    }

    else
    {
        if($_POST['heslo'] != $_POST['heslo2'])
        {
            $message[] = 'Hesla se neshodují.';
        }
    }

    $select = mysql_query("SELECT jmeno FROM uzivatele WHERE jmeno='".mysql_real_escape_string($_jmeno)."'");
    if(mysql_num_rows($select) > 0)
    {
        $message[] = 'Zvolte jiné uživatelské jméno, zadané již existuje.';
    }

    if(empty($message))
    {
        $_heslo = mysql_real_escape_string($_heslo);
        $_jmeno = mysql_real_escape_string($_jmeno);
        $_email = mysql_real_escape_string($_email);
        $_ukladaneheslo = hash(sha512, $_heslo);
        mysql_query("INSERT INTO uzivatele VALUES('".$_jmeno."','".$_ukladaneheslo."','".$_email "','0','ne','ne')");
        header('Location: ../registrace.php?ok=1');
    }
}

```

Obrázek 26 – Kód prováděný po kliknutí na tlačítko registrace

Zdroj: Autor

Realizace jednotlivých chybových hlášek je vidět na následujícím obrázku.

- Pole jméno je prázdné.
- Pole e-mail je prázdné.
- Pole heslo je prázdné.

Registrace uživatele

<input type="text"/>	Přihlašovací jméno
<input type="text"/>	E-mail
<input type="text"/>	Heslo
<input type="text"/>	Heslo znovu
<input type="button" value="Registrovat"/>	

Obrázek 27 – Ukázka chybových hlášek formuláře

Zdroj: Autor

Zde můžete vidět strukturu tabulky, do které jsou informace o uživateli do databáze ukládány. Hesla jsou zašifrovaná proti případnému odcizení.

+ Nastavení		id	jmeno	heslo	email	admin	stahovani	schvaleni_registrace
<input type="checkbox"/>				1	admin	66ff5c67288a3bedecbc1e22c413d2d6252b3996af2e4ef489...	vmerunka@gmail.com	1 ano ano
<input type="checkbox"/>				2	jaroslav.spacek	c927a689e122d80ad13a6860ce2a1dc23b2e38094e2d5950fe...	jardaspacak@centrum.cz	0 ne odebrana
<input type="checkbox"/>				3	honzik.konvalina	4a1387a8218907c8e58714969956d5094cd3c5a0d9f849e7bf...	jankonvalina@seznam.cz	0 ano ano
<input type="checkbox"/>				4	PetrCukr235	b213f3ab6a8a03ecc4c883250f0d7ac4e027a597947bdc6ef...	Petr.cukrar@gmail.com	0 ne ne
<input type="checkbox"/>				5	petruska208	6d48b1be436a605153a4843faf79571a67aaf1ae6c3c50c2b3...	petrakorinkova@centrum.cz	0 ne ano
<input type="checkbox"/>				6	jandi6637	ce6d1c92efb10927a0a58ec3286a90cb9ab7563ffde426cfe0...	jjirijanda@gmail.com	0 ano odebrana
<input type="checkbox"/>				7	pepikNovak	3424b6e64bea73f11020a834445cae2bcc670ddc75024fc0ce...	josef.novak@gmail.com	0 ano ne
<input type="checkbox"/>				8	nedved25	fe3531be62e7b3adb87a416030f0cff8ddaec2b3e796e6a90...	tomasnedved@seznam.cz	0 ne ne

↑ Zaškrtnout vše Zaškrtnuté:

Obrázek 28 – Struktura tabulky uživatelé

Zdroj: Autor

Kompletní vzhled sekce je prezentován na následujícím obrázku.

Obrázek 29 – Kompletní sekce registrace

Zdroj: Autor

4.3.3 Sekce přihlášení uživatele

Po úspěšné implementaci registrace bylo třeba zrealizovat, aby se registrovaný uživatel mohl do aplikace přihlašovat. To zahrnuje kontrolu zadaných údajů a jejich porovnání s registračními údaji uloženými v databázi. V případě úspěchu se uživatel úspěšně přihlásí, v opačném případě bude upozorněn o nesprávnosti zadaných údajů. Pro ověření budeme zde opět potřebovat připojení k databázi, které je realizované odkázáním na soubor s uloženým kódem. Registrační formulář obsahuje pouze dvě pole (Přihlašovací jméno, Heslo). Poté co uživatel zadá jméno a

heslo bude nutné zadané heslo převést funkcí SHA512, která byla použita při ukládání do databáze, aby se řetězce porovnali. Pokud otisk hesla v databázi a otisk zadaného bude stejný, přihlášení bude úspěšné. Přesměrujeme také uživatele na úvodní stránku aplikace (index.php). Když je přihlášení úspěšné, tak vytvoříme proměnné session (takzvaného sezení uživatele). S těmito proměnnými je v některých částech aplikace operováno především z hlediska práv. V opačném případě bylo zadáno špatné heslo. Formulář v případě nevyplnění polí nebo zadaného špatného jména či hesla pošle hlášku chybné přihlašovací údaje. Na následujícím obrázku je kód, kterým je realizován celý proces přihlášení uživatele.

```

if(isset($_POST['login']))
{
    $_jmeno = mysql_real_escape_string(trim($_POST['jmeno']));
    $_heslo = mysql_real_escape_string(trim($_POST['heslo']));
    $_ukladaneheslo = hash(sha512, $_heslo);
    $select = mysql_query("SELECT * FROM uzivatele WHERE jmeno='$_jmeno' AND heslo='$_ukladaneheslo'" or die(mysql_error());
    $udaje = mysql_fetch_assoc($select);
    if(mysql_num_rows($select) == 1)
    {
        session_regenerate_id();
        $_SESSION['jmeno'] = $_jmeno;
        $_SESSION['heslo'] = $_heslo;
        $_SESSION['admin'] = $udaje['admin'];
        $_SESSION['stahovani'] = $udaje['stahovani'];
        $_SESSION['schvaleni_registrace'] = $udaje['schvaleni_registrace'];
        header("Location: index.php");
    }
    else
    {
        $error_message = 'Chybné přihlašovací údaje';
    }
}

```

Obrázek 30 – Kód provedený po kliknutí na tlačítko přihlásit

Zdroj: Autor

Po úspěšném přihlášení do aplikace by také nebylo špatné někde vidět napsané, jaký uživatel je přihlášen a mít možnost se z aplikace správně odhlásit. Výpis informací o přihlášeném uživateli s možností odhlásit se vidíte realizovanou na následujícím obrázku s kódem. Další obrázek ukazuje finální vzhled sekce a vypadá těchto pár řádků kódu v praxi. Tyto informace o uživateli budou uvedené v každé sekci v případě přihlášeného uživatele.

```

<?php
if($_SESSION["jmeno"]) {
?>
    Welcome, you are login as <strong><?php echo $_SESSION["jmeno"]; ?></strong> Click here to <a href="odhlaseni.php"><button type="button">Logout</button></a>.
<?php
}else{}
?>

```

Obrázek 31 – Kód zobrazující informace o přihlášeném uživateli

Zdroj: Autor



Welcome, you are login as [honzik.konvalna](#). Click here to [Logout](#)

Zde se můžete přihlásit

Přihlášení uživatele

<input type="text"/>	Jméno
<input type="text"/>	Heslo
<input type="button" value="Přihlásit"/>	



Obrázek 32 – Finální vzhled sekce přihlášení a informace o přihlášeném

Zdroj: Autor

Jak si jistě každý všiml, informace obsahují tlačítko realizující odhlášení uživatele. Po úspěšném odhlášení je uživatel přesměrován zpět na obrazovku pro přihlášení. Kliknutí na tlačítko provede kód umožňující odhlášení uživatele, který je zobrazen na obrázku níže.

```
<?php
session_start();
unset($_SESSION["id"]);
unset($_SESSION["jmeno"]);
unset($_SESSION["admin"]);
unset($_SESSION["stahovani"]);
unset($_SESSION["schvaleni_registrace"]);
header("Location:login.php");
?>
```

Obrázek 33 – Kód obsluhující tlačítko pro odhlášení

Zdroj: Autor

4.3.4 Sekce administrace

Po úspěšném vytvoření funkcionality registrace nových uživatelů a systému přihlášení se můžeme vrhnout na administrátorskou sekci. V té by dle návrhu měl být administrátor schopný měnit práva uživatelů na stahování hotových verzí časopisu a upravovat stav registrací uživatelů, které jsou uloženy v databázi. Stránka by tedy měla nejprve obsahovat výpis všech

registrací do nějakého přehledného formátu. Jako jednoduché a relativně přehledné řešení se nabízí použití tabulky. Na obrázku vidíte kód, které realizuje výpis dat z databáze do tabulky.

```
<?php
require_once('db.php');
$result=mysql_query('SELECT id,jmeno,email,admin,stahovani,schvaleni_registrace FROM uzivatele ORDER BY id;');

echo "<table border=1>";
if ($row=mysql_fetch_assoc($result))
{
echo "<tr>";
foreach($row as $key => $value)
{
echo "<th>".$key;
}
echo "<tr>";
foreach($row as $key => $value)
{
echo "<td>".$value;
}
}

while ($row=mysql_fetch_assoc($result))
{
echo "<tr>";
foreach($row as $key => $value) {echo "<td>".$value; }
}
echo "</table>";
echo "<br />"
?>
```

Obrázek 34 – Kód pro realizaci výpisu dat z databáze do tabulky

Zdroj: Autor

Pod tabulkou by se měl nacházet formulář, pomocí něhož bude možno zadat do databáze změny práv a stavu registrace, případně řádek s registrací úplně smazat. Formulář by měl obsahovat dvě různá tlačítka. Jedno pro změnu údajů v databázi a druhé pro úplné smazání záznamu v případě odstranění registrace uživatele. Registrace zde bude mít tři stavy. Schválená registrace je v databázi vedena s hodnotou schvaleni_registrace „ano“, neschválená registrace jako „ne“ a dočasně odebraná registrace jako „odebrana“. Při schválené registraci může přihlášený uživatel vkládat návrhy článků, při neschválené nebo odebrané registraci články vkládat prozatím nemůže. Informace o této skutečnosti bude uživateli zobrazena pod informacemi o přihlášeném uživateli v každé záložce, do které má možnost tento uživatel přistupovat. Jedná se o všechny sekce aplikace kromě administrace a vkládání článků. Na obrázku níže je vidět informace zobrazovaná dotčeným uživatelům s neschválenou nebo dočasně odebranou registrací.



Welcome, you are login as **Jaroslav.spacek**. Click here to [Logout](#)

Vaše registrace je vedena jako dočasně odebraná nebo neschválená. Vyčkejte, až jí admin schválí. Z tohoto důvodu nemůžete zatím vkládat články.

Obrázek 35 – Informace o stavu registrace

Zdroj: Autor

Formulář obsahuje tři pole. Dle pole ID uživatele se identifikuje konkrétní záznam v databázi, který má být upraven nebo smazán. Další dvě pole slouží pro úpravu údajů v databázi (stahování časopisu a schválení registrace). V případě použití tlačítka uložit změny (update údajů v databázi) je nutné vyplnit všechna tři pole. Předpokládejme, že administrátor ví, co má do polí vyplnit. V případě stahování časopisu zadává pouze hodnoty „ano“ a „ne“ a v případě změny stavu registrace zadává hodnoty „ano, „ne“ a „odebrana“. Když chce smazat kompletně určitou registraci v databázi, použije tlačítko smaž řádek. V tomto případě stačí vyplnit pouze pole ID uživatele, dle kterého bude záznam smazán. Další dvě pole není při mazání nutné vyplňovat.

Následující obrázky znázorňují kód, který se provede po stisku tlačítka uložit změny a smaž řádek.

```

if(isset($_POST['edit']))
{
    $message = array();
    $id = trim(htmlspecialchars($_POST['id']));
    $stahovani = trim(htmlspecialchars($_POST['stahovani']));
    $schvaleni_registrace = trim(htmlspecialchars($_POST['schvaleni_registrace']));

    if(empty($id))
    {
        $message[] = 'Vyplňte pole ID uživatele, u kterého mají být provedeny změny.';
    }

    if(empty($stahovani))
    {
        $message[] = 'Vyplňte pole stahování časopisu.';
    }

    if(empty($schvaleni_registrace))
    {
        $message[] = 'Vyplňte pole schválení registrace.';
    }

    if(empty($message))
    {
        $id = mysql_real_escape_string($id);
        $stahovani = mysql_real_escape_string($stahovani);
        $schvaleni_registrace = mysql_real_escape_string($schvaleni_registrace);
        mysql_query("UPDATE uzivatele SET stahovani = '{$stahovani}', schvaleni_registrace = '{$schvaleni_registrace}' WHERE id = '{$id}'") or die("Nelze upravit údaje v databázi: " . mysql_error());
        header("Location: ./administrace.php?ok=1");
    }
}

```

Obrázek 36 – Provedený kód po stisknutí tlačítka ulož změny

Zdroj: Autor

```

if(isset($_POST['delete']))
{
    $message = array();

    $id = trim(htmlspecialchars($_POST['id']));

    if(empty($id))
    {
        $message[] = 'Vyplňte pole ID uživatele, u kterého mají být provedeny změny.';
    }

    if(empty($message))
    {
        $id = mysql_real_escape_string($id);
        mysql_query("DELETE FROM uzivatele WHERE id = '{$id}'") or die("Nelze upravit údaje v databázi: " . mysql_error());
        header('Location: ./administrace.php?ok=1');
    }
}

```

Obrázek 37 – Kód provedený po stisku tlačítka smaž řádek

Zdroj: Autor

Po úspěšném vyplnění změn práv nebo smazání záznamu je admin přeměrován zpět na stránku administrace a zobrazí se mu potvrzení změn. Navíc znovu uvidí aktualizovanou databázi a může si okamžitě zkontrolovat, jestli všechny operace proběhly správně. Funkcionalita je dostupná pouze uživateli, který má v databázi nastavenou hodnotu admin na 1. Ostatním sekce není přístupná. Na obrázku níže jsou zobrazeny chybové hlášky a kompletní vzhled finální verze sekce.

SLOVJANI.info
časopis pisaný v medzúslovjanskom jazyku

ISSN 2570-2116
p-ISSN 2570-2108

administrace | registrace | přihlášení | vkládání článků | o časopisu | redakce | přegledanje článkov | poučenie avtoram | poučenie čitateľjam | archiv

Welcome, you are login as admin. Click here to [Logout](#).

Seznam uživatelských registrací

- Vyplňte pole ID uživatele, u kterého mají být provedeny změny.
- Vyplňte pole stahování časopisu.
- Vyplňte pole schválení registrace.

id	jmeno	email	admin	stahovani	schvaleni_registrace
1	admin	vmerunka@gmail.com	1	ano	ano
2	jaroslav.spacek	jaraspacak@centrum.cz	0	ne	odebrana
3	honzik.konvalina	jankonvalina@seznam.cz	0	ano	ano
4	PetrCukr235	Petr.cukrar@gmail.com	0	ne	ne
5	petruska208	petrakorinkova@centrum.cz	0	ne	ano
6	jandic637	jirijanda@gmail.com	0	ano	odebrana
7	pepikNovak	josef.novak@gmail.com	0	ano	ne
8	hedved25	tomasnedved@seznam.cz	0	ne	ne

Změna dat v databázi

ID uživatele pro provedení změn

Stahování časopisu

Schválení registrace uživatele

Obrázek 38 – Finální vzhled sekce administrace s chybovými hláškami

Zdroj: Autor

4.3.5 Sekce vkládání návrhů článků

Poslední samostatnou sekcí, kterou bylo třeba realizovat bylo vkládání návrhů článků do databáze. Byl navrhnout formulář s větším počtem polí, pomocí něhož se zadaná data budou dát uložit do databáze. Vzhledem k většímu počtu polí než u formulářů realizovaných v předchozích sekcích, bylo třeba vymyslet jejich přehlednější zobrazení. Kdyby byla všechna pole zobrazena pod sebou, formulář by se roztáhl do celkem slušné délky a bylo by nutné při jeho vyplňování značně scrollovat po stránce. Byl proto použit takzvaný grid countainer, který rozdělí itemy formuláře do dvou a stane se tak přehlednější. Návrh obsahuje devět polí formuláře, a nakonec tlačítko pro odeslání vyplněných hodnot do databáze. Realizace ukládání dat do databáze je obdobná, jako při registraci nového uživatele. Při nevyplnění některého z polí se zobrazí chybová hláška, že pole není vyplněno. V tomto případě už ale nepotřebujeme nic šifrovat, jelikož uživatel nezadá žádné citlivé údaje. Pole URL implementuje validaci dostupnou v URL pomocí jeho nastavení na type="URL". Dále bylo potřeba, aby byla sekce dostupná pouze pro uživatele, kteří mají registraci schválenou administrátorem. Nepřihlášeným uživatelům také nebude sekce dostupná. Na obrázcích dostupných níže vidíte realizaci podmínky pro přístupnost sekce, chybové hlášky a kompletní vzhled sekce.

```
<?php if (!$_SESSION["jmeno"] || $_SESSION["schvaleni_registrace"] == "odebrana" || $_SESSION["schvaleni_registrace"] == "ne")
{
}
else
{
?>
<li><a href="vkladani.php" style="background-color: #02447E; color: white;">
<strong>vkládání článků</strong><br>
<span id="condensed">insertion of articles</span></a></li>
<?php } ?>
```

Obrázek 39 – Podmínka viditelnosti sekce

Zdroj: Autor

Vkládání návrhů článků

- Pole hlavní autor je prázdné.
- Pole další autoři je prázdné.
- Pole titulek česky je prázdné.
- Pole titulek anglicky je prázdné.
- Pole klíčová slova česky je prázdné.
- Pole klíčová slova anglicky je prázdné.
- Pole abstrakt česky je prázdné.
- Pole abstrakt anglicky je prázdné.
- Pole URL odkaz na celý článek je prázdné.

Hlavní autor: <input type="text"/>	Další autoři: <input type="text"/>
Titulek česky: <input type="text"/>	Titulek anglicky: <input type="text"/>
Klíčová slova česky: <input type="text"/>	Klíčová slova anglicky: <input type="text"/>
Abstrakt česky: <input type="text"/>	Abstrakt anglicky: <input type="text"/>
URL odkaz na celý článek: <input type="text"/>	<input type="button" value="Vložit článek"/>

Obrázek 40 – Chybové hlášky

Zdroj: Autor

The screenshot shows the top part of the SLOVJANI.info website. The logo 'SLOVJANI.info' is in red and blue. Below it, the text 'časopis pisaný v međzuslovjanskem jeziku' is visible. A navigation bar contains links: registrace, přihlášení, vkladání článků, o časopisu, redakcija, přegledanje člankov, poučjenje avtoram, poučjenje čitateľjam, and archiv. A user login message says 'Welcome, you are login as honzik.konvalina. Click here to Logout'.

Vkládání návrhů článků

Hlavní autor: <input type="text"/>	Další autoři: <input type="text"/>
Titulek česky: <input type="text"/>	Titulek anglicky: <input type="text"/>
Klíčová slova česky: <input type="text"/>	Klíčová slova anglicky: <input type="text"/>
Abstrakt česky: <input type="text"/>	Abstrakt anglicky: <input type="text"/>
URL odkaz na celý článek: <input type="text"/>	<input type="button" value="Vložit článek"/>

Obrázek 41 – Kompletní vzhled sekce vkladání návrhů článků

Zdroj: Autor

4.3.6 Odkazy pro stahování hotových verzí časopisu

Nakonec bylo třeba zrealizovat zobrazování odkazů pro stahování hotových verzí časopisu pro uživatele, kteří na to mají právo. To se dotýká tří aplikačních sekcí. První z nich je sekce o časopisu, kde je zobrazeno nejnovější vydání časopisu. V sekci archivu jsou dostupná veškerá

hotová čísla časopisu. Odkaz bude dostupný pro každé hotové vydání časopisu. Nakonec je v archivu dostupná možnost vyhledání hotových čísel časopisu prostřednictvím formuláře pomocí zadaných parametrů. Po zobrazení výsledků bude odkaz dostupný u každého nalezeného výsledku. Odkaz byl vymyšlen tak, že u každého hotového čísla bude zobrazen obrázek titulní stránky časopisu v rámečku a pod ním direktivní odkaz. Po kliknutí na obrázek nebo na odkaz se uživateli zobrazí dokument ve formátu PDF, který lze stáhnout. Na přiloženém obrázku je vidět kód realizace odkazu a poté grafická podoba.

```

if ($_SESSION["stahovani"] == 'ano')
{
echo "<table><col><col><tr>\n";
echo "<td class='col-issue'>";
echo "<a href='archive/" . $filename . ".pdf'>";
echo "<img class='img-issue' src='archive/" . $filename . ".jpg'></a><br>";
echo "<a href='archive/" . $filename . ".pdf'>PDF format</a>";
echo "</td>\n<td>";
}
else {}

```

Obrázek 42 – Kód realizace odkazu

Zdroj: Autor

2017 (2)



Merunka, V., Steenbergen van, J. **Slovjanska kulturna diplomacija, SWOT analiza, strategija, i taktika do budućnosti, aktivnost Slovjanskoj unije v Čehiji** (in English: Slavic Cultural Diplomacy - SWOT analysis, strategy and tactics to the future, Slavic Union activities) in: *Slovjani.info*, 2017, 2(2), pages 63-76. ISSN 2570-7108 (print), ISSN 2570-7116 (on-line). Available from: <http://slovjani.info>.
Članok govori o teoriji i praksi Slovjanskoj kulturnoj diplomacije u država Slovjanska Unija v Čehiji. Prva část teksta govori o potrebnosti kulturnoj diplomacije. Vтора část jest SWOT analiza i formulacija strategije s pomoću modela AIDAS kako metodu upravljanja kulturnih projekti. Treća část govori o programu društva Slovjanska unija v Čehiji.
ključne slova: Slovjanska kulturna diplomacija; SWOT analiza; AIDAS model; Slovjanska unija v Čehiji; konferencija o međuslovjanskom jeziku; međunarodny festival dni slovjanskoj kulturny

This paper describes methods and tools of the Slavic cultural diplomacy, which aim to improve the current status of Slavic nations in the context of today's globalized world. The first part is explaining the need for Slavic cultural diplomacy followed by an all-Slavic analysis of internal and external factors that cause the present dangerous situation. The following text is an overview of theoretical research and practical international projects that could be used. The main part of this paper is a detailed SWOT analysis and a proposal for a new Slavic strategy for the future. This paper describes the AIDAS method of managing individual projects and, in the formulation of general strategy, uses the J.W. Berry's acculturation hypothesis about four variants of intercultural and international relations. This paper reflects the author's practical experience in the recent all-Slavic movement and supports the idea of better peaceable Slavic involvement in the processes of building the modern globalized world. There is an appendix to the program objectives of the Slavic Union at the end of this paper.

keywords: cultural diplomacy, SWOT analysis, AIDAS model, Slavic Union in the Czech Republic, Conference on InterSlavic Language, days of Slavic culture

2017 (1)



Merunka, V. **Prva međunarodna lingvistična konferencija o međuslovjanskom jeziku CISLa 2017** in: *Slovenská unie*, 2017, 2(1), pages 22-23. ISSN 2464-756X (print), ISSN 2464-7578 (on-line). Available from: <http://slovjani.info>.
Conference on InterSlavic Language 2017. <http://cisla.slavic-union.org/> jest prvo međunarodno sašadanje ljudi, koji uže mnogo let sorabotajut na Internetu. Smyslom konferencije jest štriti jedi drugog, bezposredny obmēn pogledov, vzajemna inspiracija, i glubočija sorabota do dalečiešego razvijā našego obžego interesa, ktorimi sut slovjanske skonstruovane zonalne jezky, vključajuč možnosť jhogo širenja do praxi.
ključne slova: CISLa 2017, međuslovjansky jezyk, konferencija

2016 (2)



Kulhavý, A., Merunka, V. **Slovanská vzájemnost** in: *Slovenská unie*, 2016, 1(2), pages 1-5. ISSN 2464-756X (print), ISSN 2464-7578 (on-line). Available from: <http://slovjani.info>.
Referát předneseny dne 02. 03. 2016 v Senátu Parlamentu ČR na konferenci "T. G. Masaryk a 95. výročí tzv. Ruské pomoci akce"
ključne slova: slovjanska vzajemnost

Obrázek 43 – Odkazy u každého hotového čísla časopisu

Zdroj: Autor

4.4 Výsledný výběr mezi PHP a Pythonem

Nyní bych rád zhodnotil, proč jsem se rozhodl v mé práci využít jazyka PHP a Python jsem nezvolil. PHP jakožto skriptovací jazyk má velkou tradici v programování obecně. Vznikl v roce 1996 a prošel četnými změnami. Pracuje na principu architektury klient-server a jeho syntaxe se v mnoha směrech podobá programovacím jazykům C nebo C++. Je vcelku jednoduchý z tohoto důvodu na naučení a mnohé konstrukty se takzvanému céčku velmi podobají. Můžeme nalézt i formulace, které se úplně shodují. Vzhledem k velké tradici jazyka se dá v případě problémů při vývoji spousta informací a typů vyhledat na internetu. Velice příjemná je zde také relativně dobrá možnost kód PHP propojovat s HTML kódem a naopak. Tato skutečnost je při tvorbě webové aplikace nebo jejích částí vskutku žádoucí a programátorovi ulehčí v mnoha směrech práci. Dalším pozitivem využitelným při práci je také jednoduché a intuitivní propojení PHP s databázemi. Ať už se jedná o MySQL, Oracle, PostgreSQL nebo MS SQL, spojení je jednoduše realizovatelné a přístup k databázi není složitý. Navíc je PHP multiplatformní a téměř každý hostingový server ho plně podporuje. Na druhou stranu je jazyk o něco slabší z hlediska výkonnosti a rychlosti zpracování a není příliš vhodné ho používat pro aplikace obrovského rozsahu. To se ale aplikace realizované v této práci netýká, extrémně rozsáhlá není. V některých aspektech není až tak dokonalý v bezpečnosti, ale to lze za použití potřebných zabezpečovacích prvků celkem obstojně vyřešit. Jazyk PHP jsem si také vybral pro vývoj, protože s ním mám na rozdíl od Pythonu již nějaké zkušenosti.

Jazyk Python má stejně jako PHP dlouhodobou tradici. První verze byla vydána dokonce již v roce 1991 v Nizozemsku. Jazyk byl vytvořen s myšlenkou čitelnosti zdrojového kódu pro lidské oko. Dle mého názoru to ale má spíše efekt opačný. Python totiž na rozdíl od ostatních programovacích jazyků nemá jasně určené začátky a konce bloků kódu. Syntaxe používá jako oddělení částí kódu odsazení od kraje souboru a v zápise je používáno také daleko méně závorek, které jednotlivé podmínky a proměnné přehledně oddělují. Syntaxe se tolik nepodobá notoricky známým programovacím jazykům a pro nezasvěceného uživatele se může snadno stát jazyk nepřehledným. Python má k dispozici rozsáhlé knihovny, které nabízejí široké spektrum hotových úloh nebo konstruktů, které se dají využít. Velice dobře se také integruje s ostatními programovacími jazyky. Lze tedy například C, C++ nebo Javu kombinovat s Pythonem. Jazyk má podobně jako PHP problémy s pomalejším načítáním a spouštěním aplikace z důvodu, že je jazyk interpretovaný. Kompilované jazyky jako například C, C++ budou mít vždy o něco vyšší výkonnost než interpretované. Hlavním důvodem, proč nebyl

Python vybrán pro vývoj aplikace v této práci, je také složitější připojení k databázím a přístup k datům než v PHP. Na základě návrhu bylo jasné, že do databáze bude nutné přistupovat relativně často. Vrstva v Pythonu pro komunikaci s databázemi je oproti PHP mírně zaostalá a nerealizuje některé operace, jak by bylo potřeba. Python je především využíván ke tvorbě desktopových aplikací. Není příliš vhodný pro mobilní aplikace a programování her, jelikož je celkem slušný žrout paměti. Obecně se nedoporučuje programovat v Pythonu, když budeme mít nějak výrazněji omezené paměťové zdroje. Nevýhodou je zde oproti PHP i složitější hledání chyb. Python také nebyl vybrán pro tvorbu, protože s ním nemám žádné zkušenosti.

4.5 Komparace PHP s WordPressem

Jak bylo popsáno výše, pro tvorbu aplikace byl vybrán skriptovací jazyk PHP. Rád bych nejprve začal se srovnáním s redakčním systémem WordPress. Jedná se o nejpoužívanější a nejpoužívanější systém pro tvorbu webových stránek a aplikací na světě. Na jeho bázi mají své weby založené velké společnosti jako například Microsoft, dále také americký vládní White house nebo slavná hudební kapela The Rolling Stones. Za jeho pomocí můžou lidé vytvořit jakýkoliv typ webových stránek, aniž by napsali jediný řádek kódu. Podporuje vkládání jednotlivých funkčních prvků na stránky, které jsou předem vytvořeny. Zdrojový kód je tak schovaný a tvůrce stránek ho vůbec nemusí znát. WordPress má výhodu dostupnosti zcela zdarma v licenci opensource.

Ve srovnání s PHP je v redakčním systému tvorba o něco pohodlnější. Pouze na stránku umístíte předdefinované prvky a nemusíte se starat o zdrojový kód. Na druhou stranu PHP nabízí při znalosti kódu možnosti si upravit veškeré prvky na stránce, jak autor chce a není žádným způsobem omezen. Ve WordPressu sice také lze upravovat zdrojový kód, ale vzhledem k jeho automatické deklaraci to může být složité a zdlouhavé. Oproti PHP nabízí redakční systém velké množství pluginů a rozšíření, které práci vývojářům usnadňují. To je také jedním z důvodů, proč je WordPress tak populární. Slabinou redakčního systému je také nižší bezpečnost. Jelikož každá stránka funguje v redakčním systému naprosto totožně, je nutno aplikovat navíc speciální bezpečnostní opatření a pravidelně zálohovat. Navíc také weby běžící na WordPressu jsou pomalejší než weby realizované prostřednictvím programovacích jazyků. Toto je obecný problém většiny redakčních systémů a WordPress není výjimkou. Na rozdíl od PHP jsou weby fungující na bázi WordPressu celkem náročné na výkon serveru, na kterém běží. Tohle v případě PHP nemusíme řešit, jelikož PHP není na serverové požadavky příliš náročné.

4.6 Komparace PHP s OJS

Nakonec bych rád porovnal vybraný programovací jazyk PHP s redakčním systémem pro správu a publikování elektronických časopisů Open Journal Systems. Jedná se o opensource systém, který svým uživatelům dává k dispozici možnost automatického nastavení opakovaných redakčních činností, vytvoření efektivní webové prezentace časopisu včetně online publikování plných textů článků s mnoha dalšími funkcemi. Je také velice populárním ve světě, protože ho využívá okolo deseti tisíc časopisů. Nabízí svým klientům responzivní čtenářský front end s možností výběru jednoho ze tří vzhladů. Systém také disponuje možností přihlašování uživatelů, zadávání příspěvků, jejich editace a také recenzní řízení k článkům. Redakční systém také podporuje archivaci od vložení článku, zhodnocení článku až po jeho publikování.

Výhodou oproti PHP je nesporně spousta hotové práce, kterou by musel programátor vytvořit na svém vlastním webu časopisu sám. Pokud však nevyhovují uživateli přednastavené vzhlady a implementace jednotlivých funkcí, je jazyk PHP ideální cestou k tomu si vše vytvořit podle vlastních představ. Příkladem požadavků na odborný časopis může být originální vzhlad, požadavky jiné editace nebo vypisování článků nebo speciální funkce nedostupné v Open Journal Systems. Nevýhodou OJS je to, že obsahuje občasné chyby a bugy, které jsou odstraňovány pouze komunitou uživatelů. Tento postup nemusí vždy dokonale fungovat, jelikož uživatelé nemají nikdy stejně dobré znalosti dané problematiky. OJS sice vyniká jednoduchou administrací a integrovaným systémem přihlášení, ale implementované funkce nemusí všem požadavkům na funkčnost vyhovovat. V PHP na tom programátor sice stráví více času, ale může si navrhnout a implementovat cokoliv. V Open Journal Systems mohou některé požadované věci prostě chybět.

4.7 Odhad potřebného času na projekt pomocí modelu COCOMO

V této kapitole bych rád provedl odhad, kolik zaberou odhadem rozšiřující práce, které bylo třeba v aplikaci provést. Pro odhad byl použit základní model COCOMO, který bere v úvahu pouze odhadovaný počet řádků kódu pro konstrukci požadovaných funkcionalit a konstanty, jejichž hodnoty závisí na typu projektu. Projekt byl zde vyhodnocen jako organický, jelikož vývojový tým obsahuje málo lidí, konkrétně pouze jednoho člověka. Řeší se v něm také problémy, které jsou známé a nevyžadují speciální znalosti co se týče programovacích jazyků a přidružených technologií. Pro výpočet bude použit vzorec ve tvaru: $E = a(KLOC)^b$.

Konstanty a,b zde symbolizují koeficienty používané při organickém projektu. Pro organický projekt je hodnota konstanty $a=2,4$ a $b=1,05$. KLOC zde znázorňuje odhadované tisíce řádků kódu (number kilo lines of code), které bude třeba použít. Osobní odhad jsem zde určil 1500 řádků kódu. Jako KLOC bude tedy použito číslo 1,5. (Geekforgeeks2)

Následující odstavec bude věnován výpočtu odhadu času, který nutný věnovat k dokončení projektu. Výsledné číslo nám vyjde v takzvaných člověkoměsících. Jedna taková jednotka čítá 140 hodin. Pro srovnání s reálnou vynaloženou dobou a tohoto odhadu výsledek poté převedeme na člověkodny. Jeden takový den čítá osm hodin.

Výpočet: $E = a(KLOC)^b$; $E = 2,4 * (1,5)^{1,05}$; $E = 3,67$ člověkoměsíců

Převod: $3,67$ člověkoměsíců * $140 = 514$ hodin; 514 hodin / $8 = 64$ člověkodní (MD)

4.8 Odhad použitých řádků kódu na projekt pomocí metody funkčních bodů

Zde bych rád zrealizoval odhad délky kódu potřebných funkcionalit aplikace v důsledku použití při srovnání s reálnou délkou kódu vytvořeného autorem. Do funkčních bodů je započítán počet uživatelských vstupů a transakcí, počet uživatelských výstupů a transakcí, celkový počet uživatelských dotazů, celkový počet souborů, ke kterým se přistupuje a počet obrazovek. Všechny tyto věci jsou zohledněny ve výpočtu počtu funkčních bodů. Poté co získáme výsledný počet funkčních bodů, ho vynásobíme průměrnou délkou kódu na jeden funkční bod v použitém jazyce. Tím získáme celkový odhad délky aplikace nebo jejích funkcionalit v řádcích kódu. Ke zjištění celkového odhadovaného počtu funkčních bodů bude použit online kalkulátor dostupný pod odkazem <http://groups.umd.umich.edu/cis/course.des/cis375/projects/fp99/table.html>. Kalkulátor zohledňuje dvě části výpočtu. Nejprve do něj zadáme jednotlivé počty uživatelských vstupů, výstupů, dotazů, a externích rozhraní, do kterých aplikace přistupuje. Všechny tyto parametry se klasifikují třemi úrovněmi složitosti (simple, average a complex). Ve druhém kroku vyplňujeme důležitost čtrnácti parametrů, které jsou nebo nejsou v projektu obsaženy nebo jsou nebo nejsou důležité. Mezi tyto parametry patří například implementované zálohování a obnova dat, komunikace mezi daty, nutnost vkládání dat online, jak kritická je pro aplikaci vysoká rychlost přístupu a tak dále. Všechny tyto kritéria ohodnotíme čísly od nuly do pěti, kde nula je malý důraz a pětka je vysoká důležitost. Poté nám kalkulátor spočte počet funkčních bodů projektu. Po zadání potřebných dat a vyplnění důležitosti jednotlivých

parametrů nám kalkulátor napočítal po zaokrouhlení 61 funkčních bodů, které budou potřeba v projektu.

Nyní máme k dispozici počet funkčních bodů projektu a stačí nám vynásobit počet funkčních bodů počtem řádků v kódu na jeden funkční kód. Použit byl programovací jazyk PHP a počet řádků kódu na jeden funkční bod nalezneme v tabulce, která je dostupná na adrese <https://www.namcook.com/how-srm-works/Prog-Lang-1-CAI.pdf>. PHP zde má přiřazenu hodnotu 53,3 řádků kódu na jeden funkční bod.

Výpočet odhadu: 61 funkčních bodů * 53,3 PHP řádku na 1 bod = 3251 řádků kódu

5. Výsledky a diskuze

Na základě této práce vznikla webová aplikace zabývající se rozhraním zahraničního odborného časopisu. V první fázi práce byla sestavena analýza požadavků, které má aplikace splňovat. Jednotlivé potřeby, které měly funkcionality umět, byly přehledně zpracovány do tabulky. Dále byly definovány uživatelské role, které budou v aplikaci figurovat a byly podrobněji popsány. Následoval návrh databázového modelu, který byl realizován ve formě ER diagramu. Pro definici jednotlivých případů užití byl vytvořen diagram případů užití společně se seznamem jednotlivých případů, který byly blíže popsány a specifikovány. V další fázi návrhu byly stanoveny drátové modely (neboli wireframy) některých sekcí aplikace a představa, jak budou vypadat. Dále byl v návrhu specifikován použitý hardware a software, který byl použit v domácím prostředí pro vývoj aplikace.

Poté se plynule přešlo na fázi implementace. Zde bylo nejprve popsáno, jakým způsobem byly stránky uvedeny do provozu a jaký software pro to byl použit. Konkrétně se jednalo o detailnější popis nahrání stránek na server, použití hosting a technologie. Nejprve byla vytvořena sekce registrace nového uživatele. Tato sekce umožňuje zaregistrovat uživatele a jeho registrační údaje uložit do databáze. Následně byl zrealizován systém přihlašování a odhlašování do aplikace. Opět vznikla samostatná sekce s formulářem pro přihlášení, pomocí kterého proběhne ověření správnosti zadaných údajů z databáze. Zrušení uživatelského sezení se realizuje tlačítkem logout, které je dostupné v každé sekci aplikace v případě, že je uživatel přihlášen. Dále byla vytvořena administrační sekce, ve které jsou dostupné hlavní informace o uživatelských registracích uložených v databázi. Díky formuláři může admin měnit práva uživatelů v databázi, konkrétně právo na stahování hotových verzí časopisu. Lze také z jeho strany změnit stav uživatelské registrace na schválenou, neschválenou nebo dočasně odebranou. Administrátor má možnost registraci z databáze úplně smazat pomocí tlačítka. Identifikátorem pro změnu údajů nebo mazání registrací je uživatelské ID. Poté byla realizována sekce vkládání informací o článcích. Proces probíhá vyplněním formuláře a tlačítkem uživatel odešle informace do databáze. Přístup k této funkcionalitě mají pouze přihlášení uživatelé se schválenou registrací od administrátora. Nakonec byly realizovány odkazy pro stahování hotové verze časopisu pro uživatele, kteří disponují právem odkazy používat.

Na konci vlastní práce a po dokončení aplikace je okomentován výběr použitého programovacího jazyka, konkrétně PHP. Je srovnán s druhým navrhovaným jazykem k tvorbě programu, a to Pythonem. Komparovány jsou zde veškeré relevantní pro a proti, které hrály roli

při výběru jazyka pro zpracování aplikace. Dále je vybraný programovací jazyk PHP postupně srovnán s komerčními hotovými řešeními WordPress a s redakčním systémem pro upravování a publikaci elektronických časopisů OJS. Opět jsou zhodnoceny klíčové aspekty rozdílu použití programovacího jazyka a již hotových řešení, které jsou dnes dostupná. V následující tabulce můžete vidět shrnutí klíčových kritérií porovnávání PHP a zmíněných redakčních systémů. Výsledná webová aplikace je dostupná na internetové adrese <http://tesar.slovjani.info/>.

Tabulka 4 – Klíčová kritéria při srovnání PHP a redakčních systémů

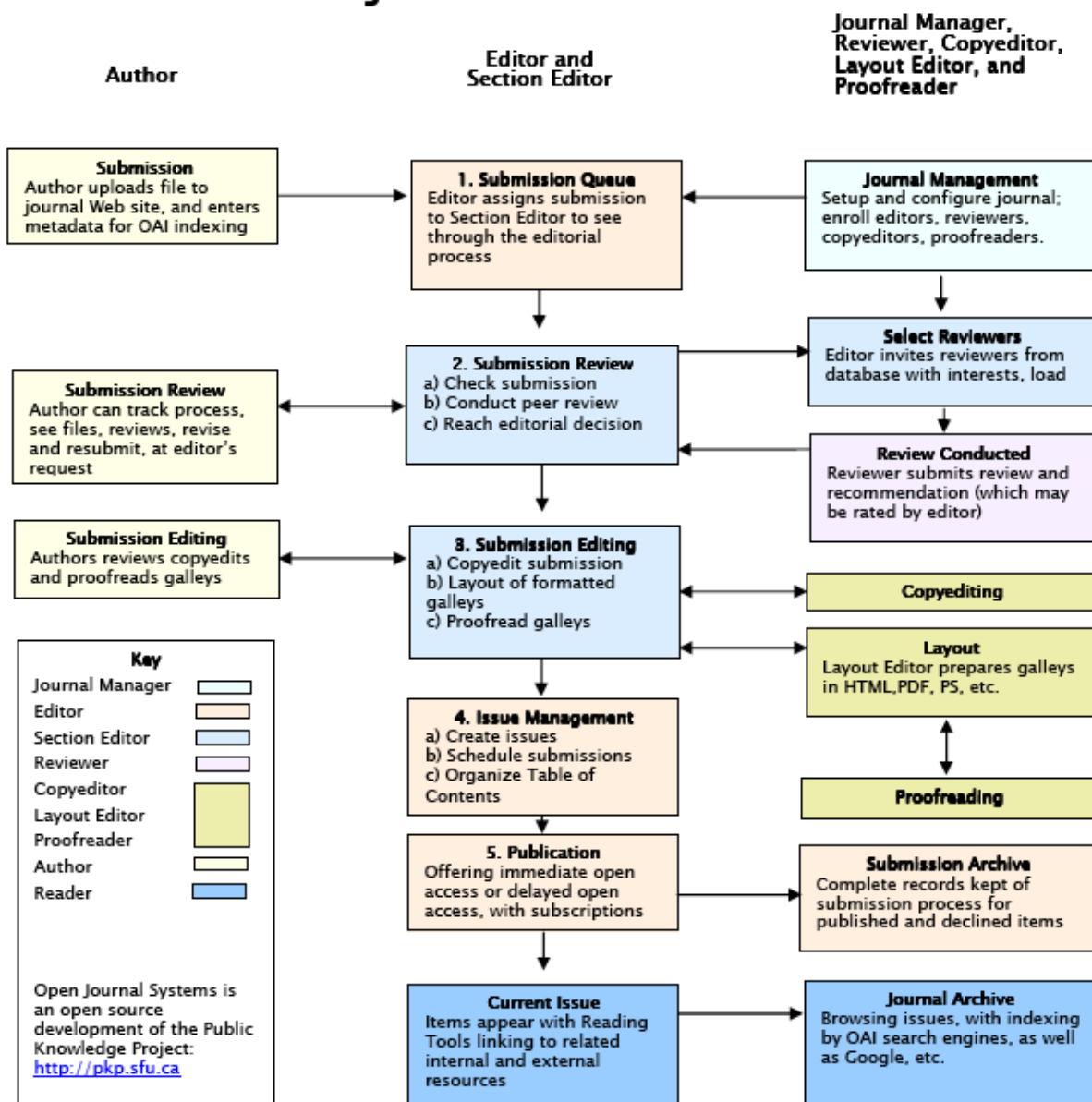
Hodnocené kritérium	PHP	WordPress	OJS
Nutnost znalosti jazyků pro tvorbu webu	Znalost nutná, jelikož si autor dělá všechno sám podle svých představ na rozdíl od WP a OJS, kde jsou prvky předdefinované.	Není nutná, ale je výhodou. WordPress kód sám generuje na základě použitých prvků a uživatel si ho může upravit.	Není nutná, systém používá již hotové prvky a funkcionality podobně jako WordPress.
Úprava zdrojového kódu	Dobrá dostupnost úprav kódu, jelikož si ho autor sám vytvořil. Ví, na jakém místě a co má přesně upravit. Kód uložený v jednotlivých souborech, které si autor sám pojmenoval.	Kód lze také upravovat, ale je daleko méně přehledný. Je automaticky vygenerovaný a nemusí být lehké se v něm vyznat.	Kód lze také upravovat, ale může být méně přehledný. Je tomu tak z důvodu, že si ho autor nevytvořil sám, ale upravuje existující.
Postup tvorby v systému	V případě tvorby v PHP začínáte sice z ničeho, ale můžete si vše navrhnout podle svých představ. Nejste omezení předdefinovanými konstrukty.	S využitím WordPressu má uživatel k dispozici spoustu předdefinovaných konstruktů a také široké spektrum pluginů, které může použít. Tvorba je pohodlnější než v PHP, ale v případě originálního vzhledu může něco chybět.	Při použití OJS nemusí uživatel vytvářet vše od začátku, pouze si navolí, co chce použít. Tvorba není tak pohodlná jako ve WP, ale není nutná širší znalost jako u PHP.
Rychlost fungování a reakce webu	Web vytvořený v PHP na požadavky uživatele reaguje pružně a rychle. Na rozdíl od redakčních systémů je v tomto ohledu spolehlivější.	Problémem všech redakčních systémů je pomalejší reakce a obecně fungování webu. Ve srovnání s PHP nebo jiným jazykem budou redakční systémy vždy pomalejší.	Problémem všech redakčních systémů je pomalejší reakce a obecně fungování webu. Ve srovnání s PHP nebo jiným jazykem budou redakční systémy vždy pomalejší.

Bezpečnost	Bezpečnost v PHP je na celkem slušné úrovni. Případné nedostatky se dají vyřešit způsoby jako například šifrování ukládaných dat nebo jinými způsoby. Slabinou je, že kódy PHP jsou dostupné na serveru. Správce serveru k nim má tedy přístup.	Slabinou WordPressu je také nižší bezpečnost. Jelikož každá stránka funguje v redakčním systému naprosto totožně, je nutno aplikovat navíc speciální bezpečnostní opatření a pravidelně zálohovat data.	OJS má z hlediska bezpečnosti problém v trošku jiném směru. Chyby a bugy jsou většinou odstraňovány komunitou uživatelů a z bezpečnostního hlediska tak nemusí být úprava úplně dokonalá.
------------	---	---	---

Dále bych se chtěl zmínit o double-blind review návrhů článků, které by bylo potřeba do aplikace také zavést. Můžeme to brát z hlediska diplomové práce jako možné rozšíření realizované webové aplikace, jelikož review článků již není součástí této práce. V dalším odstavci bude popsáno, jak by proces review návrhů mohl teoreticky vypadat.

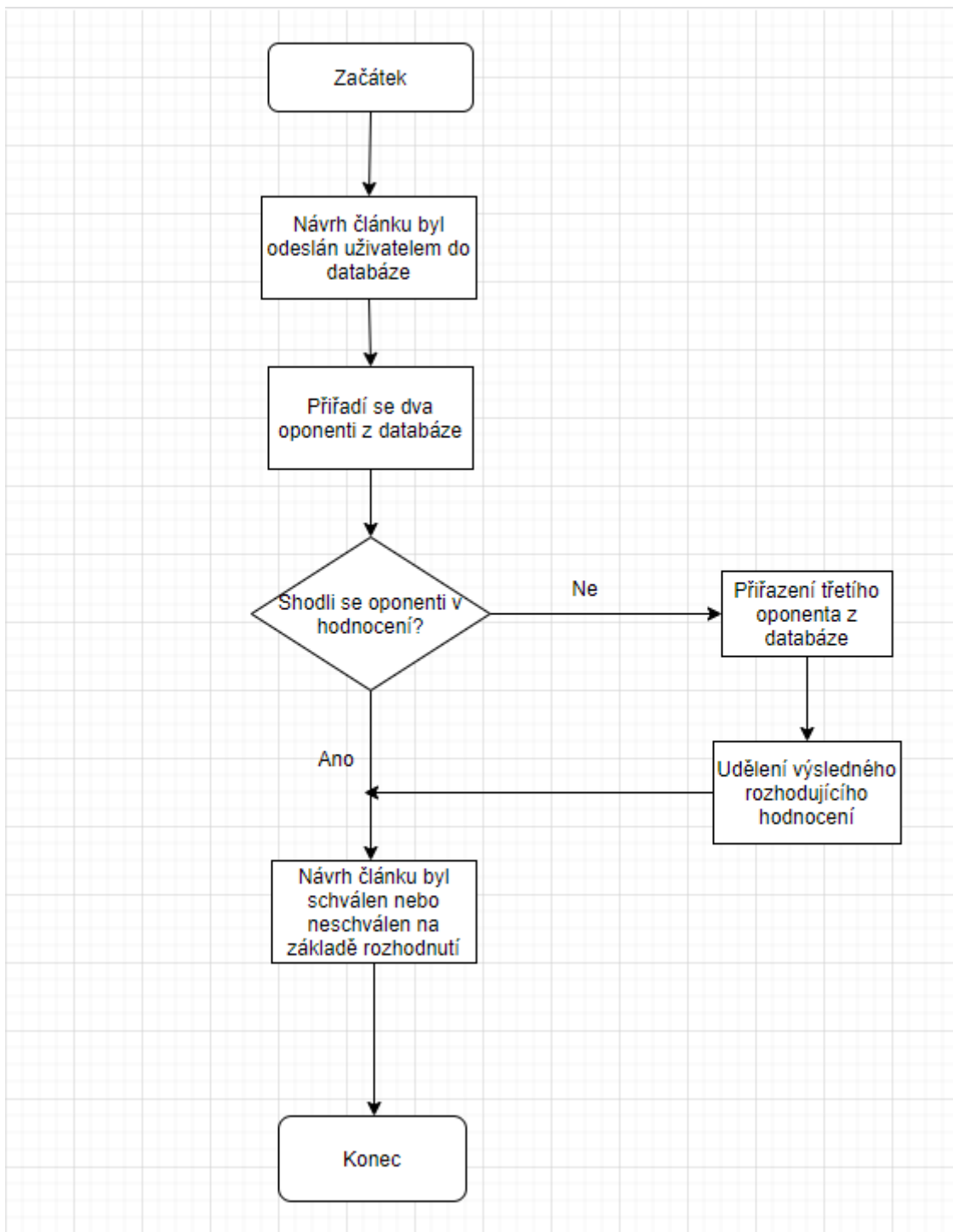
Návrhu článku, který odešle oprávněný uživatel do databáze, budou přiřazeni dva oponenti (revieweři). Každý oponent by měl mít možnost ohodnotit článek na nějaké bodové škále. Řekněme například od mínus dvou do dvou bodů. V případě, že článek bude hodnocen vyšším stupněm než nula, bude zveřejněn. V opačném případě zveřejněn v dalším čísle časopisu nebude. Pokud se oponenti vzájemně shodnou, bude článek zveřejněn nebo nezveřejněn a nic dalšího se dít nebude. Za shodu oponentů bude považováno, pokud jejich hodnocení nebude od sebe rozdílné o více než dva stupně. Neshodné hodnocení tedy může nastat například v případě, že jeden oponent ohodnotí článek -2 a druhý oponent stupněm 1 nebo 2. Pokud tato situace nastane, bude přiřazen třetí oponent, který situaci rozhodne. Na následující obrázku přikládám ukázkou, jak je review proces realizovaný v OJS od nahrání článku, přes review, editaci a případné řešení problémů až dokonce procesu zveřejnění. Na druhém obrázku poté představuji vlastní návrh, jak by review mohlo v aplikaci časopisu Slovjani vypadat. Diagram byl vytvořen v online nástroji draw.io. Je dostupný na adrese <https://www.draw.io/>.

OJS Workflow Chart



Obrázek 44 – Ukázka diagramu pracovního postupu v OJS

Zdroj: (Osgeo)



Obrázek 45 – Diagram návrhu možného review procesu

Zdroj: Autor

Nakonec bych ještě rád srovnal výsledky odhadů pomocí modelu COCOMO na počet strávených Man-Days na projektu a metody funkčních bodů pro odhad řádků kódu potřebných na projekt s reálnými hodnotami.

Začnu srovnáním při použití modelu COCOMO na počet MD, které měly být stráveny na pracích na projektu. Odhad, který vyšel za použití základního modelu COCOMO při odhadované délce 1500 řádků kódu, vyšel 64 MD. Detailnější průběh a popis výpočtu jste se již dočetli v kapitole 4.7. Reálně autor na práci na celém projektu strávil 53 MD. Zahrnuji zde práce od analýzy požadavků, návrhu, implementace a testování jednotlivých sekcí. Odhad realizovaný pomocí modelu byl zde relativně přesný. Výsledek může být odlišný z důvodu, že výsledné rozšíření aplikace obsahovalo o něco méně řádků, než odhadovaných 1500. V následující tabulce si můžete podrobněji pročíst strávený čas na jednotlivých částech realizovaného projektu.

Tabulka 5 – Strávený reálný čas při práci na požadovaných funkcionalitách

Fáze při práci na aplikaci	MD (Man days)
Analýza požadavků	4
Definice uživatelských rolí	1
Návrh databázového modelu	1
Návrh případů užití	1
Návrh jednotlivých sekcí (wireframy)	4
Implementace registrace uživatelů	10
Implementace přihlášení uživatelů	6
Implementace administrace	8
Implementace vkládání návrhů článků	8
Implementace odkazů pro stahování	2
Testování	8
MD celkem:	53 MD

Dále bych chtěl porovnat odhad celkové délky kódu pomocí metody funkčních bodů. Podle výpočtu odhadu počtu řádků pomocí zmíněné metody by měl projekt čítat 3251 řádků kódu. Vlastní rozšíření aplikace obsahuje necelých tisíc řádků zdrojového kódu. Rozdíl mezi odhadem a reálnou délkou je tedy něco přes dva tisíce řádků kódu. Nutno dodat, že hodnocení důležitosti faktorů ve druhé části počtu je však značně subjektivní. Těžko se hodnotí, co je pro aplikaci jak moc důležité a tímto faktorem může být měření dost ovlivněno. Rozhodnutí zde stojí na osobních preferencích a myšlenkách hodnotitele. Navíc také za použití různých programovacích konstrukcí nebo technik se dá kód v mnoha částech zkracovat. S touto možností odhadovaný výpočet také příliš nepočítá. Délka kódu záleží na schopnostech a zkušenostech vývojáře v jednotlivých jazycích a může se lišit od očekávané. V následující tabulce je podrobněji vypsána délka realizovaného kódu v jednotlivých rozšiřujících sekcích.

Tabulka 6 – Celková délka kódu pro realizaci jednotlivých rozšiřujících funkcionalit

Realizovaná funkcionalita	Počet řádků použitého kódu v souboru
Administrace	199
Registrace	185
Přihlášení	152
Vkládání	225
Databázové připojení	14
Odhlášení	10
Odkazy (soubory archiv, poiskaj, index)	3x10
Rozšíření navigační lišty o nové sekce (všechny původní sekce v navigaci)	6x24
Celkem řádků:	959

6. Závěr

Na základě teoretické části práce se čtenář seznámil se základními pojmy, které se týkají webových stránek a webových aplikací. Dále mu také byly představeny standardní technologie a jazyky používané při tvorbě webových stránek a aplikací včetně jejich výhod a nevýhod. Konkrétně se jedná o značkovací jazyk HTML, kaskádové styly pro úpravu vzhledu CSS, skriptovací programovací jazyk PHP a Python. Základní informace jsou zde poskytnuty také o databázových systémech jako jsou MySQL a PostgreSQL. Zmíněné jsou zde také hotové redakční systémy WordPress a systém pro upravování a publikování elektronických časopisů Open Journal Systems.

V praktické části práce byla vytvořena prvotní analýzy požadavků na funkčnosti webové aplikace a definování uživatelských rolí. Následoval návrh, který zahrnoval databázový model, diagram a seznam případů užití, drátové modely (neboli wireframy) některých aplikačních sekcí a zakončen byl specifikací použitého hardwaru a softwaru v domácím prostředí. Implementace popsala podrobnější popis a postup tvorby jednotlivých aplikačních sekcí včetně detailnějšího seznámení s kódem a vzhledem jednotlivých funkcionalit. Bylo provedeno srovnání výsledků výběru mezi PHP a Pythonem. Nakonec byl porovnán vybraný jazyk PHP s hotovými redakčními systémy WordPress a Open Journal Systems.

Ve výsledcích je zmíněno počet strávených MD s analýzou, návrhem, implementací požadovaných funkcionalit a testováním. Tyto údaje jsou zde srovnány s výsledky odhadů určených pomocí modelu COCOMO na strávené Man-Days a metodou funkčních požadavků k odhadu přibližné délky kódu pro realizaci. Srovnání předpokládané délky kódu se od reality celkem liší, protože v metodě funkčních bodů je hodnocení jednotlivých aspektů značně subjektivní. Naopak model COCOMO nemá rozdíl ve strávených MD tak markantní a celkem odpovídá reálnému množství strávené práce. Z těchto informací lze odvodit, že v programovacím jazyce PHP lze vytvořit srovnatelnou, spíše o něco lepší aplikaci, než za použití redakčního systému jako jsou WordPress nebo OJS. Je za potřebí znát základní jazyky pro tvorbu webu.

Výsledkem práce je plně funkční webová aplikace, která by měla odpovídat požadavkům určeným v zadání. Vzhled je vcelku jednoduchý a intuitivní. Hotová webová aplikace je dostupná na webové adrese <http://tesar.slovjani.info/>. Práce pro mě, jakožto autora, měla také řadu osobních přínosů. Realizací jsem se zdokonalil v používání HTML, CSS, také PHP a s ním související práci s databází. Veškeré poznatky, které jsem při tvorbě práce získal, mi budou jistě užitečné v osobním i profesním životě.

Praktická část byla vytvořena na základě objednávky mezinárodní knihovny CEEOL z Německa sídlící ve Frankfurtu nad Mohanem. Tato knihovna je hlavním poskytovatelem akademických elektronických časopisů a knih týkajících se civilizace a sociálních věd ve střední a východní Evropě.

7. Seznam použitých zdrojů

Adaptic, 2018. *Programování aplikací*. [online]. [cit. 10-30-2019]. Dostupné z WWW: <https://www.adaptic.cz/weby/programovani-aplikaci/>

Adobe, 2015. *Co jsou to webové aplikace a dynamické webové stránky?* [online]. [cit. 10-30-2019]. Dostupné z WWW: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>

Altaxo. *Dynamický vs. statický web*. [online]. [cit. 03-20-2020]. Dostupné z WWW: <https://www.altaxo.cz/provoz-firmy/tvorba-webu-a-int-marketing/firemni-web/dynamicky-vs-staticky-web>

BRÁZA, Jiří, 2005. *PHP 5 začínáme programovat*. Praha : Grada, 2005. ISBN: 80-247-1146-X

CEEOL. *CEEOL - Journal detail*. [online]. [cit. 04-01-2020]. Dostupné z WWW: <https://www.ceeol.com/search/journal-detail?id=1890>

CEEOL2. *CEEOL - Home page*. [online]. [cit. 04-01-2020]. Dostupné z WWW: <https://www.ceeol.com/>

DOMES, Martin, 2012. *Tvorba WWW stránek pro úplně začátečníky*. Brno : Computer Press, 2012. ISBN 978-80-251-2160-3

Functionpoints. *Function points project estimation*. [online]. [cit. 04-02-2020]. Dostupné z WWW: <https://www.functionpoints.org/fpa-in-practice.html>

Geekforgeeks. *History of python*. [online]. [cit. 03-09-2020]. Dostupné z WWW: <https://www.geeksforgeeks.org/history-of-python/>

Geekforgeeks2. *Software engineering - COCOMO model*. [online]. [cit. 04-02-2020]. Dostupné z WWW: <https://www.geeksforgeeks.org/software-engineering-cocomo-model/>

Help.e-padi, 2019. *admin-open-journal-system.png*. [online]. [cit. 03-22-2020]. Dostupné z WWW: <https://help.e-padi.com/wp-content/uploads/2019/10/admin-open-journal-system.png>

Helpmark, 2012. *Statické webové stránky*. [online]. [cit. 03-20-2020]. Dostupné z WWW: <https://www.helpmark.cz/slovníkpojmu/46-staticke-webove-stranky>

Helpmark2, 2012. *Dynamické webové stránky*. [online]. [cit. 03-20-2020]. Dostupné z WWW: <https://www.helpmark.cz/slovníkpojmu/32-dynamicke-webove-stranky>

Hostinger, 2020. *What is MySQL: MySQL Explained for Beginners*. [online]. [cit. 03-13-2020]. Dostupné z WWW: <https://www.hostinger.com/tutorials/what-is-mysql>

HUJER, Martin, 2015. *Jaké novinky přinese PHP 7 - zdroják*. [online]. [cit. 03-09-2020]. Dostupné z WWW: <https://www.zdrojak.cz/clanky/jake-novinky-prinese-php-7/>

Jakpsatweb, 2008. *Úvod do CSS*. [online]. [cit. 03-08-2020]. Dostupné z WWW: <https://www.jakpsatweb.cz/css/css-uvod.html>

Javatpoint, 2016. *Python history*. [online]. [cit. 03-09-2020]. Dostupné z WWW: <https://www.javatpoint.com/python-history>

Kinsta.com, 2020. *What is Wordpress? Explained for Beginners*. [online]. [cit. 03-18-2020]. Dostupné z WWW: <https://kinsta.com/knowledgebase/what-is-wordpress/>

Klikzone, 2011. *HTML návod - HTML atributy*. [online]. [cit. 03-08-2020]. Dostupné z WWW: <https://www.klikzone.cz/HTML-navod/HTML-atributy.php>

Knihovna.cvut, 2019. *Open Journal System*. [online]. [cit. 03-20-2020]. Dostupné z WWW: <http://knihovna.cvut.cz/redakce/podpora-pro-redakce/open-journal-system>

KOSEK, Jiří, 1999. *PHP - tvorba interaktivních internetových aplikací: podrobný průvodce*. Praha : Grada, 1999. ISBN: 80-716-9373-1

Kumarsher45. *html-5-page-structure.jpg (835 x 541)*. [online]. [cit. 03-22-2020]. Dostupné z WWW: <https://kumarsher45.files.wordpress.com/2015/01/html-5-page-structure.jpg>

Linuxexpres, 2012. *Otevřený přístup a open source v kontextu akademického publikování*. [online]. [cit. 03-20-2020]. Dostupné z WWW: <https://www.linuxexpres.cz/business/otevreny-pristup-a-open-source-v-kontextu-akademického>

Live.staticflickr. *44661017890_eb4f288ffe_b.jpg*. [online] [cit. 03-22-2020]. Dostupné z WWW: https://live.staticflickr.com/4836/44661017890_eb4f288ffe_b.jpg

Managementmania, 2018. *Webová stránka (Internetová stránka)*. [online]. [cit. 10-22-2019]. Dostupné z WWW: <https://managementmania.com/cs/webova-stranka-internetova>

Medium, 2017. *Advantages and Disadvantages of Python Programming Language*. [online]. [cit. 03-13-2020]. Dostupné z WWW: <https://medium.com/@mindfiresolutions.usa/advantages-and-disadvantages-of-python-programming-language-fd0b394f2121>

Metropoliscreative, 2019. *install-classic-editor-wordpress-plugin.png*. [online]. [cit. 03-22-2020]. Dostupné z WWW: <https://www.metropoliscreative.com/wp-content/uploads/2019/01/install-classic-editor-wordpress-plugin.png>

Oracle. *Co je relační databáze*. [online]. [cit. 03-13-2020]. Dostupné z WWW: <https://www.oracle.com/cz/database/what-is-database.html>

Osgeo. *Ojs_workflow.png (634 x 656)*. [online]. [cit. 04-01-2020]. Dostupné z WWW: https://wiki.osgeo.org/images/f/f4/Ojs_workflow.png

Pehapko, 2018. *Základní kurz 1: Úvod*. [online]. [cit. 03-08-2020]. Dostupné z WWW: <http://www.pehapko.cz/zakladni-kurz/1-uvod>

Php, 2017. *PHP: History of PHP - Manual*. [online]. [cit. 03-09-2020]. Dostupné z WWW: <https://www.php.net/manual/en/history.php.php>

Pphbabu, 2017. *Advantages and disadvantages of PHP*. [online]. [cit. 03-09-2020]. Dostupné z WWW: <https://www.phpbabu.com/advantages-and-disadvantages-of-php/>

PILGRIM, Marek, 2010. *Ponořme se do Python(u) 3*. Praha : CZ.NIC, 2010. ISBN: 978-80-904248-2-1

Pkp.sfu, 2014. *Open Journal Systems | Public Knowledge Project*. [online]. [cit. 03-20-2020]. Dostupné z WWW: <https://pkp.sfu.ca/ojs/>

Postgresql, 2018. *PostgreSQL: About*. [online]. [cit. 03-13-2020]. Dostupné z WWW: <https://www.postgresql.org/about/>

PROCHÁZKA, David, 2012. *PHP 6: začínáme programovat*. Praha : Grada, 2012. ISBN: 978-80-247-7578-4

Pythonforbeginners, 2018. *What is Python?* [online]. [cit. 03-09-2020]. Dostupné z WWW: <https://www.pythonforbeginners.com/learn-python/what-is-python/>

Sprava-site. 2016. *Co je to databázový systém?* [online]. [cit. 03-13-2020]. Dostupné z WWW: <https://www.sprava-site.eu/databazovy-system/>

Squareboat, 2019. *Advantages and Disadvantages of Python*. [online]. [cit. 03-13-2020]. Dostupné z WWW: <https://squareboat.com/blog/advantages-and-disadvantages-of-python>

Squelle, 2018. *Kombinování PHP a HTML kódu v Drupalu*. [online]. [cit. 03-09-2020]. Dostupné z WWW: <http://www.squelle.com/cs/publikace/kombinovani-php-html-kodu>

Themeisle.com, 2020. *What is WordPress & What is it used for? Beginner's guide (2020)*. [online]. [cit. 03-18-2020]. Dostupné z WWW: <https://themeisle.com/blog/what-is-wordpress/>

Tutroialspoint, 2016. *Python - Basic Syntax*. [online]. [cit. 03-09-2020]. Dostupné z WWW: https://www.tutorialspoint.com/python/python_basic_syntax.htm

Tvorba-webu, 2016. *PHP /základy/*. [online]. [cit. 03-08-2020]. Dostupné z WWW: <https://www.tvorba-webu.cz/php/>

Vertex-academy, 2016. *The history of the developement of HTML*. [online]. [cit.02-26-2020]. Dostupné z WWW: <https://vertex-academy.com/tutorials/en/html-history/>

VRÁNA, Jakub, 2012. *1001 tipů a triků pro PHP*. Brno : Computer Press, 2012. ISBN: 978-80-251-2940-1

W3schools, 2017. *Introduction to HTML*. [online]. [cit. 02-26-2020]. Dostupné z WWW: https://www.w3schools.com/html/html_intro.asp

Washington. *WebD2: A Brief History of HTML*. [online]. [cit. 03-20-2020]. Dostupné z WWW:

https://www.washington.edu/accesscomputing/webd2/student/unit1/module3/html_history.html

Wedesin.cz, 2016. *Co je to WordPress a jaké jsou jeho výhody a nevýhody?* [online]. [cit. 03-18-2020]. Dostupné z WWW: <https://www.wedesin.cz/co-je-to-wordpress-a-jake-jsou-jeho-vyhody-a-nevyhody>

Wpapprentice.com, 2016. *What is WordPress? A Simple Explanation For Beginners.* [online]. [cit. 03-18-2020]. Dostupné z WWW: <https://wpapprentice.com/blog/what-is-wordpress/>

8. Přílohy

8.1 Seznam obrázků

Obrázek 1 – Časopis Slovjani.info jako součást knihovny CEEOL	12
Obrázek 2 – Formulář pro registraci uživatelů na vlastní stránce	16
Obrázek 3 – Struktura HTML dokumentu	21
Obrázek 4 – Přímý zápis stylu do elementu	22
Obrázek 5 – Zápis stylů pomocí tagu style	22
Obrázek 6 – Odkaz na externí soubor styly.css	22
Obrázek 7 – Princip fungování PHP	24
Obrázek 8 – Příklad bloků příkazů v PHP	26
Obrázek 9 – HTML kód v PHP kódu	27
Obrázek 10 – PHP kód vložený do kódu HTML	27
Obrázek 11 – Jednotlivé verze jazyka Python	30
Obrázek 12 – Ukázka kódu v Pythonu	31
Obrázek 13 – Princip fungování MySQL	34
Obrázek 14 – Využití WordPressu na webu celkově	35
Obrázek 15 – Ukázka rozhraní WordPressu	36
Obrázek 16 – Ukázka administrátorského rozhraní OJS	38
Obrázek 17 – Datový model systému (ER diagram)	43
Obrázek 18 – Diagram případů užití	44
Obrázek 19 – Wireframe registrace uživatele	46
Obrázek 20 – Wireframe přihlášení uživatele	47
Obrázek 21 – Wireframe administrace	48
Obrázek 22 – Wireframe vkládání návrhů článků	48
Obrázek 23 – Wireframe odkazu pro stahování	49
Obrázek 24 – Ukázka prostředí programu FileZilla	51
Obrázek 25 – Kód realizující připojení do databáze	52
Obrázek 26 – Kód prováděný po kliknutí na tlačítko registrace	53
Obrázek 27 – Ukázka chybových hlášek formuláře	53
Obrázek 28 – Struktura tabulky uživatelé	54
Obrázek 29 – Kompletní sekce registrace	54
Obrázek 30 – Kód provedený po kliknutí na tlačítko přihlásit	55
Obrázek 31 – Kód zobrazující informace o přihlášeném uživateli	55
Obrázek 32 – Finální vzhled sekce přihlášení a informace o přihlášeném	56
Obrázek 33 – Kód obsluhující tlačítko pro odhlášení	56
Obrázek 34 – Kód pro realizaci výpisu dat z databáze do tabulky	57
Obrázek 35 – Informace o stavu registrace	58
Obrázek 36 – Provedený kód po stisknutí tlačítka ulož změny	58
Obrázek 37 – Kód provedený po stisku tlačítka smaž řádek	59
Obrázek 38 – Finální vzhled sekce administrace s chybovými hláškami	59
Obrázek 39 – Podmínka viditelnosti sekce	60
Obrázek 40 – Chybové hlášky	61

Obrázek 41 – Kompletní vzhled sekce vkládání návrhů článků	61
Obrázek 42 – Kód realizace odkazu.....	62
Obrázek 43 – Odkazy u každého hotového čísla časopisu.....	62
Obrázek 44 – Ukázka diagramu pracovního postupu v OJS.....	71
Obrázek 45 – Diagram návrhu možného review procesu	72

8.2 Seznam tabulek

Tabulka 1 – Požadavky na jednotlivé funkcionality aplikace.....	40
Tabulka 2 – Uživatelské role figurující v aplikaci	41
Tabulka 3 – Seznam případů užití.....	45
Tabulka 4 – Klíčová kritéria při srovnání PHP a redakčních systémů.....	69
Tabulka 5 – Strávený reálný čas při práci na požadovaných funkcionalitách	73
Tabulka 6 – Celková délka kódu pro realizaci jednotlivých rozšiřujících funkcionalit.....	74