

Univerzita Hradec Králové
Filozofická fakulta

Bakalářská práce

2019

Marek Dolníček

Univerzita Hradec Králové

Filozofická fakulta

Katedra pomocných věd historických a archivnictví

Systém sledující stav a dostupnost informačních systémů archivu

Bakalářská práce

Autor: Dolníček Marek

Studijní program: B3928 - Technická podpora humanitních věd

Forma studia: kombinovaná

Studijní obor: Počítačová podpora v archivnictví (BPARCHIV)

Vedoucí práce: Ing. Monika Borkovcová, Ph.D.

Hradec Králové, 2019



Zadání bakalářské práce

Autor: Marek Dolníček

Studium: F16BK0021

Studijní program: B3928 Technická podpora humanitních věd

Studijní obor: Počítačová podpora v archivnictví

Název bakalářské práce: **Systém sledující stav a dostupnost informačních systémů archivu**

Název bakalářské práce AJ: Monitoring system of status and availability of archive information systems

Cíl, metody, literatura, předpoklady:

Cílem práce je analýza informačních systémů a aplikací využívaných v archivu a vytvoření uživatelsky přívětivého systému pro sledování dostupnosti těchto systémů. Informační systémy a aplikace v archivu jsou provozovány na různých platformách, což vede k problematické správě. Cílem praktické části závěrečné práce je tak vytvořit jednotný systém, který umožní sledovat aktivity v jednotném přehledu a v případě výpadku či technické chyby upozornit správce formou notifikace o nastalé události. Systém sledující dostupnost systémů v archivu bude vytvořen jako webová aplikace.

NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7.
CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. ISBN isbn-978-80-251-3733-8. ŠTOURAČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Spisová a archivní služba ve státní správě, samosprávě a v podnikatelské sféře:3. aktualizované a přepracované vydání. Praha: Linde, 2005, s. 13. ISBN 80-7201-549-4. HLEDÍKOVÁ-Zdeňka, JANÁK-Jan, DOBEŠ-Jan. Dějiny správy v českých zemích: od počátku státu po současnost. Praha: Lidové noviny, 2005, s. 570. ISBN: 978-80-7106-906-5. MATYSKA, Luděk. Virtualizace výpočetního prostředí. Zpravodaj ÚVT MU. 2006, roč. 17, č. 2, s. 9-11. DOI: ISSN 1212-0901. BRIAN, P. Hogan. HTML5 a CSS3, Výukový kurz webového vývoje. Brno: Computer Přes, 2011. ISBN 978-80-251-3576-1. LECHNER, Tomáš, 2013. Elektronické dokumenty v právní praxi. 1. vyd. Praha: Leges, 256 s. ISBN 978-80-87576-41-0.

Garantující pracoviště: Katedra pomocných věd historických a archivnictví,
Filozofická fakulta

Vedoucí práce: Ing. Monika Borkovcová, Ph.D.

Oponent: Mgr. Radek Pokorný

Datum zadání závěrečné práce: 9.11.2018

Prohlášení studenta

Čestně prohlašuji, že tato práce je mým vlastním autorským dílem. Práci jsem vypracoval samostatně a uvedl jsem všechny prameny, literaturu a zdroje, které jsem při vypracování práce použil nebo z nich čerpal.

.....

V Hradci Králové dne

Marek Dolníček

Poděkování

Rád bych poděkoval Ing. Monice Borkovcové, Ph.D za vedení práce a odborné rady při jejím zpracování. Mé poděkování patří také panu Ing. Michalovi Munzarovi, DiS za poskytnuté informace a pomoc při jejich zpracování.

Anotace

DOLNÍČEK, MAREK. Systém sledující stav a dostupnost informačních systémů archivu. Hradec Králové. Filozofická fakulta, Univerzita Hradec Králové, 2019, 75 str. Bakalářská práce.

Cílem práce je analýza informačních systémů a aplikací využívaných v archivu a vytvoření uživatelsky přívětivého systému pro sledování dostupnosti těchto systémů. Informační systémy a aplikace v archivu jsou provozovány na různých platformách, což vede k problematické správě. Cílem praktické části závěrečné práce je tak vytvořit jednotný systém, který umožní sledovat aktivity v jednotném přehledu a v případě výpadku či technické chyby upozornit správce formou notifikace o nastalé události. Tato notifikace bude probíhat formou emailu, který upozorní správce archivu na daný výpadek. Systém sledující dostupnost systémů v archivu bude vytvořen jako webová aplikace, která bude dostupná externě z webového prohlížeče. Systém bude využívat programovací jazyk PHP, JavaScript a databázi MySQL.

Klíčová slova:

Archivní systém, notifikace, servery, virtualizace

Annotation

DOLNÍČEK, MAREK. Monitoring system of status and availability of archive information systems. Hradec Králové. Philosophical Faculty, University of Hradec Králové, 2018, 75 pp., Bachelor Degree Thesis.

The main goal of this work is to analyze information systems and applications used in the archive and create a user-friendly system for monitoring the availability of these systems. Information systems and applications in the archive run on different platforms, leading to problematic management. The main goal of a practical part of the thesis is to create a unified system that enables to monitor activities in a unified overview and in case of failure or technical error notify the administrator in the form of notification of the event. This notification will take the form of an email that notifies the archive manager of a given downtime. The system availability monitoring system in the archive will be created as a web application that will be available externally from the web browser. The system will use PHP, JavaScript and MySQL.

Key words:

Archive system, notifications, servers, virtualization

Obsah

Úvod.....	9
1 Virtualizace.....	11
1.1 Server a jeho charakteristika.....	12
1.2 Operační systémy využívané na serverech.....	14
1.3 Využití virtualizace.....	16
1.3.1 Docker.....	16
1.3.2 Fungování Dockeru.....	17
1.3.3 Příkazy v Dockeru.....	17
2. Technologie pro tvorbu webové aplikace.....	18
2.1 Backend.....	18
2.1.1 Relační databáze MySQL.....	18
2.1.2 Programovací jazyk PHP.....	19
2.2 Frontend.....	21
2.2.1 HTML a CSS.....	21
2.2.2 JavaScript.....	23
3. Archivnictví.....	26
3.1 Pojem a definice oboru.....	26
3.1.1 Historie archivnictví.....	28
3.1.2 Třídění archiválií.....	29
4. Spisová služba.....	32
4.1 Vývoj spisové služby.....	32
4.2 Definice spisovny.....	35
4.3 Komerční spisovna.....	37
4.4 Skartace.....	38

4.5 Dnešní stav spisové služby	41
5. Systémy v archivní praxi a spisovnách.....	43
5.1 Přehled a popis systémů využívaných v archivech.....	43
5.2 Přehled a popis systémů používaných pro spisovou službu	46
6. Vývoj webové aplikace.....	48
6.1 Návrh systému	48
6.1.1 Grafický návrh webové aplikace.....	48
6.1.2 Návrh databáze.....	51
6.2 Konfigurace webových služeb.....	54
6.3 Tvorba vývojového prostředí.....	59
6.3.1 Vytvoření Databáze.....	59
6.3.2 Vytvoření systému pro sledování serverů a zasílání notifikací.....	60
6.3.3 Vývoj webového systému	62
Závěr	68
Použité zdroje a literatura	70
Seznam příloh	74

Seznam zkratk

CSS – Cascading Style Sheets, kaskádové styly pro formátování webových stránek.

EMAIL – Electronic Mail, elektronická pošta

GUI – Graphical User Interface, grafické uživatelské rozhraní.

HTTP – HyperText Transfer Protocol, internetový protokol, který slouží k výměně hypertextových dokumentů mezi serverem a prohlížečem.

HTTPS – Hypertext Transfer Protocol Secure, zabezpečený internetový protokol, který slouží k výměně hypertextových dokumentů mezi serverem a prohlížečem.

IP – Internet Protocol, identifikace konkrétního zařízení.

MySQL – My Structured Query Language, systém pro řízení databází

PC – personal computer, běžný počítač používaný v domácnostech či institucích.

PHP – Personal Home Page, skriptovací programovací jazyk.

SMTP – Simple Mail Transfer Protocol, internetový protokol určený pro přenos zpráv elektronické pošty

SQL – Structured Query Language, standardizovaný strukturovaný dotazovací jazyk.

URL – Uniform Resource Locator, soubor znaků, které slouží k identifikaci přesného umístění informací v rámci internetové sítě.

WWW – World Wide Web, světová komunikační síť.

NAD – Národní archivní dědictví

Úvod

V dnešní době jsou informační technologie nedílnou součástí institucí a organizací, a tedy i archivů, které spravují velké množství koncových zařízení, které jsou připojeny na centrální systémy v příslušném archivu. Všechny tyto systémy jsou většinou připojeny v rámci jedné sítě a jsou udržovány správcem počítačové sítě daného institutu. Monitoring sítě v archivu není pravidlem, spíše naopak, a tak správce počítačové sítě archivu nemá, většinou, možnost jednoduše sledovat provoz v síti. Autorovou motivací z tohoto důvodu bylo vytvoření jednoduchého a přehledného systému, který upozorní správce počítačové sítě a informačních technologií v archivu, který tak bude moci okamžitě zasáhnout a případně obnovit funkčnost diskových polí, samotných serverů či jiných síťových prvků, u kterých byl nalezen výpadek.

Cílem práce je vytvořit ucelený systém, který bude sloužit jako přehled o aktivitách serverů a sledování činností s tím spojených. Celý tento systém bude z důvodu uživatelské přívětivosti dostupný z webového prohlížeče. Webová aplikace se zaměří na jednoduchý přístup k přehledu aktivních serverů a v případě výpadku zobrazí výpadek na digitální nástěnce. Následně zašle upozornění o výpadku na zadaný administrátorský email, který lze spravovat v rámci konfigurace webové aplikace.

Práce bude tedy zaměřena na řešení v problematice nedostupnosti používaných serverů v archivech. Cílem bude vytvořit takové řešení, které nebude složité na implementaci a bude sloužit jako přehled pro správce informačních technologií v dané instituci. Proces tvorby v praktické části cílí na nejnovější trendy a dostupná řešení pro tvorbu webových aplikací.

Teoretická část práce se bude věnovat použitým a souvisejícím technologiím praktického výstupu, jako jsou webové služby, databázové systémy či programovací jazyky, ve kterých jsou vyvíjeny webové aplikace, dále se práce soustředí na základní výklad v oblasti webových serverů a virtualizace. Čtenář se také dozví, jak funguje v praxi relační databáze MySQL, programovací jazyk PHP či v dnešní době stále populárnější JavaScript.

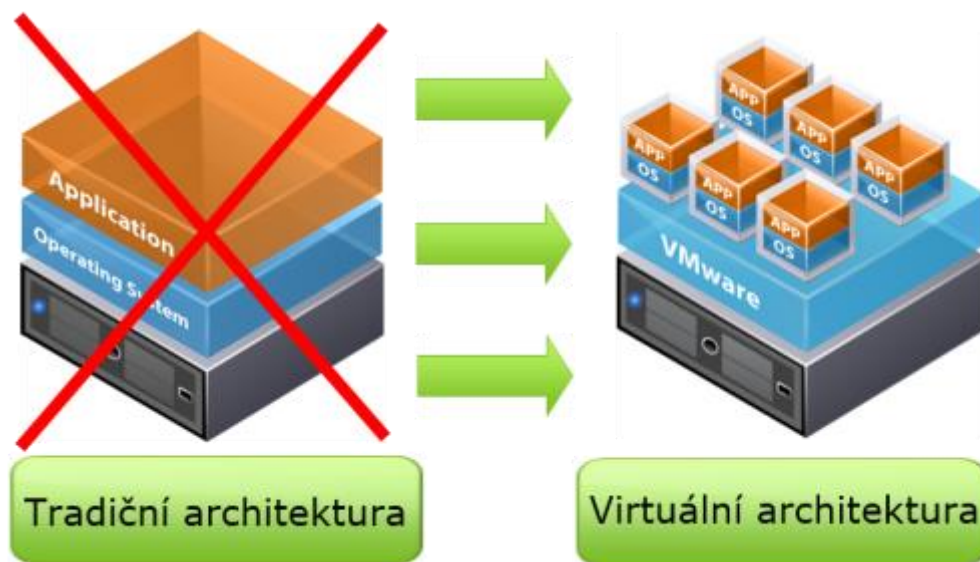
Samotnému vývoji se věnuje kapitola, která čtenáře seznámí s moderními trendy a postupy při analýze, návrhu a programování webových aplikací. Neméně důležitou částí práce jsou samotné archivní systémy, kterou budou monitorovány pomocí praktického výstupu. Další část práce se tak zaměří na samotné archivnictví, kdy tato práce popisuje jeho fungování a historický vývoj. Dále budou rozebrány jednotlivé informační systémy, které jsou nejčastěji využívány v archivech či ve státní nebo veřejné správě.

Na základě metod analýzy a syntézy využívá praktická část práce teoretických poznatků z obou popisovaných oblastí a celý postup tvorby bude zpracován jako celkový vývoj webové aplikace. Součástí bude nejen lokální vývoj, ale i nasazení na fungující server umístěný v archivu.

Přínos výsledné aplikace bude kladen především na jednoduchost a funkčnost notifikací. Navržená databáze výsledného systému bude vytvořena s ohledem na jednoduchost, přehlednost a možnou budoucí rozšiřitelnost. Cílem práce je nabídnout jednoduchou editaci záznamů, jako je úprava či přidávání nových serverů či jednotlivých uživatelů v roli administrátorů. Celý systém bude mít možnost dostupnosti jak v interní síti, tak i externě formou šifrovaného přihlášení přes webový prohlížeč.

1 Virtualizace

Pod pojmem virtualizace si lze představit technologii, která vytváří kopie určitých zdrojů, kterými mohou být například procesor počítače, paměti či disky. Tyto kopie vznikají formou konceptů, které se nazývají virtuální objekty. Tyto objekty mají virtuální disk, paměť a procesor. Tímto způsobem lze tak uživateli poskytnout celý virtuální počítač, který je sestaven z jednotlivých virtuálních komponent. Kontrola nad celým virtuálním počítačem je tak plně řízena uživatelem, který ovšem sdílí fyzické komponenty s dalšími uživateli, kteří využívají daný fyzický stroj. ¹



Obr. 1: Ukázka virtualizace ²

Ve světě informačních technologií existuje více druhů virtualizace, ale nejdůležitější je plán virtualizace, při něm dochází k simulaci hardwaru, na kterém je pak možné spustit operační systém bez úprav, a fungují tedy stejně, jako kdyby reálně běžely na plnohodnotném hardware umístěných na fyzických serverech.

Možnosti, které virtualizace nabízí: ³

¹ MATYSKA, Luděk. Virtualizace výpočetního prostředí. Zpravodaj ÚVT MU. 2006, roč. 17, č. 2, s. 9-11. DOI: ISSN 1212-0901. Dostupné z: <http://webserver.ics.muni.cz/bulletin/articles/540.html>

² Co je to virtualizace?. Oldanygroup.cz [online]. [cit. 2019-03-29]. Dostupné z: <http://www.oldanygroup.cz/virtualizace-vmware-zakladni-informace-9/>

³ MATYSKA, Luděk. Virtualizace výpočetního prostředí. Zpravodaj ÚVT MU. 2006, roč. 17, č. 2, s. 9-11. DOI: ISSN 1212-0901. Dostupné z: <http://webserver.ics.muni.cz/bulletin/articles/540.html>

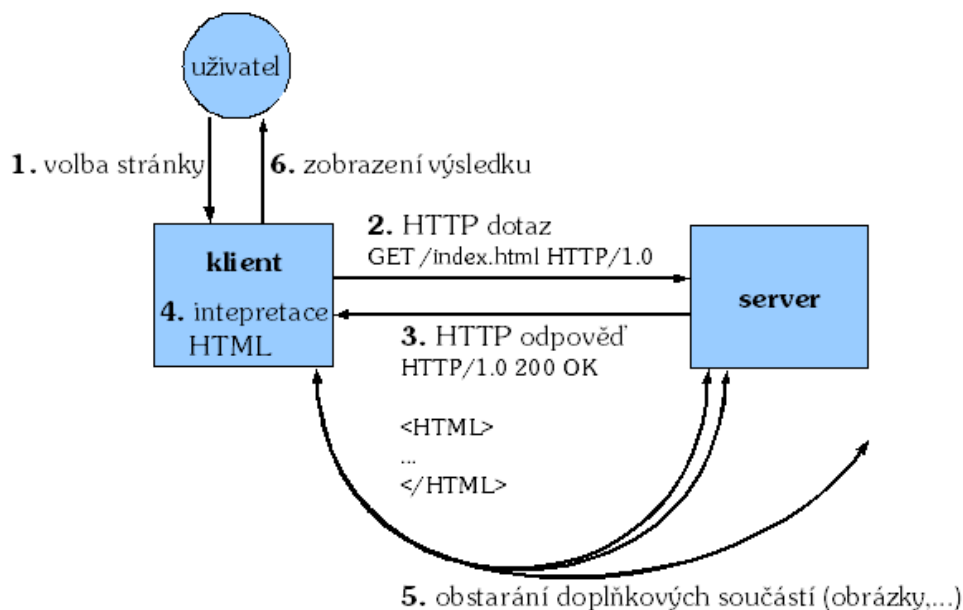
- **Jednoduchá migrace** - možnost přenést běžící virtuální stroj mezi dvěma fyzickými servery bez nutnosti restartu či výpadku používaných služeb.
- **Záloha a obnova dat** – pomocí tzv. snapshotů je možné jednoduše v daném okamžiku zálohovat obrazy pevného disku a paměti.
- **Testování a vývoj** – virtuální stroje jsou ideální pro použití jako je testovací prostředí či vývoj aplikací. Je možné je například vzdáleně restartovat, vypínat či kompletně odstavovat mimo síť.
- **Dostupnost a spolehlivost** – jednoduchá manipulace běžících virtuálních strojů mezi stroji fyzickými. Dále je také možnost nastavit u virtuálních serverů vyvážení zátěže, která se automaticky rozloží do fyzických serverů, které jí mohou poskytnout dostatečný výkon.

1.1 Server a jeho charakteristika

Server jako pojem je označení pro počítač, který má jak hardwarovou, tak softwarovou část. Tento fyzický stroj pak poskytuje různé služby dalším počítačům a aplikacím, které jsou na server napojeny. Služby mohou být různého charakteru, ale nejčastějším je forma počítačové sítě, kde jsou sdílené disky, tiskárny, autorizační servery, webové servery a další. Server může být, ale také využíván pro jeden počítač, kdy se stará o jeho připojené periferie. Počítače, které se připojují k serveru jsou nazývány klienty. Architektura, která se využívá tento princip se označuje jako architektura klient-server a je součástí většiny moderních systémů. ⁴

⁴ Jaké jsou druhy serveru. Jaknahosting.cz [online]. 13. srpna 2017 [cit. 2019-03-29]. Dostupné z: <http://jaknahosting.cz/jake-jsou-druhy-serveru/>

Komunikace klient-server



Obr. 2: Ukázka architektury klient-server ⁵

Druhy serverů, které se v současnosti vyskytují: ⁶

- **Souborový server** slouží pro ukládání dokumentů, kdy představuje centrální úložiště. Služba, kterou tento server vykonává se jmenuje Cloud.
- **Relační databázový server** se používá na ukládání strukturovaných dat, na které jsou napojeny informační systémy. Server poté vykonává i dotazy, které jsou prováděny pomocí webových serverů, a zpracovává je.
- **Webový server** zobrazuje webové stránky skrze jazyk HTML a může vykonávat dotazy pomocí různých programovacích jazyků, jako je PHP či JavaScript. Uživatelé si mohou zobrazit stránky přes protokol HTTP či dnes již standardizovaný šifrovaný protokol HTTPS.
- **Proxy server** zpracovává uživatelské dotazy a umožňuje uživatelům přístup do jiné sítě, jako je například Internet.

⁵ Jaké jsou druhy serveru. Jaknahosting.cz [online]. 13. srpna 2017 [cit. 2019-03-29]. Dostupné z: <http://jaknahosting.cz/jake-jsou-druhy-serveru/>

⁶ Tamtéž

- **Tiskový server** komunikuje se všemi tiskárnami, které jsou napojeny na stejnou síť, ve které se nachází server, a lze tak používat služby tiskáren jak interně, tak i externě.
- **Aplikační server** představuje specializovaný server pro běh určité aplikace. Jsou zde většinou umístěny speciální moduly pro podporu nestandardního programovacího jazyka či protokolu.

1.2 Operační systémy využívané na serverech

Operační systémy, které se běžně instalují do serverů, jsou uzpůsobené speciálně na komunikaci klient-server na přijímání a odpovídání požadavků klientů v rámci sítě. Tyto systémy kladou důraz především na stabilitu, protože jsou v provozu neustále a mají požadavky na výkon tak, aby spotřebovávaly co nejmenší zdroje z fyzického serveru a soustředily se na samotné procesy, které jsou v těchto operačních systémech uskutečňovány. U serverů se můžeme setkat nejčastěji pouze s příkazovou řádkou, tedy bez grafické nadstavby, ale existují i systémy, které tuto nadstavbu označovanou jako grafické uživatelské rozhraní (GUI) mají. Většina serverových operačních systémů obsahuje také nástroje pro zálohování a obnovení dat, aby v případě výpadku či náhlého výpadku fyzické komponenty bylo možné jednoduše vrátit systém do původního stavu.⁷

Pro zvolení vyhovujícího operačního systému se porovnává:

- **Bezpečnost**, především jaké funkce nabízí daný systém z pohledu zabezpečení a možnosti šifrování.
- **Administrace**, zdali má operační systém GUI či pouze příkazovou řádku neboli terminál.
- **Výkon**, jaké jsou hardwarové požadavky pro chod systému.
- **Škálovatelnost**, tedy kolik klientů může mít současně přístup v rámci systému.
- **Stabilita**, jak dlouho v porovnání s ostatními systémy dokáže pracovat bez výpadku či nějaké nečekané chyby.

⁷ Jaké jsou druhy serveru. Jaknahosting.cz [online]. 13. srpna 2017 [cit. 2019-03-29]. Dostupné z: <http://jaknahosting.cz/jake-jsou-druhy-serveru/>

- **Podpora aplikací**, které jsou vyvíjeny vývojáři třetích stran.

Operační systémy mohou být dostupné buď zdarma formou open source kódu nebo k jejich použití na serverech je potřeba licence k užití. Mezi nepoužívanější operační systémy určené pro servery patří: ⁸

Windows Server (2003, 2008, 2012, 2016 či 2019), tento operační systém je určen pro malé servery, které se nachází spíše v soukromé sféře. Vyznačuje se uživatelsky přívětivým prostředím s možností GUI. Dále jsou jeho výhody ve velké podpoře aplikací, včetně virtualizačního nástroje Docker, ze studií třetích stran a dobrá správa souborového systému či údržba a instalace. Jako nevýhodu tohoto systému můžeme považovat nutnost placené licence k provozu a velmi omezené možnosti modifikace samotného systému v případě detailnější úpravy. ⁹

Linux je volně dostupný operační systém, který nabízí velké množství balíčků, které tento systém rozšiřují. Některé balíčky ovšem mohou být zpoplatněné, ale většina z nich je také volně přístupná. Většinou se v Linuxu používá příkazový řádek, který není náročný na výkon, a tím se zvyšuje celková produktivita spouštěných aplikací, která je oproti systémům s GUI lepší. Linuxové systémy nabízejí dobrou správu souborů a sdílených tiskáren. Často se také používají jako databázové a emailové servery. Mezi nejznámější distribuce Linuxu patří:

- **Ubuntu server** je stabilní serverový systém, který je vhodný pro malé firmy především na správu uživatelů a sítě a také nabízí možnost administrace s grafickou nadstavbou neboli GUI.
- **CentOS** je systém optimalizovaný pro správu webových služeb a je tedy často využíván hostingovými firmami. Jeho výhodami je především jeho malá náročnost a bezpečnost při využití v moderních webových projektech.
- **Debian** byl předchůdce Ubuntu serveru a je dnes považován za nejstabilnější linuxovou distribuci. Vše v rámci Debianu je volně stažitelné a není potřeba žádné placené služby.

⁸ KONICK, Sinbad. Top 5 Server Operating Systems. Slideshare.net [online]. 3. prosince 2016 [cit. 2019-03-29]. Dostupné z: <https://www.slideshare.net/BlackDetah/top-5-server-operating-systems>

⁹ Tamtéž

- **Red Hat** nabízí kvalitní řešení především pro datacentra, kdy tento systém využívají jedny z největších společností na světě.¹⁰

FreeBSD je systém založený stejně jako Linux na Unixovém jádře. Ideální především pro služby, které potřebují velký výkon, je oblíbený kvůli své jednoduchosti použití a nejvyšší stabilitě ze všech zmíněných systémů. Nabízí také dobrou podporu doplňujících balíčků bez nutnosti vlastnit placenou licenci.¹¹

Mac OS X Server serverový systém vyvíjen firmou Apple. Využívá velkou část jádra FreeBSD a tudíž zaručuje i výbornou rozšiřitelnost a stabilitu celého systému. Pro chod tohoto systému je ale potřeba využívat hardware od společnosti Apple. Velkou výhodou tohoto řešení jsou neomezené uživatelské licence.¹²

Solaris open source systém, který je využíván malými až středně velkými podniky.¹³

1.3 Využití virtualizace

1.3.1 Docker

Nástroj Docker poskytuje virtualizaci operačního systému pomocí kontejnerů, kterých může být v rámci Dockeru několik. Hlavní přínos, proč používat Docker je možnost spouštět jednotlivé programy bez závislosti na operačním systému a následně tyto programy přenášet mezi jednotlivými systémy a udržet si jejich strukturu. Jednotlivé kontejnery jsou izolované, tudíž neovlivňují ostatní kontejnery a v případě neočekávané chyby tak nepadne celý fyzický stroj. Hlavními výhodami Dockeru jsou tedy přenositelnost, testování a nasazování systémů rychle, aby nedocházelo k velkým výpadkům na produkčním serveru.^{14 15}

Důvodem, proč je Docker tak oblíbený u vývojářů, je jeho jednoduchá přenositelnost mezi počítači. V praxi to tak znamená, že systémy, které se spouští

¹⁰ KONICK, Sinbad. Top 5 Server Operating Systems. Slideshare.net [online]. 3. prosince 2016 [cit. 2019-03-29]. Dostupné z: <https://www.slideshare.net/BlackDetah/top-5-server-operating-systems>

¹¹ Tamtéž

¹² Tamtéž

¹³ Tamtéž

¹⁴ Docker overview. Docker docs [online]. [cit. 2018-04-30]. Dostupné z: <https://docs.docker.com/engine/docker-overview/>

¹⁵ Tamtéž

v Dockeru, tak fungují na všech počítačích stejně a není problém je spouštět ve stejné konfiguraci, pouze na jiných strojích. Vývojářům to tak přinese stejné vývojové prostředí, které používají i jejich spolupracovníci.¹⁶

1.3.2 Fungování Dockeru

Základními prvky Dockeru jsou kontejnery, které Docker izoluje pomocí *Namespaces*, což jsou virtuální prostory, které obsahují takzvané izolační aspekty.

Mezi izolační aspekty kontejneru můžeme považovat:

- net - systém pro správu síťových rozhraní
- ipc - sdílená paměť, která se prolíná v rámci kontejnerů
- pid - každý spuštěný proces má přidělené id, kterým se identifikuje¹⁷

1.3.3 Příkazy v Dockeru

Docker je shluk objektů, které pracují s jednotlivými kontejnery, obrazy či pluginy. Obraz je jakási forma šablony, která slouží pro vytvoření kontejneru, který je pak spouštěn. Pro vytvoření vlastních obrazů, které jsou definované pomocí šablon se nazývá „Dockerfile“. V Dockerfile se pak provádějí jednotlivé operace:

BUILD sestavuje obraz, který se spouští přes Docker.

IMAGE slouží k definování obrazu, ze kterého se dál vytváří kontejner.

WORKING_DIR je příkaz, který definuje pracovní adresář pro určitou službu.

CONTAINER_NAME je název kontejneru pro danou službu.

PORTS slouží pro určení, které porty jsou veřejnými porty v rámci kontejneru.

VOLUMES využívá se na propojení cest ze všech adresářů do jednoho svazku.

LINKS odkazuje na kontejnery, které se nacházejí v jiné službě.

COMMAND přepisuje výchozí příkaz, který je nastaven v Dockerfile souboru.

¹⁶ Docker overview. Docker docs [online]. [cit. 2018-04-30]. Dostupné z: <https://docs.docker.com/engine/docker-overview/>

¹⁷ Tamtéž

2. Technologie pro tvorbu webové aplikace

Při tvorbě webových aplikací jsou důležité technologické znalosti z databázových systémů, programovacích jazyků a grafických návrhů, které se využívají při tvorbě webových aplikací. Mezi nejpoužívanější jazyky patří PHP, JavaScript, CSS či HTML. Většina těchto moderních jazyků se v poslední době stále prudce vyvíjí, proto je nutné sledovat nejnovější trendy a standardy. V rámci vývoje webových aplikací se rozlišuje vývoj na úrovni systému, který funguje na pozadí celé aplikace a grafické nadstavby, která vykonává funkci pro komunikaci aplikace mezi uživatelem a serverem.¹⁸

2.1 Backend

Pojmem backend se označuje ta část webové aplikace, která slouží ke zpracování dat a administraci dané webové stránky. Obstarává komunikaci mezi databází a serverem, kdy při tom využívá programovacích jazyků, jako mohou být Ruby, Python, Java či C++. Jako ukázka může být fungování internetového obchodu, kdy jako backend se považuje systém, který slouží ke vkládání nového zboží, úprava vlastností a cen.¹⁹

2.1.1 Relační databáze MySQL

Jedná se o populární databázový systém, který vznikl v polovině devadesátých let a v roce 2019 byla použita v deseti milionech instalací. Jeden z důvodů, proč se používá podobně jako například jazyk PHP je ten, že jeho použití na serverech je zdarma. Ve spojení s vysokým výkonem a jednoduchostí instalace na servery bez velkých hardwarových nároků z ní dělá vhodný databázový systém pro většinu datově méně náročných webových projektů. Základem databáze je strukturované řazení dat, které jsou uloženy v počítačovém systému a uspořádány tak, aby získání informací bylo možné určitým způsobem dohledat. Zkratka SQL v MySQL znamená v angličtině „Structured Query Language“ a veškeré schéma je v tomto systému dokumentováno v angličtině.

¹⁸ NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7

¹⁹ Tamtéž

Ukázka jednoduchého dotazu:

```
SELECT title FROM publication WHERE author = 'Karel Čapek';
```

MySQL databáze může obsahovat více tabulek, kdy každá obsahuje určitý počet řádků. Relace je zastoupena jednotlivými vztahy mezi tabulkami, kdy každá tabulka obsahuje sloupce a řádky a v jednotlivých polích jsou pak uchovávána data. V dotazu, který je zobrazen výše, se zobrazuje výpis dat z určité databáze, která zobrazí sloupec title (titulek publikace) z tabulky publication (publikace), jejímž autorem je Karel Čapek.²⁰

2.1.2 Programovací jazyk PHP

Jazyk PHP vznikl v roce 1996 a od té doby se neustále vyvíjí. Tento skriptovací jazyk je využíván především na serverech, kde se generuje dynamický výstup. Tedy o výstup, který je potenciálně jiný vždy, když prohlížeč dotazuje danou webovou stránku. Hodně developerů vyvíjejících aplikace spojuje PHP kód spolu s HTML, CSS, JavaScriptem a databází a vytváří tak komplexní projekt, který vyhovuje moderním standardům.²¹

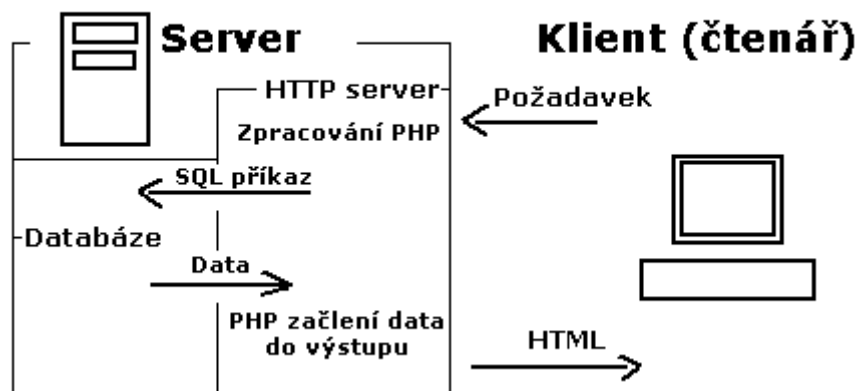
Nativně jsou PHP dokumenty ukončeny příponou .php. Když webový server rozpozná danou příponu požadovaného souboru, tak ho přidělí PHP procesoru. Pochopitelně webové servery jsou vysoce konfigurovatelné, tudíž je možné soubory končící .htm nebo .html přiřadit také PHP procesoru, který do nich dosadí výstupy, které jsou požadované.²²

²⁰ NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7

²¹ Tamtéž

²² Tamtéž

Schéma spolupráce PHP s databází



Obr. 3: Schéma spolupráce PHP s databází ²³

Celý program v PHP by měl být schopný vrátit čistý soubor vhodný pro zobrazení ve webovém prohlížeči. Nejjednodušším výstupem PHP dokumentu by měl být HTML. PHP patří k nejrozšířenějším skriptovacím jazykům pro webové účely a v květnu 2017 byl používán na více než 82 % všech webových serverů. Nejčastěji se používá v kombinaci s operačním systémem Linux a s schématickými (relačními) i neschématickými databázovými systémy. ²⁴

Ukázka fungování PHP kódu:

```
<?php
    echo "Hello world";
?>
```

Pro spuštění PHP scriptu je potřeba celý kód nejprve obalit do „<?php ?>“ tagů, následně pak server vykoná daný příkaz. Kód, který je zobrazen výše, vykoná zápis textu „Hello world“ do HTML dokumentu. ²⁵

Jazyk PHP dokáže pomocí vestavěných nebo vlastních funkcí vyvolat operace, které následně zašlou požadavek na databázový systém, je tak možné data uložit, upravit, smazat. Vše se odehrává na webovém serveru a ten poté zašle výsledek. Jak již bylo zmíněno více, oproti JavaScriptu a HTML, však PHP kód v prohlížeči není viditelný, jelikož interpreter PHP kód překládá do HTML, který ve výsledku zašle prohlížeči. Většinou se PHP užívá při tvorbě různých grafů, diskuzních fór,

²³ Jak Psát Web: Možnosti PHP [online]. [cit. 2019-04-16]. Dostupné z: <https://www.jakpsatweb.cz/php/moznosti-php.html>

²⁴ NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7

²⁵Tamtéž

počítadel či anket, které jsou propojeny s relačním databázovým systémem MySQL.²⁶

2.2 Frontend

K označení viditelní části webu, která je určená pro komunikaci mezi uživatelem a webovou aplikací, slouží pojem frontend. Jedná se o část webu, která slouží jako grafická nadstavba, která získává data z backendové části aplikace. Pro tvorbu frontendu se využívá programovací jazyka JavaScript, kaskádové styly CSS a jazyk HTML, který je základním komunikačním jazykem v rámci internetového prohlížeče. Příkladem frontendu může být například digitální nástěnka, kalendář, fotogalerie, grafická navigace či kontaktní formulář v rámci univerzitní webové stránky.²⁷

2.2.1 HTML a CSS

Zkratka HTML je z anglického názvu HyperText Markup Language. Jedná se o jazyk, který je dnes využíván na drtivé většině webových stránek, kdy společně v kombinaci s CSS, který je popsán níže, vytváří ideální definici, vzhled a význam jednotlivých webových stránek²⁸

Jazyk HTML vznikl v devadesátých letech dvacátého století. V počátku tento jazyk popisoval pouze základní strukturu stránky, jako byly seznamy či tabulky a elementy, které vytvářejí jednoduché webové stránky. V současné době se nachází pod názvem HTML5 a rozšířil se o velké možnosti interakce mezi prohlížečem a uživatelem. Součástí toho je například integrovaný přehrávač videa, podpora nových formátů obrázků a videí či různé grafické prvky.²⁹

Zkratka CSS, z anglického názvu Cascading Style Sheets, je označení pro kaskádové styly, které se používají v souvislosti s jazykem HTML pro tvorbu webových stránek a aplikací. Oproti HTML, který definuje pouze strukturu a obsah stránky, tak CSS se věnuje především podobě a vzhledu, jak bude uživatel

²⁶ NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7.

²⁷ CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. ISBN isbn-978-80-251-3733-8.

²⁸ Tamtéž

²⁹ Tamtéž

přístupovat k obsahu webové stránky. Zdrojový kód je napsán v textových souborech a stejně tak, jako je tomu u HTML kódu, je jednoduchá jeho editace.³⁰ První verze kaskádových stylů vznikla v roce 1996, která se jmenovala CSS1. V této době tuto verzi podporuje pouze prohlížeč Internet Explorer 3, jehož podpora však byla pouze ve formátování barvy a písma. Postupem času si však získala podporu i u prohlížeče Netscape a s příchodem Internet Explorer 5 si vydobyla mnohem lepší podporu, než tomu bylo dříve. Od roku 2000 vzniká druhá verze kaskádových stylů, pojmenovaná jako CSS2, která obsahuje nové vlastnosti a podporu formátování. Jednalo se například o stylování několika dalšími parametry, jako je *cursor*, *outline* či *visibility*. Dále to byla vylepšená práce s fonty a podpora médií se stránkovým vstupem. U prohlížečů různých výrobců však tato verze nikdy nebyla stoprocentně podporována, každý prohlížeč podporoval pouze část jejich funkcí. Největší podporu se dočkala u prohlížeče Mozilla Firefox, která se snažila dodržovat standardy, ale stejně plného pokrytí se nedočkala. Během roku 2011 se verze CSS2 dočkala aktualizace na verzi CSS 2.1, která opravovala některé důležité chyby a přidala definici výšky a šířky absolutně umístěných prvků. Aktuálně nejnovějším standardem je CSS3, který je plně kompatibilní s minulými verzemi. Tato nová verze kaskádových stylů nabízí nové selektory, možnost vytvářet 2D a 3D transformace objektů, zakulatit rohy elementů, použít vržený stín či vytvořit animace. Jako příklad nových atributů může posloužit `element[foo*="bar"]`, který vybírá všechny třídy, které obsahují slovo bar. Atribut `:nth-last-child(n)` označuje poslední element, který se nachází v dané sekci a `:enabled` vybere element, který je momentálně aktivní.³¹

Ukázka nových atributů podporovaných od verze CSS3:

```
element[foo*="bar"]
:nth-last-child(n)
:enabled
```

Mezi nejznámější CSS knihovny, které se užívají na webových stránkách patří Bootstrap. Tato knihovna je nejpoužívanější CSS knihovnou na internetových

³⁰ CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. ISBN isbn-978-80-251-3733-8.

³¹ BRIAN, P. Hogan. HTML5 a CSS3, Výukový kurz webového vývojáře. Brno: Computer Přes, 2011. ISBN 978-80-251-3576-1

stránkách, která zdarma poskytuje HTML, JavaScript a CSS řešení. Důvodem, proč vývojáři používají hojně Bootstrap na svých projektech je beze sporu urychlení vývoje samotných stránek či aplikace. V tom případě tedy není potřeba začínat vytvářet styly od začátku, ale postačí použít tuto knihovnu. Další známou knihovnou je například Foundation či Font Awesome.³²

2.2.2 JavaScript

JavaScript je programovací jazyk, který byl představen jako součást Webu 1.0, což byla doba, kdy bylo součástí prohlížečů běžné rozšíření jako Java, Javascript, JScript či ActiveX. Poté se obliba a vývoj JavaScriptu do dnešní doby prudce zvedla a je nepostradatelnou součástí většiny webových stránek a aplikací. JavaScript byl vytvořen, aby mohl upravovat přístup ke všem elementům v rámci HTML dokumentu. Jinými slovy, poskytuje dynamickou uživatelskou interakci jako je například validace emailové adresy v rámci webové stránky. Často se jedná o upozornění o neúplnosti údajů či nevyplněném poli.³³

V kombinaci s CSS je JavaScript jakousi formou, jak zobrazovat vizuální změny, které se provádí na stránce dynamicky. Pomocí JavaScriptu lze vytvořit velké možnosti, jak odlišit chování webové stránky od ostatních. Vývojářské firmy, které vyvíjí internetové prohlížeče, se snaží přidávat nové vlastnosti JavaScriptů do svých nových i starších verzí prohlížečů a podporují tak zpětnou kompatibilitu funkcí, byť není stoprocentní. Vývojáři Javascriptu se snaží přijít s novými myšlenkami a chtějí vzájemně zaručit kompatibilitu mezi jednotlivými prohlížeči. Stále ale existuje velké množství knihoven a jednotlivých scriptů, které nejsou podporovány v různých verzích prohlížečů. Obecně je vždy při vývoji nutné dodržovat určité standardy, které ovšem lze automaticky optimalizovat pomocí přídatných modulů, které se starají o kompatibilitu kódu.³⁴

Jak již bylo uvedeno, JavaScript byl původně vyvinut pro poskytnutí dynamické kontroly nad jednotlivými prvky v HTML dokumentu a toto je jeho poslání i do

³² ČÁPKA, David. 1. díl - Úvod do CSS frameworku Bootstrap. Itnetwork.cz [online]. itnetwork.cz [cit. 2018-04-30]. Dostupné z: <https://www.itnetwork.cz/htmlcss/bootstrap/uvod-do-css-frameworku-bootstrap>

³³ NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7

³⁴ Tamtéž

dnešních dnů. Ale více je v dnešní době využíván JavaScript jako asynchronní komunikace, kdy se všechny procesy odehrávají na pozadí webového serveru. Jinými slovy asynchronní komunikace pro webové stránky znamená, že samostatné programy, které musí být celé znovu načteny celé znovu, mohou být načítány po jednotlivých komponentách, které se načítají každá samostatně a v případě výpadku jedné komponenty se načte webová stránka místo celkové chyby. Jako příklad lze uvést část sociální webové stránky, která se stará o zobrazení náhledového obrázku profilu uživatele, který se načítá samostatně bez návaznosti na okolní části webové stránky.³⁵

Ukázka základního JavaScript kódu, která funguje napříč většinou prohlížečů:

```
<script type="text/javascript">
  document.write("Today is " + Date() );
</script>36
```

Vše, co je obsažené v tagu `<script type="text/javascript">` říká prohlížeči, že se jedná o javascriptový kód, který je spouštěn přímo v HTML kódu stránky. Kód, který se nachází v odstavci výše prohlížeči, říká, aby vykonal příkaz, který zapíše text „Today“ do aktuálního těla dokumentu. Společně s textem „Today“ se vykoná i funkce, která vypíše aktuální datum, které je nastavené v rámci serveru, na kterém se daný script spouští. Po spuštění scriptu se na webové stránce objeví následující:

```
„Today is Sun Jan 01 2017 01:23:45“
```

Součástí JavaScriptu je také Ajax, který byl vytvořen roku 2005. Je postaven na asynchronním chování JavaScriptu a formátu XML, který se stará o přenos dat mezi prohlížečem a serverem na pozadí. Ajax slouží ke komunikaci mezi PHP, HTML a JavaScriptem, který předává data mezi jednotlivými webovými službami. Pomocí Ajaxu jsou vytvářeny požadavky, které se volají pomocí akce, kterou vytváří JavaScript.³⁷

Velice oblíbenou knihovnou, která se používá ve velkém množství webových stránek, je bezesporu jQuery. Jedná se o výkonný a upravitelný JavaScriptový kód, který umožňuje poskytnout další funkcionality, které nejsou součástí nativního

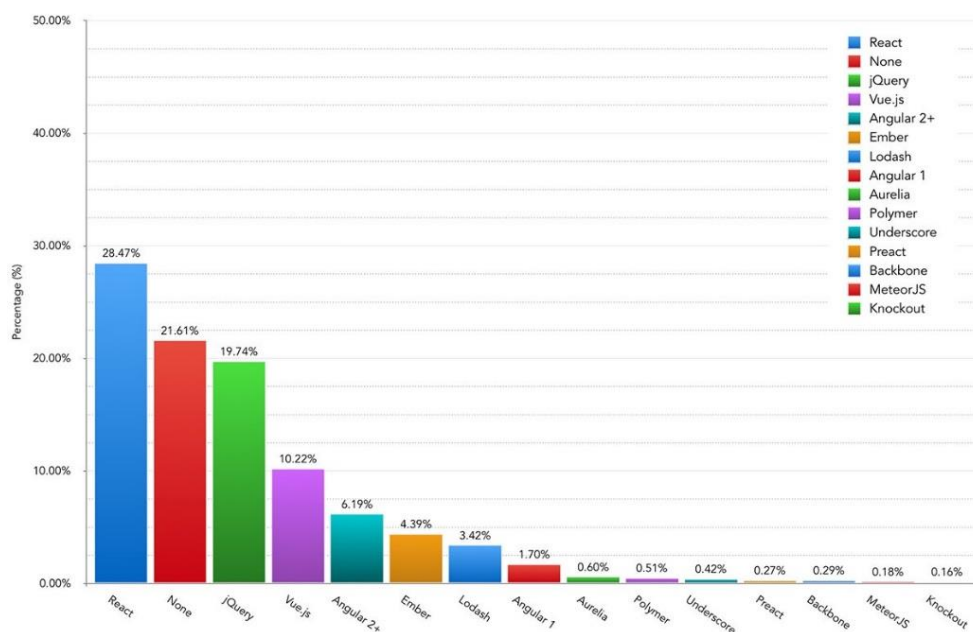
³⁵ NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7

³⁶ Tamtéž

³⁷ Tamtéž

javascriptového kódu. Součástí jQuery je CSS, které se zde používá pro animace, spuštění určitých akcí či asynchronní komunikace. S pomocí této knihovny se není potřeba zabírat kompatibilitou pro většinu běžných prohlížečů, ale také umožňuje jednoduchý přístup k HTML a DOM manipulaci, speciálním funkcím umožňující úpravu CSS či pro vytvoření profesionálních efektů a animací.³⁸

Mezi moderní trendy patří v dnešní době použití JavaScriptových frameworků, které se starají o samotné sestavení celé stránky.



Obr. 4: Graf použití JavaScriptových frameworků na webových stránkách³⁹

Jedním z nich je React, který je zaměřen na tvorbu uživatelských rozhraní. Původně byl vyvinut společností Facebook v roce 2013 a v dnešní době se těší velké oblibě mezi vývojáři převážně webových aplikací. Samotný React využívá syntaxe JSX, která překonvertuje daný kód do Javascriptu, který vykreslí celou stránku formou HTML kódu. Mezi další nejoblíbenější frameworky patří Vue.js, Angular, Polymer či Ember.⁴⁰, jak lze vidět na Obr. 4.

³⁸ NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7

³⁹ NOLAN, Ashley Watson. Launching the Front-End Tooling Survey 2019 [online]. 8. duben 2019 [cit. 2019-04-14]. Dostupné z: <https://medium.freecodecamp.org/launching-the-front-end-tooling-survey-2019-4cb2b72f0b42>

⁴⁰ BANKS, Alex a Eve PORCELLO. Learning React. 2017. ISBN 978-1-491-95462-1

3. Archivnictví

3.1 Pojem a definice oboru

Původní termín „archiv“ pochází z řečtiny a označoval úřední či vládní budovu. V dnešní době je archiv chápán jako instituce, která má za úkol uchovávat a chránit historické prameny, většinou písemné formy, které vznikaly z činnosti úřadů, právnických a fyzických osob. Všechny tyto písemnosti se nazývají v archivní terminologii jako „archiválie“. Veškeré archivy v České republice jsou pod správou ministerstva vnitra, v zahraničí mohou spadat pod ministerstvo kultury. Mezi základní funkce archivu patří: ^{41 42}

- evidence písemností (archiválií);
- sběr a selekce archiválií, které se dostanou do archivu a následně jejich předarchivní péče;
- zajištění bezpečnosti pro uložené archiválie a ochrana proti fyzickému poškození;
- zpřístupňování archiválií badatelům.

Archiv má několik významů, základní definice hovoří takto:

- Je to instituce, ve které se nachází velké množství uložených archiválií z určité sběrné oblasti, kterou spravuje daný archiv. Návštěvníci neboli badatelé mohou zde uložené archiválie shlédnout a následně je studovat.
- Představuje místo, kde se shromažďují určité písemnosti z činností daných institucí, jako mohou být úřady či firmy. Mohou se zde však také uchovávat záznamy z činností jednotlivce. Veškeré záznamy jsou upraveny výběrem a uloženy za účelem dokumentárním a správním. Celý tento soubor archiválií se nazývá archivní fond.
- Jedná se o depozitář, kde jsou uloženy veškeré archiválie z dané oblasti. ⁴³

⁴¹ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 7

⁴²] Co je archiv. Cesarch.cz [online]. Česká archivní společnost [cit. 2019-03-29]. Dostupné z: <http://cesarch.cz/archivnictvi/co-je-archiv/>

⁴³ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 7-8

Archivnictví je disciplína, která pojednává o všech věcech, které jsou spojeny nějakým způsobem s archivy a archiváliemi. Zaměstnanci, kteří pracují v archivu se nazývají archiváři a jsou odpovědní za správnou evidenci a zacházení s archiváliemi, které tam jsou uloženy. V rámci souboru archiválií je nutné přesně definovat odlišnosti, které se nachází mezi pojmy archivní sbírka a archivní fond. **Archivní sbírka** je soubor archiválií, které jsou navzájem propojeny společnými znaky, které mohou být v rámci činnosti, kterou archiválie obsahují či se shodují v rámci sběru z určité organizace. **Archivní fond** pak představuje určitý soubor archiválií, které jsou sjednoceny podle původce, kterým může být organizace.

Z historického hlediska se dříve písemnosti a archeologické nálezy umísťovaly do muzeí či knihoven. V dnešní době je však funkce těchto institucí přesně definována a každá se specializuje na určité písemnosti a předměty. Pro lepší orientaci, jsou následně uvedeny jednotlivé rozdíly mezi archivem, knihovnou a muzeem.⁴⁴

Archiv tedy uchovává písemnosti, které souvisejí s činností státních i veřejných podniků. Tyto podniky musí předávat písemnosti archivu, protože je to dáno zákonem a je to tedy pro ně nevyhnutelné. Do archivu se tak dostávají psané písemnosti, jako jsou úřední knihy či listiny, ale také zde jsou uleženy i písemnosti tištěné, do kterých patří patenty či úřední cirkuláře. Archiv se stará o samotný sběr těchto písemností a následně je i třídí a eviduje podle původce, kterým jsou dané úřady či společnosti. **Muzeum** shromažďuje, vystavuje a eviduje věci hmotné povahy, které jsou důležité z historického hlediska a mají přínos pro společnost. Muzeum tyto předměty získává sběrem, koupí či formou daru od majitele. Dříve se zde ukládaly i materiály archivní povahy, ale postupem času došlo k převedení těchto archiválií do archivů, ovšem stále můžeme nalézt i v dnešní době špatné zařazení některých materiálů. **Knihovna** je instituce, která shromažďuje literární rukopisy či sbírky tisků. Je pouze v kompetenci knihovny, které písemnosti chce u sebe uchovávat a nabízet je tak čtenářům. Nové rukopisy či knihy získává knihovna

⁴⁴ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 7-8

převážně nákupem či v rámci daru. Většina materiálů je obvykle kategorizována a tříděna podle témat.⁴⁵

3.1.1 Historie archivnictví

První zmínky o archivu jsou původně z Řecka, kdy v řečtině slovo „archein“ znamenalo určitou instituci nebo budovu instituce a v ní se shromažďovaly veškeré písemnosti. Poté byl tento termín převzat i do jiných jazyků a dnes můžeme termín chápat jako instituci, kde jsou umístěné archiválie z určité sběrné oblasti, ve které sídlí správa archivu. Návštěvníci, kteří zavítají do tohoto zařízení mohou pak archiválie spatřit a studovat.⁴⁶

Archivů je celá řada a lze je dělit podle toho, v kterém státu se určitá síť archivů vyskytuje. Mohou být různá hlediska, podle kterých se archiv rozděluje. Může se jednat o právnickou či fyzickou osobu, která v archivu uchovává své písemnosti, a dále se tyto písemnosti mohou dělit podle všeobecného rázu (většinou státní instituce) nebo se specializuje na určitý úsek. Členit archivy můžeme dále na státní, soukromé a veřejné. Ty se dále vymezují podle území, oboru činnosti či kompetence původce.⁴⁷

V historii archivnictví se nejprve cenné písemnosti uchovávaly na hliněných destičkách, papyru, pergamenu a později i papíru. Tyto písemnosti se uchovávaly kvůli důležitosti, která byla právního charakteru. Jednalo se například o uznání vlastnictví majitele či o zápisy z jednotlivých setkání. Takovéto písemnosti se dříve ukládaly do knihoven či do truhel, které se nacházely například u panovníků.⁴⁸

Celkově jsou počátky archivnictví datovány kolem roku 3000 před naším letopočtem a v této době se jednalo o pouhé zabezpečení pro chod fungování společnosti. V dílech židovského kněze Josefa Flavie bylo možné nalézt způsoby

⁴⁵ ŠTOURACHOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 7-8

⁴⁶ BARTOŠ Josef – CHOBOU, Karel. Úvod do archivnictví pro historiky. Olomouc, 2000. 97 s. ISBN 80-244-0142-8.

⁴⁷ Tamtéž

⁴⁸ Tamtéž

uchování písemností, které byly v té době důležité. Úschovy se nacházely v knihovnách či chrámech.⁴⁹

Materiály, které se používaly úplně v prvopočátku, tak byly pálené hliněné destičky. S nástupem psacích látek kolem roku 600 před naším letopočtem se začaly písemnosti uchovávat na dřevě, na papyrus či na kůži. V této době se na tento materiál ukládaly hlavně zákony a smlouvy, které zajišťovaly určitá práva. První instituce, kterou můžeme nazvat podle dnešní charakteristiky archiv byl kolem roku 460 našeho letopočtu v Athénách. Tento archiv měl i ochránce v podobě 7 mužů. Původně byl archiv umístěn na radnici, ale tu roku 267 zničili barbaři. V této době je ovšem doloženo, že podobné instituce měla i blízká řecká města.

Další evoluce archivnictví pokračovala v Římě, kde se dochovaly písemnosti z roku 449, které byly v té době umístěny v Saturninově chrámu. Do tohoto archivu se ukládaly písemnosti jako protokoly senátu, výsledky voleb či seznamy sčítání lidu a další statistické listiny. Vznikl zde i další typ archivu, a to byly archivy čistě úřední.

3.1.2 Třídění archiválií

Za archiválie můžeme považovat všechny psané, zvukové i obrazné věci, které vznikly činností svých původců, jako jsou úřady, instituce, fyzické i právnické osoby či jednotlivé podniky. Povaha archiválií může být dle původců typu umělecké, politické, správní, soudní, literární, církevní, hospodářské či obchodní. Samotné archiválie mohou být děleny i podle toho, zdali jsou veřejné či soukromé povahy. Soukromé povahy to mohou být různé korespondence, osobní doklady, deníky, novinové výstřižky, rodinné fotografie a další. Důvodem, proč se tyto věci považují jako archiválie je fakt, že jsou zdrojem tehdejších událostí a jevů, které se v té době odehrávaly.⁵⁰

⁴⁹ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 7-8

⁵⁰ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 21-25

V dřívějších dobách bylo toto rozdělení vcelku jednoduché, ale od 16. století s větší byrokracií přibýlo i více rozčlenění písemností na jednotlivé sekce. Přibýla například možnost rozdělit písemnosti na všeobecné státní písemnosti, kdy tyto písemnosti byly politicko-policejní správy a k nim se připojovaly věci týkající se soudní či vojenské činnosti.

První dělení probíhalo na úrovni písemností nadřízených institucí. Hlavní část archiválií v té době tvořily dokumenty panovníků, kteří vydávali různá rozhodnutí, pokyny, svolení a oznámení. Jedna z kategorií, které se třídily do stejné sekce, tak byly dokumenty typu majestáty, diplomy, privilegia, potvrzení o šlechtickém rodě či různé tržní výsady. Dalším druhem byly mandáty, patenty či edikty, které zveřejňovaly různé řády, vyhlášky a zákony. Pokyny vydané panovníkem, které byly adresovány státním úřadům, se nazývaly reskripty. Tyto dokumenty měly také vlastní umístění v tehdejších archivech. V neposlední řadě se třídily i kabinetní listy či vlastnoruční podpisy, které měly osobní charakter a dopis měl přímé oslovení adresáta. Těmito dopisy si panovník dopisoval přímo s jednotlivými nejvyššími úředníky a tyto písemnosti se většinou ukládaly do soukromých archivů. Dvorské dekry, což byly dokumenty v neosobní formě, kterými panovník dával určitému adresátovi na vědomí své rozhodnutí, byly také skladovány v archivu zvlášť.⁵¹

Další skupina, do které se třídily jednotlivé listiny, byly písemnosti vydávané vyššími úřady z vlastní iniciativy nebo v zastoupení panovníka. Nařízení jednotlivých úřadů, které byly formulovány neosobně a rozhodovaly o tom, jestli reskripty a dekry vyhovovaly těmto úřadům a příslušným orgánům.

Písemnosti úřadů stejně postavených se také třídily zvlášť, což byl většinou text, který neobsahoval formální text, ale měl slavnostní a přesnou formu. Jednalo se o přímou korespondenci mezi panovníky, která byla sestavena z jednoduchého oslovení, přímého textu, podpisu a data zhotovení dané listiny.⁵²

Do poslední skupiny, která se rozdělovala mezi archiváliemi, tak byly písemnosti podřízených úřadů a přípisy poddaných. V této skupině byly písemnosti

⁵¹ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z:

<https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 21-25

⁵² Tamtéž

s neutrálním charakterem, jako byly například zprávy, relace, žádosti, přednesení, stížnosti, lustra, extrakty, recesy, protokoly, inventáře a další písemnosti, které měly ponížené formulování.

S příchodem 18. století se písemnosti ve státní správě dělily na presidiální, tajná a všeobecná, kdy se snižuje počet panovníkem vydaných písemností a zvyšuje se počet správních a normativních pokynů, zákonů a vládních nařízení.

Od poloviny 19. století se dalším typem veřejných písemností stávají písemnosti městské, církevní či stavovské. Poté v období průmyslové revoluce se k tomu přidávají i písemnosti hospodářského typu, tedy vzniklé z obchodní, živnostenské či zemědělské činnosti, které definovala v té době nauka o hospodářských aktech. Celkově se tedy dá jako zlomové období určit okolo 18. století, kdy do úschovy archiválií zasahuje velký rozvoj administrativy a poté v 19. století s příchodem kapitalismu narůstá i množství účetních a hospodářských písemností, které je nutné uchovávat.⁵³

⁵³ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>, str. 21-25

4. Spisová služba

Spisová služba je soubor činností, technických a věcných prostředků vedoucích ke správě dokumentů. Přičemž dokumenty mohou být jak analogové (spisová služba v listinné podobě), tak digitální či elektronické (elektronický systém spisové služby), případně i předmětové.

4.1 Vývoj spisové služby

Spisová služba prošla ve svém vývoji několika periodami:

Období kancelářů

Kancelář byl původně nazýván celý úřad, v průběhu vývoje se tak nazývala jen ta část, která zajišťovala přijetí, vyřízení a vypravení písemností. Po čase se od něj oddělil i sklad vyřízených spisů, a nakonec i archív. Jelikož písemná agenda ve středověkých kancelářích obsahovala ještě velmi málo písemností, tak kanceláře té doby se zabývaly i jinými veřejnosprávními záležitostmi.

Na našem území vznikla ve 12.století Česká královská kancelář řadící se mezi naše nejstarší úřady. V čele této kanceláře byl nejvyšší kancléř Království českého. Od roku 1225 jím byl vždy vyšehradský probošt, který se vesměs rekrutoval ze stavovské obce. Tento nejvyšší kancléř disponoval velkou pečeti království a vydával královské písemnosti. Zároveň fungoval jako garant proti panovníkově svévoli. Během svého trvání získala kancelář další pravomoci včetně soudní. V roce 1607 došlo k přejmenování České královské kanceláře na Českou dvorskou kancelář a v roce 1624 byla přemístěna do Vídně, kde fungovala jako nejvyšší správní a soudní instalace vlády Habsburků nad zeměmi koruny české. V roce 1749 byla Česká dvorská kancelář zrušena a nahrazena direktoriem. Direktorium byl ústřední orgán veřejných a finančních záležitostí, který fungoval pro české i rakouské země a byl zaměřen na politickou a finanční správu. Zrušení české dvorské kanceláře a vznik Direktorie je považováno za konec právní samostatnosti českého státu.⁵⁴

Po roce 1526 s nástupem Habsburků na český trůn došlo k reorganizaci úřadů, začíná byrokracie a v tomto období vzniká i spisová služba.

⁵⁴ Česká kancelář dvorská 1599–1608, Praha 1931.

Období aktových registratur

Datuje se od 16.století s nástupem Habsburků na český trůn, kdy začaly velké změny, které se dotkly i spisové služby. Důsledkem vzniku nových zeměpanských úřadů (česká a dvorská kancelář, komory, apelační soud) je narůstání aktového materiálu. Následkem toho je začátek tradice byrokratického správního aparátu. Výsledkem pak je nárůst počtu pracovníků, neboť dochází k rozdělení práce v kanceláři na několik speciálně na sebe navazujících úkonů, zvyšují se nároky na dokumentaci oběhu spisů, zavádí se přesnější evidence o uložení spisů a vznikají tzv.aktové registratury. Vlastní koloběh spisů probíhal přijmutím u kancléře, který ho projednal se svým sekretářem, došlé spisy evidoval registrátor ve svých knihách, písemný návrh vyřízení vypracoval koncipista, schválil sekretář, načisto opsal ingrosátor, čistopis zpoplatnil taxátor a odeslal expediktor.

Období osvícenské

Další velké změny přišly v 18.století za panovníků Marie Terezie a Josefa II. V tomto období dochází k dalšímu zkvalitnění spisové služby. Z důvodu narůstající agendy nastává rušení kolegiálního systému rozhodování. Dále dochází k osamostatňování registratur, kam se ukládaly neužívané písemnosti, a také spisy ze zrušených klášterů a úřadů. Důsledkem byla zvyšující se byrokracie a vytvoření hierarchie pracujících tříd. V roce 1706 byly vydány první instrukce pro spisovou službu pro krajské úřady. K zásadnímu zvýšení spisové služby došlo až v roce 1746, kdy bylo Marií Terezií vydáno nařízení o přijatých a vyřízených spisech, kterým se měnilo dobrovolné vedení protokolů na povinné. Pro zemské úřady, vlády a pro gubernia byly vydány nové manipulační řády. Za Josefa II. vznikl evidenční systém spisové služby založený na podacím protokolu a číslu jednacím, což mělo význam pro archivaci spisů. Byly zrušeny evidenční pomůcky, zůstal jen všeobecný podací protokol. Přes veškeré značné množství změn a úprav právního prostředí se v tomto období nepodařilo prosadit zásadu jednotné spisové služby ve všech institucích v habsburských zemích.⁵⁵

⁵⁵ HLEDÍKOVÁ-Zdeňka, JANÁK-Jan, DOBEŠ-Jan. Dějiny správy v českých zemích: od počátku státu po současnost. Praha: Lidové noviny, 2005, s. 570. ISBN: 978-80-7106-906-5.

Rok 1848

Revoluční rok 1848 znamenal pro habsburskou monarchii nejen vnitřní, ale i zahraničněpolitické ohrožení. Rakouská diplomacie byla v tomto období určována třemi směry, které se navzájem prolínaly s tzv. německou otázkou, bojem proti uherské revoluci a rakouskými aktivitami na Apeninském poloostrově. Toto revoluční období se stalo mezníkem na několik následujících desetiletí ve spisové službě. Vznikly nové politické, soudní a finanční úřady s rozsáhlou legendou a v této době vznikala ministerstva a oddělení politické a soudní správy první instalace. Dále došlo k sjednocení v oblasti justiční správy. V roce 1897 vznikl nový jednací řád, který byl ve spisové službě velkým pokrokem vpřed. Tento jednací řád omezil roli podacího rejstříku, zavedl spisovou značku a ústní líčení.

Vznik ČSR

Stát Čechů a Slováků převzal fungující habsburský systém spisové služby. Později ho upravoval k vyšší efektivitě a k specifikům nově vzniklého státu. V roce 1928 vznikl Kancelářský Řád, který jen upravil předpisy z období Rakousko-Uherského a jeho úprava z roku 1935 byla jen novelou předchozího. Zůstala v něm zachována ústřednost podacích deníků u okresních úřadů a vedení deníků zemských úřadů a policejních ředitelství podle skupin. Nově v něm bylo zavedení speciálního deníku.

Období německé okupace

Byl zaveden a oblíben desetinný registraturní plán.

Rok 1945

Po skončení II.světové války nastává návrat k systému první republiky. Ten byl narušen převzetím moci ve státě komunistickou stranou v roce 1948. V roce 1950 byly vydány státní normy a spisový a archivní řád. Důsledkem vznikl velký nedostatek vyškolených pracovníků. Zásadní změny měl přinést nový spisový plán k 1.1.1949, který vydržel do roku 1951, následoval pak další do roku 1954. Kvůli

značné pestrosti evidenčních a ukládacích systémů spisová služba nebyla v ústředních úřadech jednotná.⁵⁶

4.2 Definice spisovny

Co je spisovna? Zákon o spisové a archivní službě⁵⁷ jí definuje jako místo určené k uložení, vyhledání a předkládání dokumentů pro potřebu původce a k provádění skartačního řízení. V příručním slovníku naučném je zase spisovna označena jako „*Oddělení úřadu (podniku, ústavu) sloužící k ukládání a evidenci vyřízených spisů*“.⁵⁸ Zajímavou definici však najdeme v *Ottově slovníku naučném*. *Spisovnou je označena „sbírka spisů některého ústavu neb úřadu, pak i místnost, kde se spisy ty ukládají. Při velikých úřadech bývá v registratuře (spisovně) zaměstnána celá řada úředníků. V jejich čele bývá registrátor správce spisovny. Někdy se však registrátorem nazývá úředník, který registruje, t. j. zapisuje došlé spisy neb ústní sdělení do jistých rejstříků (knih, protokolů). Registrátorovi se v tomto smyslu se říká registrant. Úředníci registratury náleží k úředníkům manipulačním.*“⁵⁹

Zde je zřetelně vidět rozdíl mezi archivem, který je podle stejného zákona definován jako zařízení, které slouží k ukládání archiválií a péči o ně. A v *Ottově slovníku naučném* je archiv popsán takto: „*Archiv jest sbírka písemných, případně tištěných památek rázů úředního, jež jsou důležité pro dějinný nebo právní vývoj buď jednoho, buď více předmětů, mající v přítomnosti či budoucnosti posloužiti vědě, správě veřejné, nebo soukromým zájmům za doklad. Dále nazýváme archivem místo, kde chovají se památky toho druhu – archiválie.*“⁶⁰ Vzhledem k tomu, že na sebe vzájemně tyto instituce navazují, musejí spolu blízce spolupracovat. Spisovny se musejí řídit radami a doporučeními od archivů a archivy nad nimi mají dohled.

⁵⁶ ŠTOURAČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z:

<https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

⁵⁷ *Příruční slovník naučný III. díl*. Praha: Nakladatelství Československé akademie věd, 1966, s. 830.

⁵⁸ *Tamtéž*

⁵⁹ *Tamtéž*

⁶⁰ *Tamtéž*

Základním rozdílem tedy je, v jaké fázi života se se zde dokumenty vyskytují. Ve spisovně jsou uzavřené spisy, nebo momentálně nepotřebné dokumenty, které by mohl ještě původce k něčemu potřebovat. Takto čekají do uplynutí skartační lhůty, aby byly podrobeny skartačnímu řízení. V archivu jsou již jen vybrané dokumenty, které prošli skartačním řízením a ze kterých se automaticky po převzetí stávají archiválie. Archiválie jsou poté uloženy a pracovníci archivu se o ně starají, aby vydrželi pro další generace badatelů.⁶¹

Dalším velkým rozdílem je přístup k uloženým materiálům. Ve spisovně mají přístup jen zdejší pracovníci a určití zaměstnanci původce. Ostatní si musejí sjednat povolení k nahlédnutí u adekvátní osoby. V archivu může do archiválií nahlédnout kdokoli, kdo splní určité předpoklady, vyjmenované v zákoně. V případě některých archiválií je nutno počkat, dokud neuběhne určitá doba, aby se mohly poskytnout veřejnosti například z důvodu utajení či vypršení autorských práv.⁶²

V mnoha věcech jsou si, ale spisovna a archiv podobné. Například v obou zařízeních musí být zaměstnanci se správným zaměřením a tím pádem schopnostmi se správně starat o uložené materiály. Takovíto zaměstnanci musí mít patřičné vzdělání a pokud možno přímo vysokoškolský titul z oboru spisové služby. Taktéž pro zde uložené materiály je nutno dodržovat podobné podmínky. To z důvodu, aby se již během předarchivní péče nepoškozovali uložené dokumenty. Spisová služba má tedy kritický význam v péči o budoucí archiválie. Bez ní by nebylo moc co archivovat.⁶³

Spisovou službou se zabývá hlavně zákon o Spisové a archivní službě. Dále také platné vyhlášky č. 645/2004 Sb., ustanovení zákona o archivnictví a spisové službě v platném znění a č. 259/2012 Sb., o podrobnostech výkonu spisové služby.

⁶¹ *Příruční slovník naučný III. díl.* Praha: Nakladatelství Československé akademie věd, 1966, s. 830.

⁶² ŠTOURÁČOVÁ, Jiřina. *Archivnictví*. Brno: Masarykova univerzita, 2013. [ISBN 978-80-210-6512-3.](https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf) Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

⁶³ Tamtéž

4.3 Komerční spisovna

Komerční spisovny jsou specifickým druhem spisoven, hlavní rozdíl od klasických spisoven je, že ve svých prostorech ukládají dokumenty jiných původců za úplatu, a to z několika důvodů. Jedním z důvodů může být zánik původce, a tudíž potřeba někde uložit dokumenty, kterým ještě nevypršela skartační lhůta. Dalším důvodem může být neschopnost původce zajistit adekvátní podmínky pro spisovou službu u sebe, a tak nechá komerční spisovnu, aby se starala o jeho dokumenty. Takováto spisovna musí splňovat klasické podmínky a navrch získat koncesi od živnostenského úřadu spolu s vyjádřením státního oblastního archivu ke splnění oněch podmínek. Oblastní archiv poté vede evidenci existujících komerčních spisoven.⁶⁴

Pokud chce podnikatel vést koncesovanou živnost „Vedení spisovny“, musí být naplněny tyto kroky:

- Musí požádat o udělení koncese na příslušný živnostenský úřad v místě, kde chce komerční spisovnu vést. K žádosti musí přiložit i doklady prokazující úplné střední nebo úplné střední odborné vzdělání. Dále pak tři roky praxe v administrativě. K provozování činnosti pak potřebuje schválení provozovny a bezúhonnost všech zaměstnanců.
- Živnostenský úřad předá v rámci žádosti k posouzení stavebně technický a bezpečnostní stav provozovny k posouzení příslušnému státnímu oblastnímu archivu.
- Státní oblastní archiv si poté dohodne termín, ve kterém archivní inspektoři provedou šetření v provozovně. Během šetření se musí doložit, že je vše ve spisovně v souladu s § 61 odst. 2 a 4 archivního zákona. To dokáže předložením všech příslušných dokumentů jako třeba snímkem příslušné části územního plánu obce, nebo výpisem ze stavební dokumentace.
- Po provedení šetření státní oblastní archiv vydá stanovisko ke zjištěným skutečnostem. Na celou tuto operaci má od doručení žádosti

⁶⁴ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

živnostenskému úřadu třicet dnů. Živnostenský úřad je nucen se stanoviskem archivu řídit.

Zájemci o udělení koncese „Vedení spisovny“ mohou požádat pracovníky archivu o poskytnutí metodické pomoci v dané záležitosti.⁶⁵

4.4 Skartace

Skartace je činnost, při které se vyřazuje ze spisovny nebo jednotlivých registratur písemný a ostatní materiál trvalé hodnoty od písemností a ostatního materiálu, který ztratil význam z hlediska úřední potřeby, politického, hospodářského nebo vědeckého. Vyřazené dokumenty mohou být určeny ke zničení (tato skupina se nazývá skart, vlastní fyzické zničení se neprovádí v rámci skartace), nebo k dlouhodobému uložení v archivu.⁶⁶

Začátky řízené skartace lze najít v 16.stol. kdy vznikla potřeba uvolnit místo v přeplněných registraturách zaplněnými nepoužívanými spisy. To se dělo vesměs velmi zmateně bez návaznosti na časovou souvislost. Výsledkem byly problémy s fungováním institucí a s jejich dokumentací. Největší typ této skartace v českém prostředí se uvádí nařízení Fridricha Viléma Haugwitze, který nařídil roku 1749 po zrušení české dvorské kanceláře spálit většinu písemných dokumentů. Tímto chaosem vznikla potřeba vnést do likvidace nepotřebných tiskovin systém. Nejstaršími nařízeními v českých zemích o vedení archivů byly císařský reskript z 31. října 1753 nařizující městským písařům vést v pořádku městské registratury a archivy a nařízení českého zemského gubernia ze 6. ledna 1805 o soupisu listin do roku 1700 v městských archivech.

V roce 1856 oběžník pražského místodržitelství vydal nařízení v němž se ukládala povinnost péče o archivy a úřadům byl nařízen archivní dozor. Vznikem

⁶⁵ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

⁶⁶ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

Československa přešlo archivnictví pod Ministerstvo školství a národní osvěty, ale připravit nový zákon o archivnictví se nepodařilo nikdy prosadit.

Po druhé světové válce patřilo archivnictví pod ministerstvo vnitra, kde byla v roce 1953 ustavena Státní archivní komise, která připravila legislativní podklady. Od 1. 11. 1953 byla zřízena. V roce 1954 bylo jako první legislativní úprava v této oblasti přijato vládní nařízení č. 29/1954 Sb. o archivnictví.

V roce 1968 došlo k federativnímu uspořádání Československa, vznikla národní ministerstva, a v roce 1974 schválila Česká národní rada na návrh ministerstva vnitra České socialistické republiky zákon o archivnictví č. 97/1974 Sb. ze 17. října 1974. Dle tohoto zákona vznikly Státní ústřední archiv, sedm oblastních archivů a 76 okresních archivů.⁶⁷

V roce 1989 došlo ke společenským změnám, byla vydána novela jako zákon č. 343/1992 Sb. z 29. dubna 1992 s platností od 1. září 1992. Vzhledem k dalším změnám bylo třeba v novém prostředí usadit spisovou službu na úrovni zákonné normy byl přijat zákon o archivnictví a spisové službě, přijatý parlamentem ČR dne 30. června 2004 s platností od 1. ledna 2005. Tento nový zákon potvrdil zachování archivnictví a spisové služby v pravomoci ministerstva vnitra. Původce vydává vnitřním předpisem spisový řád. Ten stanovuje základní pravidla pro výkon spisové služby. Cílem je jednoznačně definovat výkon spisové služby. Určuje tedy životní cyklus dokumentů. Je v něm stanoveno, jací pracovníci mají právo a povinnosti pracovat s dokumenty. Důležitou součástí spisového řádu je spisový a skartační plán. Zde je seznam dokumentů a jejich rozdělení do skupin se spisovými a skartačními znaky a lhůtami, kdy musí projít skartačním řízením.⁶⁸

U každého druhu dokumentu je uveden název a spisový znak (spisový plán) a zároveň se rovnou k dokumentu uvede skartační znak a skartační lhůta (tj. skartační režim, tato část normy je pak skartační plán). Skartační lhůta je dobou, po kterou

⁶⁷ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z:

<https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

⁶⁸ ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z:

<https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

má uzavřený spis u původce zůstat, než bude zařazen do skartačního řízení a skartační znak předjímá, co se spisem ve skartačním řízení stane. K vyznačení skartačního znaku se používají písmena A (archiv), S (skart nebo stoupa) nebo V (výběr), za která se napíše číslice, udávající počet roku skartační lhůty. Skartační lhůta začíná běžet 1. lednem následujícího roku po vyřízení dokumentu nebo uzavření spisu. Například značka S5 znamená dát do skartu (stoupy) za pět roků, A8 znamená předat do archivu k trvalému uložení za osm let a značka V3 znamená, že po třech letech se bude rozhodovat, zda dát písemnost do skartu nebo do archivu. Skartační řízení je součástí spisové služby, kterou spravuje stát dle zákona o archivnictví a spisové službě. Skartační řízení se řídí podle schváleného skartačního plánu a určuje tak postup při vyřazování dokumentů ze spisovny.

Nejdůležitější jsou dokumenty označené u původce spisovou značkou „A“, tyto dokumenty mají nejvyšší vypovídací hodnotu a jako takové se zasílají do archivu. Taktéž musí být ke skartačnímu řízení předloženy v psané či tištěné podobě. Soukromoprávní původci musejí vždy předkládat v tištěné formě dokumenty uvedené v příloze 1 zákona. Dokumenty, které nemají trvalou hodnotu, a tudíž jsou z dlouhodobého hlediska postradatelné označujeme jako „S“. Ty se dají předložit ke skartaci i mimo jiné jako datový záznam na disketě, CD či DVD.

Posledním označením dokumentů je skartační znak „V“ ten značí výběr. Významem jest, že jsou to dokumenty, u kterých se očekává přeřazení během skartačního řízení k dokumentům označeným jako „A“ a proto se taktéž musí předložit v tištěné podobě. Skartační lhůta začíná běžet 1. lednem následujícího roku po vyřízení dokumentu nebo uzavření spisu. Například značka S3 znamená dát do skartu (stoupy) za tři roky, A5 znamená předat do archivu k trvalému uložení za pět let a značka V3 znamená po třech letech rozhodnout, zda dát do skartu nebo do archivu. Ukládací lhůty se pohybují od několika let až po desítky let a jsou i různé podle znaků.

Stane-li se, že některé dokumenty v elektronické formě se nedají převést do tištěné podoby. V tomto případě se předkládají na typizovaných nosičích a v čitelném formátu, který je dostupný editorům či audio nebo video přehrávačům. Přičemž standarty určí Ministerstvo vnitra, odbor archivní správy či Národní archiv.

Skartační řízení probíhá za účasti pracovníka instituce na straně jedné a pracovníkem dohlížejícího archivu na straně druhé a společně hodnotí písemnosti a vytváří návrh na uschování písemnosti nebo její likvidaci. Při hodnocení písemnosti se přihlíží především k její obsahové hodnotě, ale věnuje se pozornost i k hodnotě vnější, tj. materiál, technika provedení, způsob a kvalita zhotovení. Písemnosti před rokem 1850, i při jejich méně významném obsahu, pro jejich stáří se neskartují. Dále se písemnosti následovně: Nejprve se hodnotí písemnosti týkající se vzniku instituce, jejich kompetencí, respektive jejího zániku. Na základě těchto znalostí lépe můžeme vyhodnotit písemnost a zařadit další materiál. Dále hodnotíme písemnosti ve vztahu k hlavní činnosti původce fondu jako např. směrnice, záznamy důležitých jednání vedoucích orgánů instituce nebo technickou dokumentaci. A jako třetí posuzujeme písemnosti dle jejich vědeckých hodnot. Archivář vyhodnocuje stav spisové služby instituce, účastní se vlastní přípravy skartace, připravuje dokumentární materiál k předání do archivu a provádí vnitřní skartaci v archivu. Za správné provedení skartačního řízení je zodpovědný vedoucí pracovník spisové služby.⁶⁹

4.5 Dnešní stav spisové služby

Nyní upravuje zásady spisové služby zákon č. 499/2004 Sb. a na něj navazující vyhláška č. 259/2012 Sb. a novela č. 283/2014 Sb. S rozvojem používání počítačů na úřadech od přelomu 20. století bylo potřeba legislativně upevnit a sjednotit elektronickou spisovou službu, tzv. eGovernment. Snahy zahájené zákonem č. 365/2000 Sb. byly dovršeny zákonem č. 300/2008 Sb. O elektronických úkonech a autorizované konverzi dokumentů spisovou službu dnes vykonávají zákonem uvedení původci. Lze jí vykonávat buď v listinné podobě za použití podacích deníků, nebo v elektronické podobě za použití elektronického systému spisové služby. Oblast pokrývá celou škálu činností, včetně příjmu, zpracování, označení, vyhodnocení, archivace a skartace dokumentů. Nezáleží, zda se jedná o úřední

⁶⁹ ŠTOURAČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

dokumenty, elektronické originály, datové zprávy nebo ručně psané tiskoviny. U všech oblastí platí, že některé zásady jsou striktně požadovány zákonem, další mohou být upraveny interními směrnici organizací. Dokumentem je tak každá písemná, obrazová, zvuková nebo jinak zaznamenaná informace, ať již v podobě analogové, či digitální, která byla vytvořena původcem, nebo byla původci doručena. Původcem se rozumí každý, z jehož činnosti vznikají dokumenty. Původci se dělí na veřejnoprávní a soukromoprávní.⁷⁰

Povinnost vykonávat spisovou službu mají ze zákona určení původci, což jsou organizační složky státu, státní příspěvkové organizace, státní podniky, organizační složky a právnické osoby založené nebo zřízené územními samosprávnými celky, školy a vysoké školy, právnické osoby zřízené zákonem a zdravotnická zařízení, kraje, hlavní město Praha, obce s pověřeným obecním úřadem a obce se stavebním a matričním úřadem, městská část nebo městský obvod územně členěného statutárního města a městská část hlavního města Prahy, na něž byla státem přenesena alespoň část působnosti obce s pověřeným obecním úřadem nebo působnosti stavebního nebo matričního úřadu.⁷¹ Spisová služba tak zajišťuje odbornou správu elektronických a listinných předmětových dokumentů.

⁷⁰ Spisová a archivní služba ve státní správě, samosprávě a v podnikatelské sféře:3. aktualizované a přepracované vydání. Praha: Linde, 2005, s. 13. ISBN 80-7201-549-4.

⁷¹ Tamtéž

5. Systémy v archivní praxi a spisovnách

Systémy, které se používají v jednotlivých archivech jsou vybírány s ohledem na stability, jednoduchost, přehlednost a rozšiřitelnost. Jedná se především o systémy založené na určitém typu databáze, které pomáhají archiváři s tvorbou archivních pomůcek, jejich vyhledávání a se správou většiny archivních fondů, které se v daném archivu nacházejí. Další programy užívané v archivech se týkají komunikačních, síťových či bezpečnostních záležitostí.

5.1 Přehled a popis systémů využívaných v archivech

Janus Archiv jedná se o databázový program, který slouží k tvorbě archivních pomůcek pro jednotlivé archivní fondy a k jednotlivému vyhledávání v rámci těchto pomůcek, k provádění tiskových činností pomůcek nebo jejich částí.

Mezi hlavní výhody programu Janus Archiv patří:

- Popis všech typů archiválií v rámci jednoho systému. Různé druhy archiválií, jako například mapy či obrazové dokumenty, mohou být popisovány v jednotném prostředí.
- Hierarchie záznamů v archivním fondu.
- Souvislost s principy mezinárodního standardu pro popis archiválií.
- Možnost přidávání nových atributů u popisu archiválií či nových polí do ukládacího formuláře.

V programu Janus Archiv vznikají archivní pomůcky jako databázové soubory, které se třídy podle jednotlivých záznamů v databázi. Program nabízí úpravu polí vkládacího formuláře pro jednotlivé archiválie jako jsou rukopisy, fotografie, zvukové dokumenty, mapy či sfragistické materiály. Pro archiválie běžné povahy jsou formuláře sjednocené. V případě převodu papírových archivních pomůcek do elektronické podoby umožňuje program jednoduchou konverzi v rámci systému. Janus Archiv také nabízí velké možnosti vyhledávání v rámci své databáze a umožňuje vytvářet kombinované dotazy.⁷²

⁷² STÁTNÍ OBLASTNÍ ARCHIV V PRAZE. Příloha č. 2 - Popis aplikací Státního oblastního archivu v Praze: Příloha č. 2 k zasedávací dokumentaci č.j.: 7773/30-2011. 2011. Dostupné také z: http://www.soapraha.cz/documents/praha/20120704170804-Priloha_c._2_k_zadavaci_dokumentaci.pdf

Bach VadeMeCum představuje modulární informační systém, který je postaven na specifických požadavcích na zpracování, prezentování a uložení archivních fondů, které se nachází ve státní i soukromé archivní správě. Software také podporuje vytváření archivních pomůcek pro zpracování archiválií. Program je navržen tak, aby efektivně rozšířil možnosti pracovníků v archivu pro zpracování a přístupu do historických materiálů a badatelům ke zkoumání archiválií v rámci informačních technologií.⁷³

Mezi hlavní výhody programu VadeMeCum patří:

- Možnost vytvořit fondy přehledné a prohledatelné v rámci dotazů
- Poskytuje efektivnější způsob evidence
- Ukládají data na jednom centrálním místě
- Snadný způsob pomocí webového řešení, kdy lze přistupovat k evidenci externě

Mezi jednotlivé součásti patří:

- Produkční část ProArchiv, který splňuje veškeré zákonné požadavky na zpracování archiválií. Má v sobě i podporu pro nová základní pravidla ke zpracování archiválií. Umožňuje také importovat data nejen z původních archivních pomůcek ale i z evidence zpracované starými pravidly. Součástí software jsou i nástroje na práci s digitalizáty archiválií, uložení a evidence podle typu archiválií. Dále nabízí standardy pro popis matrik, sfragistického materiálu a pořádací schémata jednotlivých typů archivních souborů.⁷⁴
- Prezentační portál poskytuje ucelené informace o zpracovaných fondech a archiváliích, v kterých lze jednoduše vyhledávat a prohlížet v digitální podobě.
- Badatelna umožňuje podporu pro evidování, vydávání a přípravu archivního materiálu
- Archivní repozitáře slouží pro ukládání dat a k ochraně zpracovaných archiválií, který funguje na centrální bázi.⁷⁵

⁷³ Bach systems, s.r.o.: Bach Vademecum [online]. [cit. 2019-04-23]. Dostupné z: <http://www.bach.cz/produkty/archivni-vademecum/>

⁷⁴ Tamtéž

⁷⁵ Tamtéž

Bach PROARCHIV – Pečetě – program evidující sfragistický materiál, jako jsou pečetidla, pečetě či sbírkové odlitky. Na operačním systému MS Windows funguje zcela bez problémů a existuje i online verze tohoto programu.⁷⁶

PEvA je program, který je určený primárně pro ukládání základních evidencí Národního archivního dědictví v síti archivů České republiky. Program také kromě povinné evidence Národního archivního dědictví umožňuje i doplňkovou evidenci, která je rozšířitelná. Pro plnou funkčnost programu PEvA je nutné na uživatelských počítačích se systémem Windows mít nainstalovaný MS RunTime moduly VFP ve verzi 8.⁷⁷

Mezi povinné evidence NAD patří:

- Evidence archivních pomůcek
- Kompletní evidence listů Národního archivního dědictví
- Evidence všech změn

Doplňkové evidence NAD obsahují jednotlivou lokaci.

REGiA slouží pro práci se Základním registrem Českého statistického úřadu. Oproti verzi z minulosti zvané Registr, nabízí grafické rozhraní je jednoduché a přehledné pro uživatele. Nabízí ovládání podobné jako ostatní aplikace typu PEvA či Bach. Základní úlohou pro program REGiA je práce s registrem Českého statistického úřadu, kde nabízí vyhledávání či tisk a dále umožňuje editaci jednotlivých poznámek, které se nachází v registru včetně zápisu čísel jednotlivých archivů. Poté může být v centrálních státních oblastních archivů program využíván pro vyhledávání a zobrazování registrů jednotlivých oblastních archivů.⁷⁸

WordPress je hojně využívaný redakční systém (CMS), který pokrývá zhruba 33% všech webů a toto číslo roste každým dnem. Je vhodný pro použití na blogy, informační portály, zábavní weby či webové aplikace. Wordpress kombinuje jednoduchost a komplexnost pro uživatele a autory článků, kteří vykonávají základní úkony, jako je vkládání, editace či mazání článků. Ovšem Wordpress

⁷⁶ STÁTNÍ OBLASTNÍ ARCHIV V PRAZE. Příloha č. 2 - Popis aplikací Státního oblastního archivu v Praze: Příloha č. 2 k zasedávací dokumentaci č.j.: 7773/30-2011. 2011. Dostupné také z: http://www.soapraha.cz/documents/praha/20120704170804-Priloha_c._2_k_zadavaci_dokumentaci.pdf

⁷⁷ Tamtéž

⁷⁸ Tamtéž

nabízí i velkou škálu rozšíření, kdy pomocí pluginů lze webové prostředí dostatečně rozšířit.⁷⁹

Antivir AVG – antivirový program, který se stará o bezpečnost sítě v daném archivu.

IM Jabber klient Spark – program, který umožňuje okamžité posílání zpráv mezi jednotlivými pracovníky v rámci archivu.

Mail Admin – program, který zprostředkovává komunikaci pomocí emailů, kdy uživatelé se připojí pomocí protokolu IMAP. Nachází se většinou přímo na fyzickém serveru archivu.

5.2 Přehled a popis systémů používaných pro spisovou službu

Spisová služba WISPI

Jedná se o informační systém, který poskytuje zpracování dokumentů v rámci podatelny či spisovny v rámci určité organizace. Umožňuje také evidenci dokumentů, jednacích čísel, přeposílání dokumentů v rámci společnosti, vytváření spisů se zpracováním a ukládáním přímo do archivu. Spisová služba WISPI si zakládá především na přehledném uživatelském rozhraní a snadném používání pomocí standartního webového prohlížeče. Zároveň lze program propojit s elektronickými úložišti, které garantují čitelnost a věrohodnost dle eIDAS.

Součástí Spisové služby WISPI jsou následující moduly:

- WISPI DS – pro odesílání, přijímání a ukládání datových zpráv.
- WISPI ePodatelna – přijímání a odesílání dokumentů pomocí emailu.
- WISPI eSpisovna
- WISPI SSL – evidence, vyřizování dokumentů a spisů.

Dále umožňuje také integraci CzechPointOffice, spojení se šablonami v MS Office, evidenci požadavků GDPR či integraci s dalšími systémy.

GINIS

Představuje nejrozšířenější informační systém, který je používán ve veřejné správě. Umožňuje velké množství úkonů, které lze provádět s dokumenty. S programem GINIS lze vytvářet, vyhledávat či odesílat k úschově různé typy

⁷⁹ WordPress.org [online]. [cit. 2019-04-25]. Dostupné z: <https://wordpress.org/about/features>

dokumentů. Nabízí dohledání dokumentů dle filtrů a předání spisů a dokumentů do spisovny.⁸⁰

ELISA

Je monitorovací software, který sleduje písemnosti v organizaci od přijetí na podatelnu až po archivaci a skartaci. Díky tomu mají organizace možnost sledovat a mapovat chování jednotlivých písemností. Program také ulehčuje práci s evidencí jednotlivé přijaté a odeslané pošty, zapisování do knih či mnohem rychlejší předávání a vyřizování dokumentů.⁸¹

Spisová služba – program, který se používá pro sledování oběhu dokumentů. Jedná se o plné a fungující řešení v rámci ukládání elektronických dokumentů. Sleduje celý životní cyklus dokumentu, jako je jeho vznik, poté uložení v úložišti, tvorba spisu a následné uložení do spisovny, a nakonec jeho odeslání do digitálního archivu.

e-Spis

Software, který se používá při správě, ukládání a řízení dokumentů ve veřejné správě. Nabízí pohodlnou možnost správy dokumentů v rámci velké organizace až dva tisíce zaměstnanců. Nabízí jednoduché napojení do programů třetích stran. Je využíván Českým báňským úřadem či Českou národní bankou.⁸²

ARSYS

Program, který se stará o registraci pošty, kdy vytvoří elektronický záznam, který je použitý při přijetí do podatelny. Tento elektronický záznam obsahuje informace o zásilce a odesílateli. Program ARSYS je používán ministerstvem pro místní rozvoj a ministerstvem práce a sociálních věcí.⁸³

AthenA

Jedná se o program, který nabízí řešení pro archivaci dokumentů a spisů. Řeší také jejich skartaci a uchovávání její vizuální podoby včetně verzování.⁸⁴

⁸⁰ Produkty. GORDIC, [online]. Jihlava: GORDIC, [cit. 2016-04-05]. Dostupné z: <https://www.gordic.cz/produkty/>

⁸¹ KUNT, Miroslav a Tomáš LECHNER, 2015. Spisová služba. 1. vyd. Praha: Leges, 400 s. ISBN 978-80-7502-083-3

⁸² Tamtéž

⁸³ Tamtéž

⁸⁴ Tamtéž

6. Vývoj webové aplikace

Jak bylo v úvodu práce uvedeno, praktickým výstupem práce je webová aplikace sloužící pro monitorování serverů, na kterých jsou umístěny systémy využívané v archivnictví.

Celý systém je vytvořený pomocí programovacího jazyka Javascript + PHP. Dále se zde využívají webové jazyky HTML a CSS, které slouží pro vykreslování stránky v prohlížeči. Systém se bude nacházet na určité webové adrese, která bude dostupná externě a administrátor se tak bude mít možnost přihlásit i z jiného místa, než je archiv. Celá aplikace využívá relační databázi MySQL, ke které se připojuje a z které čerpá veškerá data o serverech, uživatelích, notifikacích a zapisuje zde i chybové hlásky.

6.1 Návrh systému

6.1.1 Grafický návrh webové aplikace

Nejprve je nutné připravit vývojové prostředí, ve kterém se bude vyvíjet celá aplikace. Toto prostředí se bude skládat z textového editoru, virtuálního serveru zvaného Docker, který v sobě obsahuje jak webový server, tak i server pro odesílání notifikací na emailovou adresu a do systému.

Před samotným vývojem je nutné udělat grafický návrh, jak taková webová aplikace bude vypadat a poté je potřeba provést i návrh jednotlivé databáze a jakých služeb bude daná aplikace využívat. Návrhy musí být dostatečně uživatelsky přívětivé a mají splňovat nároky na chod aplikace, aby byla plynulá a přehledná.

Pro vytvoření přehledů je vhodné použít některý z grafických programů, jako jsou Adobe Photoshop, GIMP, Affinity Designer či Sketch. V tomto případě byl zvolen Affinity Designer, který je z této skupiny programů nejnovější a je vhodný pro návrhy webových stránek díky svým integrovaným funkcím. Pro náčrt struktury webové stránky je použito obrazců, které představují určité elementy na stránce. Program umí exportovat vytvořený návrh do všech klasických obrázkových přípon a není tak problém ho otevřít na většině dnešních zařízení.

Návrh spočívá v nakreslení důležitých částí webových stránek, které budou nejvíce používány v rámci interakce mezi uživatelem a systémem. V rámci návrhu se

neuvádí jednotlivé detaily, které mohou být dodělané poté během vývoje. Vytvoří se tak pouze kostra, která později může být doplněná o některé detaily, jako může být jednotlivé uspořádání barev či nepodstatné detaily u jednotlivých částí stránek. První návrh se týká přihlašovací stránky, kdy je použita základní kostra, která je k vidění u většiny webových stránek. Při zobrazení stránky by se měl objevit dostatečně velký text oznamující, kde se uživatel momentálně nachází a o jaký přihlašovací formulář se jedná. Dále se zde vyskytuje klasický přihlašovací formulář, který je složen ze dvou inputů, které slouží pro zadání přihlašovacích údajů. Oba inputy by měly být dostatečně velké a neměly by obsahovat kontrastní barvy, díky kterým by text nemusel být viditelný v porovnání s pozadím. U hesla by při zadávání neměly být vidět písmena a místo nich by se měly zobrazovat pouze hvězdičky. Je to z důvodu, aby nebylo viditelné, jaké heslo se zadává v případě, že se administrátor přihlašuje z nějakého veřejného místa.

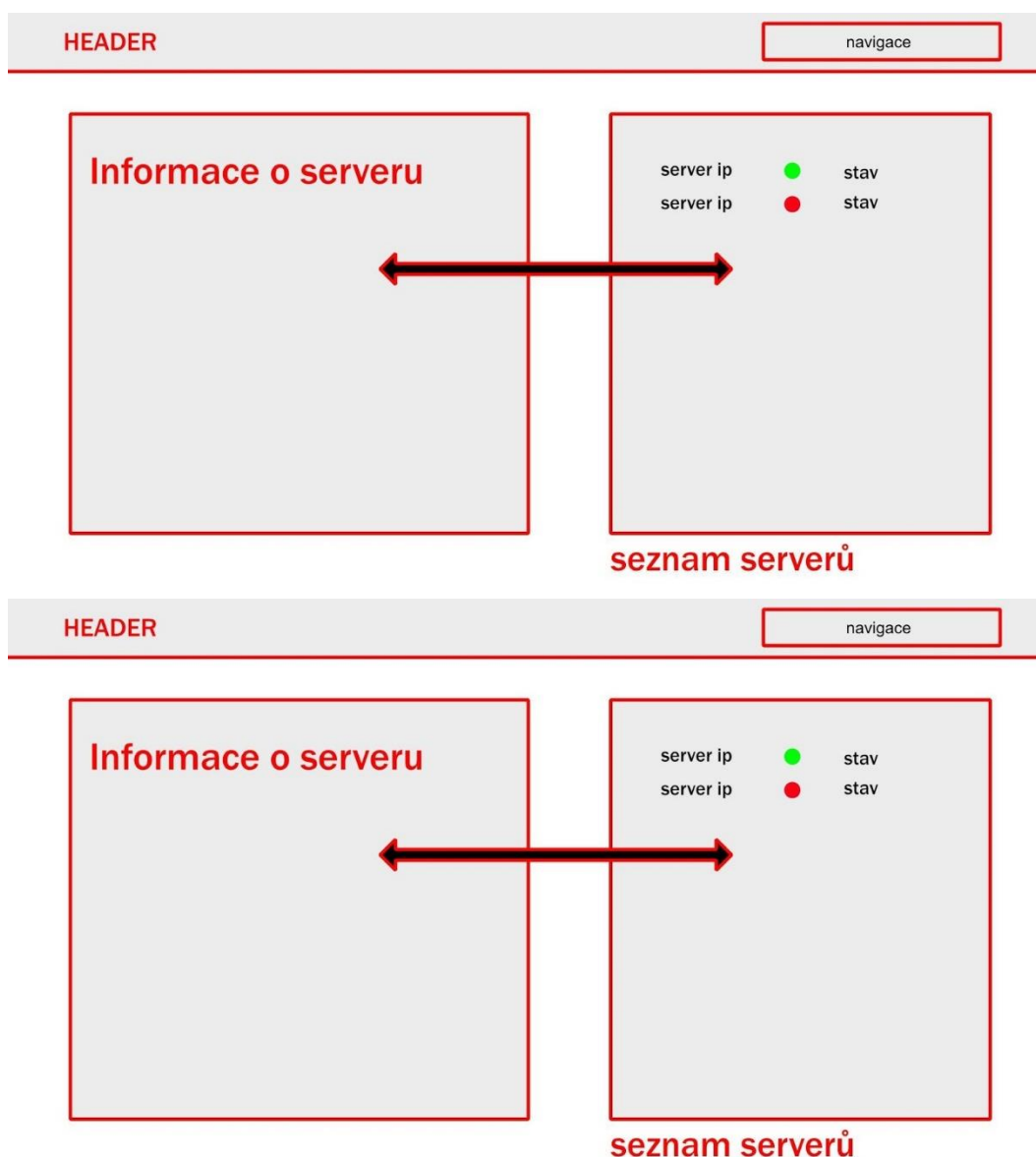
Posledním důležitým elementem je přihlašovací tlačítko, kterým uživatel potvrzuje vyplněné inputy a poté se celý formulář odešle na server, kde se přihlašovací údaje zpracují a server povolí uživateli přístup do systému či mu zobrazí chybovou hlášku o nesprávnosti přihlašovacích údajů. V případě chybného zadání údajů systém zobrazí hlášku o nevalidním uživatelském jméně a heslu.



Diagram showing the layout of a login form. At the top is the title "PŘIHLÁŠENÍ". Below it are two input fields: "INPUT - UŽIV. JMÉNO" and "INPUT - HESLO". At the bottom is a button labeled "Přihlašovací tlačítko".

Obr. 5: Návrh přihlašovací obrazovky, zdroj: vlastní zpracování

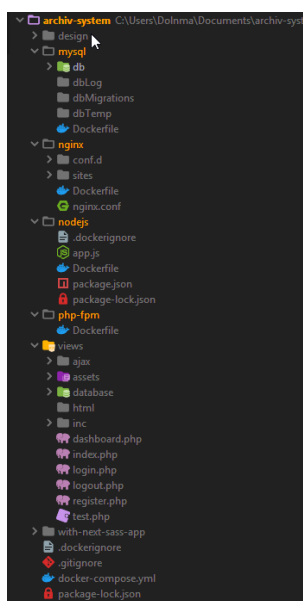
Po úspěšném přihlášení je poté uživatel přesměrován přímo do systému, kde může sledovat aktivitu serverů formou nástěnky, kde se nachází všechny servery a informace o jejich aktuálním stavu. Toto uživatelské rozhraní musí zobrazovat menu neboli navigaci, dále panel pro zobrazení jednotlivých serverů v rámci celého systému a při kliknutí na jednotlivý server se zobrazí i informace o jeho aktivitě a k jakým účelům je daný server používán. Vše by mělo být ideálně na jedné obrazovce bez procházení více stran a v rámci rychlosti by se mělo přepínání vykonávat v rámci jednoho panelu.



Obr. 6: Návrh administračního rozhraní, zdroj: vlastní zpracování

Jelikož se jedná o webovou aplikaci, tak zde budou potřeba programy se správnou syntaxí programovacího jazyka jako je HTML, CSS, JavaScript či PHP. Pro jejich editaci je zvolen editor PHPStorm, který je dostupný pro studentské účely zcela zdarma. Jedná se o jeden z nejlepších nástrojů pro editaci těchto typů souborů a je využíván velkými softwarovými firmami po celém světě.

Protože se jedná o komplexní aplikaci, tak není potřeba editor různě nastavovat. Vše je již připraveno na práci se soubory, které budou umístěny na webový server. Pomocí tohoto programu bude možné vidět i chyby, které se během programování mohou vyskytnout, protože editor obsahuje chytrý indexátor, který dokáže zanalyzovat kód a upozornit tak uživatele, že není napsán podle norem a standard daného jazyka. V tomto editoru je potřeba vytvořit jednotlivé soubory, které budou aplikovány v rámci projektu a vytvoříme tak celou strukturu, která bude dostatečně přehledná a obsáhlá vůči velikosti projektu. Zde je zobrazení struktury souborů, které se nacházejí v tomto projektu:



Obr. 7: Ukázka souborové struktury projektu, zdroj: vlastní zpracování

6.1.2 Návrh databáze

Součástí návrhu aplikace je i návrh databáze, na které tato aplikace bude běžet. Jako typ databáze byla zvolena databáze MySQL, která je vhodná na většinu dnešních webových projektů a aplikací díky své jednoduchosti a rozšiřitelnosti. Návrh

databáze autor utváří tak, aby splňoval nároky na fungování webové aplikace. Důležité je vytvořit jednoduchou strukturu, která se postará o uložení všech podstatných informací bez nutnosti vytvářet velké množství tabulek a sloupců v nich obsažených. Databáze celého projektu je vytvořena z celkem pěti tabulek, které jsou rozděleny na dvě části. Dvě tabulky se týkají uživatelských účtů, které slouží pro přístup do systému a tři tabulky řeší závislosti mezi jednotlivými servery a jejich aktuálními stavy.

Tabulka *users* viz Tab. 1 slouží především pro zadání jednotlivých uživatelů, kteří mají mít oprávnění pro přihlášení do systému a evidenci samotných uživatelů v rámci systému.

Tabulka users		
Název sloupce	Datový typ	Účel sloupce
id	int, délka 5 znaků	unikátní identifikační číslo uživatele
email	varchar, délka 255 znaků	uložení emailové adresy uživatele
username	varchar, délka 255 znaků	uživatelské jméno
password	varchar, délka 200 znaků	heslo pro přihlášení uživatele do systému
reg_time	timestamp	datum posledního přihlášení
permission_id	int, délka 5 znaků	vyplněné id pravomocí, které jsou definovány v tabulce <i>permissions</i>

Tab. 1: Přehled datových typů u tabulky users, zdroj: vlastní zpracování

Další tabulka *permissions* (Tab. 2) slouží pro zadávání jednotlivých pravomocí, které jsou přiřazovány k jednotlivým uživatelům. Jedná se o udělení práv na určité úkony, které může uživatel v daném systému vykonávat.

Tabulka permissions		
Název sloupce	Datový typ	Účel sloupce
id	int, délka 5 znaků	unikátní identifikační číslo pravomoci uživatele
a	tinyint, délka 1 znak	určité právo, které je zapsáno číslem 0 v případě pasivního statusu či 1 v případě aktivního statusu
b	tinyint, délka 1 znak	určité právo, které je zapsáno číslem 0 v případě pasivního statusu či 1 v případě aktivního statusu
c	tinyint, délka 1 znak	určité právo, které je zapsáno číslem 0 v případě pasivního statusu či 1 v případě aktivního statusu

Tab. 2: Přehled datových typů u tabulky permissions, zdroj: vlastní zpracování

V tabulce s názvem *servers* (Tab.3) jsou zapsány informace o serveru a technické parametry, které se poté využívají při sledování serverů. Většina těchto parametrů je povinných pro plnou funkčnost systému.

Tabulka servers		
Název sloupce	Datový typ	Účel sloupce
id	int, délka 5 znaků	unikátní identifikační číslo serveru
name	varchar, délka 255 znaků	název serveru
description	varchar, délka 255 znaků	popis serveru či komentář
ip	varchar, délka 100 znaků	IP adresa serveru, která obsahuje jak čísla, tak i interpunkční znaménka
port	int, délka 5 znaků	číslo portu, na kterém daný server běží
type_id	int, délka 11 znaků	zadaný typ serveru, který je definován v tabulce <i>type</i>
platform	int, délka 11 znaků	typ zařízení

Tab. 3: Přehled datových typů u tabulky servers, zdroj: vlastní zpracování

Tyto servery je však důležité sledovat a jejich chování evidovat. K tomu je vytvořena tabulka s názvem *logs* (Tab.4), kam se zapisují výpadky serverů, které sledujeme.

Tabulka logs		
Název sloupce	Datový typ	Účel sloupce
id	int, délka 5 znaků	unikátní identifikační číslo chybové hlášky
server_id	int, délka 5 znaků	id serveru, ke kterému se daná chybová hláška vztahuje
message	varchar, délka 255 znaků	daná chybová hláška
status	tinyint, délka 1 znak	aktuální stav serveru, jestli je aktivní či není dostupný
date	datetime	zaznamenaný přesný čas měření

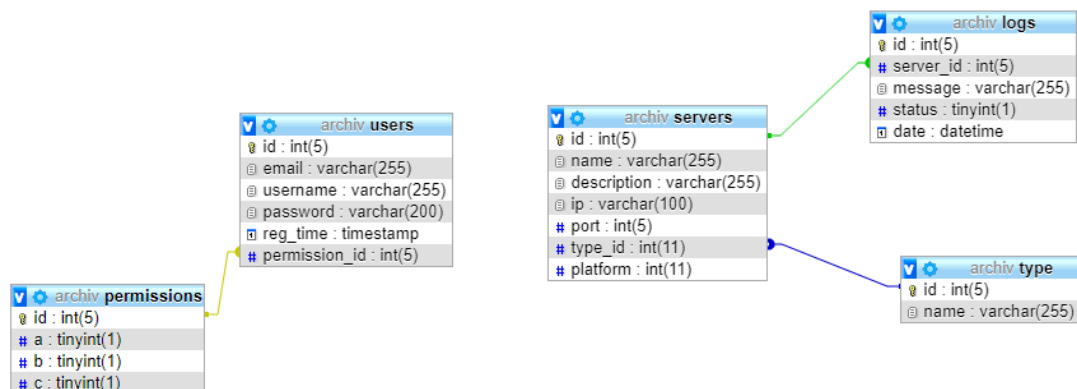
Tab. 4: Přehled datových typů u tabulky logs, zdroj: vlastní zpracování

Jelikož se v archivech nachází více druhů serverů a každý z nich má jinou funkci, je možné sledovat jednotlivá zařízení, jako může být síťová tiskárna, jednotlivé počítače nebo elektronická zařízení, která jsou připojená v rámci sítě. Pro toto odlišení je vytvořena tabulka *type*(Tab. 5), která se vždy váže k danému serveru.

Tabulka type		
Název sloupce	Datový typ	Účel sloupce
id	int, délka 5 znaků	unikátní identifikační číslo typu serveru
name	varchar, délka 255 znaků	jméno či popis typu serveru

Tab. 5: Přehled datových typů u tabulky type, zdroj: vlastní zpracování

Pro lepší přehlednost databázové struktury autor vytváří návrh databáze ve webovém programu *phpMyAdmin*, který umožňuje grafické zobrazení aktuální databáze. Toto zobrazení je vidět na Obr. 8.



Obr. 8: Návrh databáze, zdroj: vlastní zpracování

6.2 Konfigurace webových služeb

Po úspěšném vytvoření návrhu databáze je nutné přejít k samotné instalaci vývojového prostředí. V tomto případě byl zvolen pro vývojové prostředí virtualizační nástroj zvaný Docker, který obstarává činnost všech webových služeb jako je PHP, NodeJs, MySQL či Nginx. Docker jako takový se nastavuje v souboru s názvem *Dockerfile*, který je vytvořen v každé složce s danou službou. Nejprve tedy bude nutné vytvořit jednotlivé služby zvlášť a poté lze všechny tyto služby spustit pomocí příkazu *docker-compose*, který tyto knihovny všechny vyvolá v určitém pořadí.

První webová služba, kterou autor vytváří, se jmenuje Nginx a slouží pro správné zobrazování statického obsahu ve webovém prohlížeči. Stará se o vytváření jednotlivých stránek, ke kterým se přistupuje pomocí *url* adres z webového prohlížeče. V rámci *Dockerfile* je nutné definovat knihovnu příkazem *FROM*, ze které daný obraz s webovou službou pochází a porty příkazem *EXPOSE*, na kterých daná služba poběží.

Ukázka kódu pro vytvoření služby Nginx v rámci Dockeru:

```
FROM nginx:alpine
CMD ["nginx"]
EXPOSE 80 443
```


V další fázi je nutno definovat službu MySQL, což je služba, která má za úkol vytvoření databáze a její fungování. Podobně jako tomu bylo u vytvoření služby Nginx, tak je zde potřeba definovat soubor *Dockerfile*, kterým je potřeba stáhnout určitý balíček se službou MySQL opět příkazem *FROM*, který je v rámci takzvaného *DockerHubu*. Portem pro komunikaci s databází se na většině serverů používá *port* 3306.

Ukázka kódu pro vytvoření databázové služby MySQL v rámci systému Docker:

```
FROM mysql:8.0.15
CMD ["mysql"]
EXPOSE 3306
```

Podstatnou částí pro fungování projektu je podpora jazyka PHP, který obstarává komunikaci mezi databází a webovým serverem a jeho cílem je vykreslit správně data v určité podobě. V konfiguraci *Dockerfile* pro tuto službu se definuje balíček s PHP v poslední verzi, tedy 7.4. Celý tento balíček je s parametrem „alpine“, který určuje, že má být balíček v nejomezenější verzi bez přebytečných nadstaveb a knihoven. Během konfigurace se musí také definovat příkazy *docker-php-ext-install*, *pdo* a *pdo_mysql* pro instalaci dodatečných knihoven, které zaručují fungování v systému Docker a funkční propojení s databází MySQL.

Ukázka kódu pro vytvoření služby PHP v souboru *Dockerfile*:

```
FROM php:fpm-alpine
RUN docker-php-ext-install pdo pdo_mysql
CMD ["php-fpm"]
EXPOSE 9000
```

Poslední částí v rámci Dockeru je zprovoznění JavaScriptového balíčkového systému zvaného NodeJS, který se stará o sledování dostupnosti serverů. Docker spustí JavaScriptový skript, který vykonává určité činnosti, které jsou ve spolupráci s databází, a probíhá tak čtení a zapisování dat. Pro plné fungování NodeJS je potřeba instalace knihoven pro plnou funkčnost daného skriptu příkazem *RUN npm install*, který provede instalaci balíčků do složky *node_modules* v rámci projektu, ze kterého se dále čerpá původní script, který je umístěn ve složce *nodejs* s názvem *app.js*. V *Dockerfile* je definována osmá verze *nodejs* s označením „jessie“, která podporuje nové standardy v oblasti JavaScriptu. Dále se musí provést

pomocí příkazu *RUN mkdir* také vytvoření pracovní složky s názvem *app*, do které se přenesou všechny požadované soubory.

Kód, který se nachází v Dockerfile pro zprovoznění NodeJs:

```
FROM node:8-jessie
RUN mkdir /app
WORKDIR /app
COPY package.json /app
RUN npm install
COPY . /app
CMD ["npm", "start"]
```

Po vytvoření všech důležitých konfiguračních souborů, které jsou použity při spouštění Dockeru, může autor začít s vytvořením souboru v *docker-compose.yml*, který se stará o sestavení jednotlivých částí, které jsou definované v Dockerfile v jednotlivých složkách, které jsou pojmenovány podle názvu webových služeb. Záhloví souboru musí obsahovat verzi sestavení vyjádřenou názvem „version“, kdy je použita verze 3.7, která patří mezi aktuálně nejnovější. V další fázi je nutné definovat jednotlivé služby parametrem „services“.

Pro službu PHP je nutné určit parametrem „volumes“, kde bude její kořenová složka, tedy v tomto případě složka *views*, pro zobrazování obsahu, který se vykresluje skrze jazyk PHP. Zde jsou uloženy veškeré soubory s příponou *.php* a ostatní soubory, které jsou potřeba pro zobrazování stránek v prohlížeči. V údaji „build“ je zapsáno, v jaké složce se nachází soubor Dockerfile. V tomto případě tedy *php-fpm*.

Ukázka celého kódu služby PHP v souboru *docker-compose.yml*:

```
php-fpm:
  build:
    context: ./php-fpm
  volumes:
    - ./views:/var/www
```

Služba, která obstarává fungování databáze je nazvána jako *mysql*, kdy v rámci její definice je nutné určit, na jakých portech bude pracovat. V tomto případě jde o port 3306, vnitřní 3307, který se definuje proměnnou „ports“. Dále autor musí zadat parametr „*--default-authentication-plugin=mysql_native_password*“, který se definuje v rámci proměnné „command“ a určuje, aby se používalo heslo nastavené v konfiguračním souboru zadané v proměnné „environment“, což je pro tento

projekt „root“. Z důvodů zobrazení logů a ukládání databáze do složek je nutné zadat cesty i do parametru „volumes“, který definuje, kam se mají data z databáze lokálně ukládat. Další služba, která je úzce spjatá s MySQL se nazývá *adminer*. Jedná se o systém, vytvořený v programovacím jazyce PHP, který se zaměřuje na komunikaci mezi uživatelem a databází. Obsahuje grafickou nadstavbu, která se zobrazuje v rámci webového prohlížeče, a přes kterou může uživatel interagovat se systémem. Port, na kterém poběží služba *adminer*, je nastaven autorem na hodnotu 8080, takže bude systém uživateli dostupný při zadání do prohlížeče adresu *http://localhost:8080*. Atribut „depends_on“ znamená, že spuštění služby *adminer*, bude čekat na úspěšné spuštění služby zajišťující chod PHP jazyka.

Ukázka celého kódu databázové služby MySQL a *adminer* v souboru *docker-compose.yml*:

```
mysql:
  build:
    context: ./mysql
  ports:
    - "3307:3306"
  command:
    - "--default-authentication-plugin=mysql_native_password"
  environment:
    - MYSQL_ROOT_PASSWORD=root
  volumes:
    - ./mysql/db:/var/lib/mysql
    - ./mysql/dbTemp:/temp/mysql
    - ./mysql/dbLog:/var/log/mysql
    - ./mysql/dbMigrations:/docker-entrypoint-initdb.d

adminer:
  image: adminer
  restart: always
  ports:
    - 8080:8080
  depends_on:
    - php-fpm
  links:
    - mysql
```

Dvě poslední webové služby Nginx a NodeJS jsou spouštěny obdobným způsobem jako služby uvedené v textu výše. Nginx běží na portu 80 pro protokol *http* a na portu 443 pro zašifrovaný protokol *https*. Přes parametr „volumes“ jsou

nastavovány cesty ke konfiguračním samotného Nginx, jako jsou pravidla pro přístupování k *url* adresám a oprávněním. V případě NodeJS je potřeba nastavit porty, v tomto projektu 3000, a celá služba má vytvořenou závislost pomocí proměnné „depends_on“ na databázové službě *mysql*.

Celý konfigurační kód pro spuštění webových služeb Nginx a NodeJS v souboru *docker-compose.yml*:

```
nginx:
  build:
    context: ./nginx
  container_name: webserver
  restart: unless-stopped
  ports:
    - "80:80"
    - "443:443"
  volumes:
    - ./views:/var/www
    - ./nginx/nginx.conf:/etc/nginx/nginx.conf
    - ./nginx/sites:/etc/nginx/sites-available
    - ./nginx/conf.d:/etc/nginx/conf.d
    - ./nginx/logs:/var/log/nginx
  depends_on:
    - php-fpm

nodejs:
  build:
    context: ./nodejs
  volumes:
    - ./nodejs:/nodejs/app/
    - ./nodejs/app/node_modules/
  ports:
    - 3000:3000
  container_name: nodejs
  restart: always
  depends_on:
    - mysql
```

Po dokončení konfigurace v souboru *docker-compose.yml* je nutné všechny tyto služby vytvořit příkazem:

```
docker-compose build
```

Tento příkaz sestaví jednotlivé balíčky, jejichž definice je vždy zadána v příslušném adresáři se službou v souboru Dockerfile. Po úspěšném sestavení

všech balíčků může dojít ke spuštění všech služeb, které jsou definovány v souboru *docker-compose.yml*. Příkaz pro spuštění všech webových služeb:

```
docker-compose up
```

6.3 Tvorba vývojového prostředí

6.3.1 Vytvoření Databáze

Po vytvoření a spuštění všech důležitých webových služeb v rámci systému Docker, může autor začít s vytvořením databáze. Pro přístup do databáze využívá webové služby adminer, která je přístupná přes internetový prohlížeč na webové adrese *http://localhost:8080*. Zde je nutné se přihlásit přes přihlašovací obrazovku, kam je nutné zadat přihlašovací údaje. Tyto údaje jsou závislé na předešlé konfiguraci v souboru *docker-compose.yml*. Pro přihlášení je tedy nutné zadat do pole server hodnotu „mysql“, do pole uživatel hodnotu „root“ a do pole heslo hodnotu „root“. Po úspěšném přihlášení je nutné vytvořit novou databázi, která bude mít název „archiv“ a porovnávání „utf8_czech_ci“, kvůli podpoře české diakritiky. Dále následuje vytvoření jednotlivých tabulek logs, permissions, servers, type a users dle databázového návrhu, který autor vytváří na začátku projektu. Pro všechny tyto tabulky autor práce používá porovnávání „utf8_bin“. Po vytvoření všech tabulek včetně sloupců a propojení pomocí klíčů je databáze nastavena a může se přejít na vytváření samotného systému.

The screenshot shows the Adminer interface for a MySQL database named 'archiv'. The main content area displays a table listing for 'Tabulky a pohledy'. The table has columns for 'Tabulka', 'Úložště', 'Porovnávání', 'Velikost dat', 'Velikost indexů', 'Volné místo', 'Auto Increment', 'Řádků', and 'Komentář'. The rows listed are 'logs', 'permissions', 'servers', 'type', and 'users', along with a '5 celkem' (5 total) row. Below the table, there are buttons for 'Analyzovat', 'Optimalizovat', 'Zkontrolovat', 'Opravit', 'Vyprázdnit', and 'Odstranit'. There is also a dropdown menu for 'Přesunout do jiné databáze' and buttons for 'Přesunout' and 'Zkopírovat'.

Tabulka	Úložště	Porovnávání	Velikost dat	Velikost indexů	Volné místo	Auto Increment	Řádků	Komentář
logs	InnoDB	utf8_bin	2 637 824	491 520	4 194 304	63 499	~ 33 380	
permissions	InnoDB	utf8_bin	16 384	0	0	2	0	
servers	InnoDB	utf8_bin	16 384	16 384	0	4	~ 2	
type	InnoDB	utf8_bin	16 384	0	0	5	~ 2	
users	InnoDB	utf8_bin	16 384	16 384	0	3	~ 1	
5 celkem		utf8_czech_ci	2 703 360	524 288	0			

Obr. 9: Zobrazení databáze v systému adminer, zdroj: vlastní zpracování

6.3.2 Vytvoření systému pro sledování serverů a zasílání notifikací

Pomocí nakonfigurování souboru Dockerfile a následného spuštění služby NodeJs v rámci příkazu *docker-compose up* je možné začít s vytvářením skriptu pro sledování aktivity serverů, zapisování do databáze a zasílání emailů v případě výpadku. Celý tento systém je vytvořen pomocí softwarového systému NodeJS založeném na programovacím jazyce JavaScript, který se stará o celý chod systému pro sledování dostupnosti serverů a odesílání notifikací na určitý email.

Celý skript je uložen v souboru *app.js*, který se nachází ve složce *nodejs*. Tento skript se poté spouští pomocí NodeJS příkazem „npm start“. V úvodu souboru se vždy definují jednotlivé knihovny, které jsou nutné pro fungování všech funkcí v daném skriptu. Vždy je nutno zadat proměnnou dané služby vyjádřenou jako „const“ a poté pomocí „require“ si danou knihovnu vyžádá, kdy cesta dané knihovny je definována v souboru *package.json*.

Ukázka načtení jednotlivých balíčků v souboru *app.js*:

```
const express = require('express');
const app = express();
const router = express.Router();
const mysql = require('mysql');
const ping = require("net-ping");
const nodemailer = require('nodemailer');
```

Pro připojení k databázi je nutné vytvořit konfigurační proměnnou, která definuje přihlašovací údaje pro přihlášení k MySQL serveru. Skript se bude připojovat k databázi pomocí funkce *createConnection*. Zde autor práce zadává údaje, které byly zadány při konfiguraci služeb v systému Docker. Do parametru „host“ je zadáno jméno služby *mysql* a tato služba běží na portu 3306, která je určena v parametru s názvem „port“. Pro přihlášení do databáze je také důležité zadat uživatelské jméno a heslo, kdy je nutno vyplnit položky „user“ a „password“. Jelikož může být více databází, tak je zde nutné uvést i její název. Ten se zadává do parametru „database“ a v tomto případě je to název „archiv“.

Ukázka napojení na databázi MySQL v souboru *app.js*:

```
var con = mysql.createConnection({
  host: "mysql",
  port: "3306",
  user: "root",
```

```
password: "root",
database: "archiv"
});
```

Po úspěšném připojení k databázi je nutné zajistit kontrolu dostupnosti serverů v reálném čase. Tohoto autor docílí pomocí knihovny *net-ping*, která obsahuje funkci *pingHost*, která získává data z určitých IP adres a poté lze zjistit, zda je server dostupný či nikoliv. Pro vykonání této funkce je zapotřebí získat jednotlivé IP adresy serverů, které se nachází v databázi. Toho autor docílí pomocí funkce *query*, která vykoná zadaný SQL příkaz, který se posílá v rámci funkce.

Ukázka SQL příkazu pro získání IP adres serverů z databáze:

```
SELECT id,ip FROM `servers` ORDER BY `id`
```

Pokud funkce *pingHost* získá odpověď od IP adresy daného serveru, tak její stav zapíše do databáze v tabulce s názvem logs. Rozhodnutí o tom, zdali je server aktivní je tvořeno formou podmínky *IF*, která v případě, že daná IP adresa není dostupná, tak zapíše do sloupce „status“ hodnotu 0. Pakliže je server aktivní, tak se v kódu použije *ELSE* a aktivita se zapíše hodnotu 1. Ukázka SQL příkazu, který zapíše aktuální status serveru do databáze:

```
INSERT INTO logs (server_id, message, status, date) VALUES (' + serverId + ', ' + message + ', '1', ' + date + '')
```

Zasílání notifikací formou emailu obstarává balíček s názvem *nodemailer*, který se definuje na začátku souboru *app.js*. Autor práce tento balíček nastavil tak, aby využíval emailových schránek společnosti Google a stačilo zadat pouze přihlašovací údaje k účtu. Pro konfiguraci tohoto kroku je potřeba vyplnit parametry ve funkci *createTransport*, která je součástí balíčku *nodemailer*. Zde je potřeba vyplnit parametr „host“ a dále přihlašovací údaje v parametru „auth“. Pro testovací účely při vývoji projektu byl použit smtp server společnosti Google.

Ukázka konfigurace zdrojové emailové schránky pro odesílání notifikací v souboru *app.js*:

```
let transporter = nodemailer.createTransport({
  host: 'smtp.gmail.com',
  port: 587,
  secure: false,
  requireTLS: true,
  auth: {
    user: '',
```

```
    pass: ''
  }
});
```

Po nastavení odesílatele notifikací je také nutné nastavit formát a adresu, kam se mají notifikační emaily odesílat. Pro tuto úpravu je vytvořena proměnná *mailOptions*, kde jsou zapsány všechny potřebné parametry, které jsou používány při odesílání emailů v rámci funkce *sendMail*. Nastavení emailu se mění v závislosti na splnění podmínky spjaté s funkcí *pingHost*, která je definována v textu výše.

Ukázka vytvoření konfigurace pro odesílání emailů v souboru *app.js*:

```
let mailOptions = {
  from: 'dolnma9@gmail.com',
  to: 'dolnma9@gmail.com',
  subject: 'Oznámení o výpadku serveru',
  text: 'Byl zjištěn výpadek archivního serveru. Zkontrolujte prosím dostupnost archivních systémů.'
};
```

Ukázka funkce pro odeslání emailu dle instrukcí definovaných proměnnými v souboru *app.js*:

```
transporter.sendMail(mailOptions, (error, info) => {
  if (error) {
    return console.log(error.message);
  }
  console.log('success');
});
```

6.3.3 Vývoj webového systému

Všechny soubory, které jsou určeny pro zobrazení systému ve webovém prohlížeči se nachází v adresáři s názvem *views*. Zde se nachází soubory s příponou *.php*, *.html*, *.css* či *.js*. Na úvod je důležité vytvořit základní kostru PHP dokumentu. Je tedy nutné vytvořit soubor s názvem *index.php*, který slouží jako domovská stránka při zadání adresy *http://localhost* do webového prohlížeče. Zde je nutné vložení souboru *config.php*, přes který probíhá přihlášení do *MySQL* databáze a získávání či odesílání dat. Pro určení *meta tagů* na stránce, načtení *CSS* a *JavaScript* souborů je potřeba definovat soubor *header.php* pomocí funkce *require_once*, který tyto soubory definuje. Poté následuje obsah souboru *index.php*, kterým je přihlašovací formulář, který obsahuje tag *FORM* a v něm se nachází jednotlivé *INPUTY* pro

zadávání přihlašovacích údajů. Na konci souboru se uvádí zápatí formou souboru *footer.php*, kde jsou uvedeny převážně JavaScriptové soubory, které se spouští při načtení stránky.

Ukázka kódu pro vložení souborů s konfigurací a html s meta tagy v souboru *index.php*:

```
require_once "inc/config.php";  
require_once "inc/header.php";
```

V souboru *header.php* jsou obsaženy všechny důležité *html meta tagy* a cesty na CSS soubory, které slouží pro naformátování stránky. Tyto soubory autor vložil do složky *assets*, která obsahuje všechny CSS soubory, ale i všechny *JavaScriptové* soubory. CSS soubory jsou rozděleny podle toho, zdali se jedná o určitou knihovnu nebo jsou to styly vytvořené speciálně pro tento projekt. Pro vlastní styly se používá soubor *style.css*. Pomocí těchto souborů je pak vytvořen celý přihlašovací formulář, který využívá především knihovny zvané *Bootstrap* pro správné rozložení elementů na stránce a jejich obarvení.



Obr. 10: Zobrazení přihlašovacího formuláře v prohlížeči, zdroj: vlastní zpracování

Poté autor vytváří základní administrační rozhraní, ve kterém jsou vidět informace o aktivitách serverů. Tato webová stránka je vytvořena formou jedné stránky, na které se vyskytuje seznam serverů a sledování jejich aktivity. Nejprve je nutné vytvořit soubor s napojením na databázi MySQL, kterou autor vytvořil v dřívějších krocích. Vytvořený soubor s názvem *DB.php* se nachází ve složce *inc* a podsložce *classes*, která definuje jednotlivé PHP funkce, které se využívají v interakci mezi prohlížečem a serverem. Pro připojení k databázi je nutné definovat proměnnou *\$con*, do které se poté vkládají přihlašovací údaje, které autor nastavil v rámci

webových služeb. Mezi povinné údaje pro připojení k databázi patří zadání adresu serveru, port, název databáze, jméno uživatele a heslo.

Ukázka PHP kódu pro připojení systému k MySQL databázi:

```
$con = new PDO( 'mysql:host=mysql;port=3306;dbname=archiv', 'root', 'root' );
```

Po úspěšném připojení systému k databázi může autor začít s vytvářením grafického rozhraní administrace. V horní části administrace se nachází lišta, která je naformátovaná pomocí jazyka CSS. Je zde vytvořena třída s názvem „horni-lista“, která má nastavenou výšku 70 pixelů a barvu pozadí jako šedivou.

Ukázka CSS kódu v souboru *assets/style.css*:

```
.horni-lista {  
    color: white;  
    height: 70px;  
    background-color: gray;  
}
```

V pravém horním rohu této lišty se nachází tlačítko pro odhlášení ze systému. Po jejímž kliknutí bude uživatel v prohlížeči přesměrován na soubor *logout.php*, který se nachází v kořenovém adresáři *PHP* souborů. Soubor *logout.php* slouží pro odstranění všech údajů, které jsou nastaveny po přihlášení uživatele a odpojí tak uživatele z daného sezení, v anglickém jazyce „session“.

Souhrn serverů a jednotlivých aktivit je rozdělen na dvě části obrazovky, kdy je toto rozdělení vytvořeno pomocí CSS tříd s názvy „col-6“, což znamená, že zabírá šest sloupců z celkových dvanácti, které jsou vytvořené pomocí mřížky definované v CSS knihovně *Bootstrap*. Každá z těchto částí je vytvořena pomocí tabulky, tedy *html* tagu *table*, která sestaví v prohlížeči jednoduchou grafickou tabulku s vloženými daty. Pro výpis jednotlivých dat do tabulky je používán *SQL* dotaz skrze programovací jazyk *PHP*. V případě zobrazení všech logů, které jsou ukládány *NodeJS* skriptem do databáze, je potřeba vytvořit *SQL* dotaz, který vypíše všechny řádky z tabulky *logs* seřazené podle nejnovějšího. V jazyce *PHP* je vypíšeme použitím proměnné *\$con* a následné vyvolání funkce *prepare* s vloženým *SQL* dotazem.

Ukázka PHP kódu pro vypsání všech dat z tabulky *logs*:

```
$logList = $con->prepare("SELECT * FROM `logs` ORDER BY id DESC LIMIT 10");
```

Poté je nutné tento SQL dotaz vyvolat funkcí *execute* a následně data načíst pomocí funkce *fetchAll*. Pakliže chceme, aby se vypsaly všechny řádky, tak musíme využít cyklu *foreach*, který opakuje vyvolání jednotlivých sloupců. Do tabulky pak data vkládáme formou *html* řádků pojmenovaných tagem *TR* a jednotlivých sloupců tagem *TD*. V této tabulce je vytvořeno celkem pět sloupců s názvy „id“, „IP serveru“, „Zpráva“, „Dostupnost“ a „Čas kontroly“.

Ukázka PHP kódu pro zobrazení všech dat z tabulky *logs*:

```
foreach ($logs as $log) {
    echo '<tr><td>' . $log['id'] . '</td><td>' . $log['server_id'] . '</td><td>' .
    $log['message'] . '</td><td>' . $log['status'] . '</td><td>' . $log['date']
    . '</td></tr>';
}
```

Souhrn všech sledovaných serverů autor práce umístil do pravé části obrazovky. Tento souhrn je opět vytvořen formou *html* tabulky, která má přiřazenou CSS třídu s názvem „table-striped“, která každému lichému řádku vytvoří šedé pozadí. Pro zobrazení všech serverů, tedy všech záznamů z tabulky *servers*, se používá podobně jako u vypsání všech logů *SQL* dotaz *SELECT*. Pro zobrazení dostupnosti serveru vytvořil autor *PHP* funkci s pojmenováním *showStatus*, která získá požadované ID serveru a poté jej vloží jako filtr pro vypsání dat skrze *SQL* dotaz. Poté funkce dotaz vykoná a vrátí daný status, který se váže k danému serveru.

Ukázka PHP kódu funkce pro zobrazení aktuální dostupnosti serveru:

```
function showStatus($serverId) {
    $con = DB::getConnection();
    $logList = $con->prepare("SELECT * FROM `logs` WHERE server_id = $serverId ORDER
    BY date DESC LIMIT 1");
    $logList->execute();
    $logs = $logList->fetchAll();
    foreach ($logs as $log) {
        return $log['status'];
    }
}
```

Pro zobrazení dostupnosti serveru se autor rozhodl vytvořit grafický obrazec v podobě tečky, která se zbarvuje do dané barvy podle toho, jestli je určitý server aktivní či nikoliv. Celý tento obrazec je vytvořen pomocí jazyka *CSS*, kdy autor

definuje jeho výšku, šířku, zakulacení a pozadí v *css* třídě pojmenované *dot*. Pro aktivní server se autor rozhodl použít zelenou barvu tečky a pro nedostupný červenou.

Ukázka *CSS* kódu pro vytvoření grafické podoby dostupnosti serveru:

```
.dot--online {
    height: 25px;
    width: 25px;
    background-color: #02bb00;
    border-radius: 50%;
    display: inline-block;
}
.dot--offline {
    height: 25px;
    width: 25px;
    background-color: #bb000b;
    border-radius: 50%;
    display: inline-block;
}
```

Pomocí *PHP* podmínky potom autor v *html* tabulce určuje, zdali se má použít zelená či červená tečka informující o dostupnosti serveru. V použité podmínce *IF – ELSE* je definováno, že pokud daný server obsahuje ve sloupci „status“ v tabulce *logs* hodnotu 1, znamená to, že je aktivní a zobrazí se zelená tečka. Pakliže se jedná o hodnotu 0, tak se zobrazí tečka červená. Uvnitř podmínky se vždy mění pouze jednotlivé řádky, kde jsou definovány *css* třídy pro zobrazení barevných teček.

Ukázka *PHP* podmínky pro zobrazení dostupnosti serveru:

```
if ( showStatus($serverId) === 1) {
echo '<tr><td>' . $server['id'] . '</td><td>' . $server['ip'] . '</td><td>' .
$server['name'] . '</td><td>' . $server['description'] . '</td><td><div
class="dot--online"></div></td></tr>';
} else {
echo '<tr><td>' . $server['id'] . '</td><td>' . $server['ip'] . '</td><td>' .
$server['name'] . '</td><td>' . $server['description'] . '</td><td><div
class="dot--offline"></div></td></tr>';
}
```

Autor se také rozhodl použít automatickou aktualizaci celé administrační stránky bez nutnosti obnovovat stránku ve webovém prohlížeči. Toho je docíleno pomocí JavaScriptové funkce *location.reload*, která provede obnovu aktuální webové stránky v prohlížeči. Dále je ještě nutné definovat prodlevu dosaženou díky funkci *setTimeout*, kdy její hodnotu autor nastavil na hodnotu 10000, tedy deset sekund.

Ukázka JavaScriptového kódu v souboru *main.js*:

```
setTimeout(function(){ location.reload(); }, 10000);
```

Na následujícím obrázku je vidět konečná verze webové aplikace sloužící pro administraci archivních serverů.

Administrace archivních serverů Odhlast se

Nástěnka

Přihlášený email: admin@uhk.cz, přihlášení v 2019-02-25 19:09:44

#	IP serveru	Zpráva	Dostupnost	Čas kontroly
80300	1	192.168.31.89 je aktivní	1	2019-07-15 14:51:20
80299	2	192.168.31.203 je aktivní	1	2019-07-15 14:51:20
80298	4	192.168.0.101 je aktivní	1	2019-07-15 14:51:20
80297	1	192.168.31.89 je aktivní	1	2019-07-15 14:51:10
80296	2	192.168.31.203 je aktivní	1	2019-07-15 14:51:10
80295	4	192.168.0.101 je aktivní	1	2019-07-15 14:51:10
80294	1	192.168.31.89 je aktivní	1	2019-07-15 14:51:00
80293	2	192.168.31.203 je aktivní	1	2019-07-15 14:51:00
80292	4	192.168.0.101 je aktivní	1	2019-07-15 14:51:00
80291	1	192.168.31.89 je aktivní	1	2019-07-15 14:50:50

#	IP serveru	Jméno	Popis	Aktuální stav
1	192.168.31.89	Mobil	Xiaomi mi 8	●
2	192.168.31.203	Zarovka	Xiaomi Bulb	●
4	192.168.0.101	Notebook Lenovo W530	Osobní notebook	●

Obr. 11: Zobrazení vytvořené administrace ve webovém prohlížeči, zdroj: vlastní zpracování

Závěr

Cílem teoretické části bylo seznámit čtenáře s vytvářením moderních webových aplikací. V této části je vysvětleno, jak funguje virtualizace serverů pomocí systému Docker a jakým způsobem funguje webový server a webová aplikace. Dále bylo cílem seznámit čtenáře i se základními pojmy spojenými s archivem a jeho historií a dále popsat fungování od minulosti až po současnost.

Cíl teoretické části byl úspěšně splněn. Čtenáře se podařilo seznámit s fungováním webového serveru, jaké technologie se používají při tvorbě webové aplikace a jaké jsou moderní trendy při tvorbě webových aplikací. Tyto technologie lze nalézt v kapitole 1 a 2. V třetí kapitole tato práce také seznamuje čtenáře s tím, jakou činnost vykonávají archivy, jejich typy a jak funguje spisová služba.

Cílem praktické části bakalářské práce bylo vytvoření systému pro sledování stavu jednotlivých počítačových zařízení v rámci počítačové sítě. Systém měl umožňovat v případě nedostupnosti nějakého serveru v daném archivu automatické odeslání notifikace na emailovou adresu správce počítačové sítě archivu. Autor měl za úkol vytvořit přehledný a jednoduchý systém pro sledování serverů v prostředí webového prohlížeče. Celý tento systém by měl fungovat v rámci virtualizačního programu Docker a měl by tak nabízet jeho jednoduchou implementaci mezi servery.

Návrh a vytvoření systému se zdařil a autor práce vytvořil systém, který je vcelku jednoduše možný přenést pomocí virtualizačního nástroje Docker na běžící systém umístěný v archivu. Součástí systému je vytvořená databáze, která ukládá data z chování serverů a jejich jednotlivý popis. Systém se skládá ze dvou částí, jedna část je samotný skript, který se v reálném čase stará o sledování dostupnosti serverů a zasílání notifikací, a druhá část je webová aplikace, přes kterou může administrátor sledovat aktivitu serverů přes internetový prohlížeč. Postup vytváření projektu je zaznamenán v kapitole 6 Vývoj webové aplikace.

Jedním z hlavních cílů této práce bylo osvojení si znalostí z vývoje webových aplikací a systému Docker a rozšíření si znalostí o systémech využívaných v archivech. Autor této práce měl již nějaké zkušenosti s tvorbou webových stránek, ale žádné zkušenosti s virtualizační službou Docker. Praktická část práce

tedy autora obohatila o spoustu nových znalostí a zkušeností. Vytvoření praktické části bylo z časového hlediska náročnější oproti původnímu odhadu, ale nakonec byl cíl vytvoření celého fungujícího systému splněn.

Tato práce by mohla pomoci technickým pracovníkům archivů ve sledování archivních serverů či počítačů a zamezit tak velkým výpadkům, které se v současnosti na hodně pracovištích dějí.

Použité zdroje a literatura

[1] MATYSKA, Luděk. Virtualizace výpočetního prostředí. Zpravodaj ÚVT MU. 2006, roč. 17, č. 2, s. 9-11. DOI: ISSN 1212-0901. Dostupné z: <http://webserver.ics.muni.cz/bulletin/articles/540.html>

[2] Co je to virtualizace? Oldanygroup.cz [online]. [cit. 2019-03-29]. Dostupné z: <http://www.oldanygroup.cz/virtualizace-vmware-zakladni-informace-9/>

[3] POMAZAL, Jiří. Virtualizace v kostce. Systemonline.cz [online]. [cit. 2019-03-29]. Dostupné z: <https://www.systemonline.cz/clanky/virtualizace-v-kostce.htm>

[4] Webový server, Wikipedia [online]. [cit. 2018-04-30]. Dostupné z: https://cs.wikipedia.org/wiki/Webov%C3%BD_server

[5] Jaké jsou druhy serveru. Jaknahosting.cz [online]. 13. srpna 2017 [cit. 2019-03-29]. Dostupné z: <http://jaknahosting.cz/jake-jsou-druhy-serveru/>

[6] Docker overview. Docker docs [online]. [cit. 2018-04-30]. Dostupné z: <https://docs.docker.com/engine/docker-overview/>

[7] KONICK, Sinbad. Top 5 Server Operating Systems. Slideshare.net [online]. 3. prosince 2016 [cit. 2019-03-29]. Dostupné z: <https://www.slideshare.net/BlackDetah/top-5-server-operating-systems>

[8] ŠTOURÁČOVÁ, Jiřina. Archivnictví. Brno: Masarykova univerzita, 2013. ISBN 978-80-210-6512-3. Dostupné také z: <https://digilib.phil.muni.cz/data/handle/11222.digilib/128576/monography.pdf>

[9] Co je archiv. Cesarch.cz [online]. Česká archivní společnost [cit. 2019-03-29]. Dostupné z: <http://cesarch.cz/archivnictvi/co-je-archiv/>

- [10] BARTOŠ Josef – CHOBOT, Karel. Úvod do archivnictví pro historiky. Olomouc, 2000. 97 s. ISBN 80 - 244- 0142 – 8.
- [11] NIXON, Robin. Learning PHP, MySQL & JavaScript. Květen 2018. ISBN 978-1-491-97891-7.
- [12] NOLAN, Ashley Watson. Launching the Front-End Tooling Survey 2019 [online]. 8. duben 2019 [cit. 2019-04-14]. Dostupné z: <https://medium.freecodecamp.org/launching-the-front-end-tooling-survey-2019-4cb2b72f0b42>
- [13] Spisová a archivní služba ve státní správě, samosprávě a v podnikatelské sféře:3. aktualizované a přepracované vydání. Praha: Linde, 2005, s. 13. ISBN 80-7201-549-4.
- [14] HLEDÍKOVÁ-Zdeňka, JANÁK-Jan, DOBEŠ-Jan. Dějiny správy v českých zemích: od počátku státu po současnost. Praha: Lidové noviny, 2005, s. 570. ISBN: 978-80-7106-906-5.
- [15] *Příruční slovník naučný III. díl.* Praha: Nakladatelství Československé akademie věd, 1966, s. 830.

[16] *Ottův slovník naučný: ilustrovaná encyklopaedie obecných vědomostí. Dvacátýprvý díl.* V Praze: J. Otto, 1904, s. 407.

[17] *Ottův slovník naučný: ilustrovaná encyklopaedie obecných vědomostí. Druhý díl.* V Praze: J. Otto, 1889, s. 691.

[18] Jak Psát Web [online]. [cit. 2019-04-16]. Dostupné z: <https://www.jakpsatweb.cz/php/moznosti-php.html>

[19] Bach systems, s.r.o.: Bach Vademecum [online]. [cit. 2019-04-23]. Dostupné z: <http://www.bach.cz/produkty/archivni-vademecum/>

[20] CASTRO, Elizabeth a Bruce HYSLOP. HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. ISBN isbn-978-80-251-3733-8.

[21] ČÁPKA, David. 1. díl – Úvod do CSS frameworku Bootstrap. Itnetwork.cz [online]. itnetwork.cz [cit. 2018-04-30]. Dostupné z: <https://www.itnetwork.cz/htmlcss/bootstrap/uvod-do-css-frameworku-bootstrap>

[22] WordPress.org [online]. [cit. 2019-04-25]. Dostupné z: <https://wordpress.org/about/features>

[23] BRIAN, P. Hogan. HTML5 a CSS3, Výukový kurz webového vývojáře. Brno: Computer Přes, 2011. ISBN 978-80-251-3576-1

[24] STÁTNÍ OBLASTNÍ ARCHIV V PRAZE. Příloha č. 2 - Popis aplikací Státního oblastního archivu v Praze: Příloha č. 2 k zasedávací dokumentaci č.j.: 7773/30-2011. 2011. Dostupné také z: http://www.soapraha.cz/documents/praha/20120704170804-Priloha_c._2_k_zadavaci_dokumentaci.pdf

[25] KUNT, Miroslav a Tomáš LECHNER, 2015. Spisová služba. 1. vyd. Praha: Leges, 400 s. ISBN 978-80-7502-083-3

[26] Produkty. GORDIC, [online]. Jihlava: GORDIC, [cit. 2016-04-05]. Dostupné z: <https://www.gordic.cz/produkty/>

Seznam příloh

Příloha 1 – Příložené CD	75
--------------------------------	----

Příloha 1 – Přiložené CD

Přiložené CD obsahuje:

- kód vývojového prostředí s webovým projektem,
- PDF verzi bakalářské práce.