

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky

Vývoj aplikace pro dokumentaci sítě SAN

Bc. Radek Tykva

Diplomová práce
2019

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Radek Tykva**
Osobní číslo: **E17542**
Studijní program: **N6209 Systémové inženýrství a informatika**
Studijní obor: **Informatika ve veřejné správě**
Název tématu: **Vývoj aplikace pro dokumentaci sítě SAN**
Zadávající katedra: **Ústav systémového inženýrství a informatiky**

Zásady pro vypracování:

Cílem práce je vývoj webové aplikace pro dokumentaci sítě SAN v topologii Switched Fabric. Na základě charakteristiky možných konfigurací tohoto typu sítě bude navržena a vytvořena aplikace pro potřeby správce sítě SAN. Využity budou nástroje datového modelování a nástroje pro vývoj webových aplikací.

Osnova:

- Základní pojmy související se zpracovávanou problematikou.
- Tvorba modelů.
- Vývoj aplikace.

Rozsah grafických prací:

Rozsah pracovní zprávy: **cca 50 stran**

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

Seznam odborné literatury:

BORONCZYK, Tim. *MySQL okamžitě*. Brno: Computer Press, 2016. ISBN 978-80-251-4737-5.

ČEGAN, Lukáš. *Vývoj webových aplikací v PHP a NetBeans*. Pardubice: Univerzita Pardubice, 2015. ISBN 978-80-7395-858-9.

ŠIMONOVÁ, Stanislava. *Databázové systémy I*. Pardubice: Univerzita Pardubice, 2013. ISBN 978-80-7395-702-5.

TATE, Jon, Pall BECK, Hector Hugo IBARRA, Shanmuganathan KUMARAVEL a Libor MIKLAS. *Introduction to Storage Area Networks*. IBM International Technical Support Organization, 2012. ISBN 0738441430.

WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL*. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1.

Vedoucí diplomové práce: **doc. Ing. Stanislava Šimonová, Ph.D.**
Ústav systémového inženýrství a informatiky

Datum zadání diplomové práce: **3. září 2018**

Termín odevzdání diplomové práce: **30. dubna 2019**

doc. Ing. ~~Romana Provanžlková~~, Ph.D.
děkanka

L.S.

doc. Ing. Pavel Petr, Ph.D.
vedoucí ústavu

V Pardubicích dne 3. září 2018

PROHLÁŠENÍ

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Beru na vědomí, že v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, a směrnicí Univerzity Pardubice č. 9/2012, bude práce zveřejněna v Univerzitní knihovně a prostřednictvím Digitální knihovny Univerzity Pardubice.

V Pardubicích dne 25. 11. 2019

Radek Tykva

PODĚKOVÁNÍ:

Touto cestou bych rád poděkoval své vedoucí práce doc. Ing. Stanislavě Šimonové, Ph. D., za její odbornou pomoc, cenné rady, trpělivost a poskytnuté materiály, které mi pomohly při zpracování diplomové práce.

V neposlední řadě bych chtěl poděkovat mé rodině a mým blízkým za podporu při tvorbě této práce a celém mém studiu.

ANOTACE

Tato práce přibližuje problematiku dedikovaných datových sítí SAN (Storage Area Network) postavených na technologii Fibre Channel a věnuje se vývoji webové aplikace pro dokumentaci sítě typu Switched Fabric. Aplikace má za úkol poskytovat přehledné informace o konfiguraci dokumentované SAN. Vývoj aplikace je realizován tvorbou řady prototypů. Úvodní prototypy jsou vytvořeny pro konkrétní prostředí, další pro obecné nasazení pro síť typu Switched Fabric. Pro vývoj aplikace jsou použity technologie PHP, JavaScript a MySQL.

KLÍČOVÁ SLOVA

Datové sítě SAN, Switched Fabric, Webová aplikace

TITLE

Application development for the Storage Area Network documentation

ANNOTATION

This paper approaches the issue of dedicated data network SAN (Storage Area Network) based on Fiber Channel technology and deals with the development of a web application for Switched Fabric network documentation. The application is designed to provide clear configuration information documented by SAN. Development of the application is realized by creating a number of prototypes. Introductory prototypes are created for specific environments, others for general use for Switched Fabric network. PHP, JavaScript and MySQL technologies are used for application development.

KEYWORDS

Storage Area Network, Switched Fabric, Web application

OBSAH

ÚVOD	11
1 STORAGE AREA NETWORK (SAN)	12
1.1 FUNKCE SAN	12
1.2 PŘÍNOSY SAN	12
2 SWITCHED FABRIC STORAGE AREA NETWORK.....	15
2.1 SINGLE SWITCH TOPOLOGIE	16
2.2 CASCADE/RING TOPOLOGIE.....	17
2.3 MESH TOPOLOGIE	18
2.4 VRSTVENÉ TOPOLOGIE.....	18
2.5 KONFIGURACE SWITCHED FABRIC	21
2.6 DÍLČÍ SOUHRN	22
3 NÁVRH POSTUPU PRÁCE	23
3.1 ROZHODNUTÍ O POUŽITÍ APLIKACE A JEJÍM FORMÁTU.....	23
3.2 ROZHODNUTÍ O ZPŮSOBU VÝVOJE APLIKACE A VLASTNÍ VÝVOJ.....	23
3.3 VYHODNOCENÍ POUŽITÉHO POSTUPU VÝVOJE APLIKACE.....	23
4 ÚČEL APLIKACE A TECHNICKÉ PROSTŘEDKY VÝVOJE APLIKACE	24
4.1 ÚČEL APLIKACE	24
4.2 FORMÁT APLIKACE A TECHNICKÉ PROSTŘEDKY VÝVOJE APLIKACE	24
4.2.1 <i>JavaScript</i>	25
4.2.2 <i>PHP</i>	25
4.2.3 <i>MySQL</i>	25
4.3 DÍLČÍ SOUHRN	26
5 NÁVRH A VÝVOJ APLIKACE.....	27
5.1 PROTOTYP 1	28
5.1.1 <i>Datový model</i>	29
5.1.2 <i>Aplikace – prototyp 1</i>	38
5.1.3 <i>Menu aplikace</i>	39
5.1.4 <i>Napojení na databázi</i>	40
5.1.5 <i>Okna aplikace</i>	41
5.2 PROTOTYP 2	48
5.2.1 <i>Doplňkové požadavky</i>	48
5.2.2 <i>Změny v datovém modelu</i>	48
5.2.3 <i>Změny v aplikaci</i>	50
5.3 PROTOTYP 3	54
5.3.1 <i>Analýza podtypů</i>	54
5.3.2 <i>Změny v datovém modelu</i>	54
5.3.3 <i>Změny v aplikaci</i>	56
5.4 DÍLČÍ SOUHRN	57
6 VYHODNOCENÍ APLIKACE A POUŽITÉHO POSTUPU VÝVOJE	58
6.1 VYUŽITÍ, VÝZNAM APLIKACE	58
6.2 METODA TŘÍ PROTOTYPŮ	58
7 ZÁVĚR	60
POUŽITÁ LITERATURA A ZDROJE	61
PŘÍLOHY	63

SEZNAM TABULEK

<i>Tabulka 1: Switched Fabric SAN a její varianty topologie</i>	20
<i>Tabulka 2: Technické prostředky použité pro vývoj aplikace</i>	26
<i>Tabulka 3: Struktura databázové tabulky Fabric</i>	32
<i>Tabulka 4: Struktura databázové tabulky Device</i>	33
<i>Tabulka 5: Struktura databázové tabulky PortDevice</i>	33
<i>Tabulka 6: Struktura databázové tabulky Zoneset</i>	34
<i>Tabulka 7: Struktura databázové tabulky Zone</i>	34
<i>Tabulka 8: Struktura databázové tabulky Nod</i>	35
<i>Tabulka 9: Struktura databázové tabulky PolZoneset</i>	36
<i>Tabulka 10: Struktura databázové tabulky PolZone</i>	37
<i>Tabulka 11: Seznam doplňkových požadavků na aplikaci</i>	48
<i>Tabulka 12: Struktura databázové tabulky Device</i>	49
<i>Tabulka 13: Seznam rozšiřujících požadavků na aplikaci</i>	54
<i>Tabulka 14: Struktura databázové tabulky Nod</i>	55
<i>Tabulka 15: Struktura nové databázové tabulky PolLoop</i>	56
<i>Tabulka 16: Vývoj aplikace pomocí tří prototypů</i>	58

SEZNAM OBRÁZKŮ

<i>Obrázek 1: Konsolidace úložišť</i>	13
<i>Obrázek 2: Příklad topologie Fibre Channel SAN</i>	15
<i>Obrázek 3: Topologie Fibre Channel SAN</i>	16
<i>Obrázek 4: Topologie Single Switch</i>	17
<i>Obrázek 5: Topologie Cascade a Ring</i>	17
<i>Obrázek 6: Topologie Mesh</i>	18
<i>Obrázek 7: Topologie Edge Core</i>	19
<i>Obrázek 8: Topologie Edge Core Edge</i>	20
<i>Obrázek 9: Cisco MDS 9148 Fabric Switch</i>	21
<i>Obrázek 10: Příklad zónování</i>	22
<i>Obrázek 11: Model postupu při vývoji aplikace</i>	27
<i>Obrázek 12: Síť SAN společnosti Povodí Labe, státní podnik</i>	28
<i>Obrázek 13: Koncept tří úrovní</i>	29
<i>Obrázek 14: Use Case diagram</i>	29
<i>Obrázek 15: Relační model</i>	30
<i>Obrázek 16: Relační model prototypu 1</i>	31
<i>Obrázek 17: Přehled tabulek datového modelu</i>	32
<i>Obrázek 18: Struktura databázové tabulky Nod</i>	35
<i>Obrázek 19: Vazby databázové tabulky Nod</i>	36
<i>Obrázek 20: Struktura databázové tabulky PolZone</i>	37
<i>Obrázek 21: Vazby databázové tabulky PolZone</i>	38
<i>Obrázek 22: Menu aplikace – úvodní okno aplikace</i>	39
<i>Obrázek 23: Kód menu aplikace v JavaScriptu</i>	40
<i>Obrázek 24: Úvodní okno Fabrics - kód pro práci s databází</i>	41
<i>Obrázek 25: Úvodní okno aplikace</i>	41
<i>Obrázek 26: Výpis koncových zařízení definovaných ve Fabric_SAN1</i>	42
<i>Obrázek 27: Úprava záznamu</i>	42
<i>Obrázek 28: Okno Devices</i>	43
<i>Obrázek 29: Okno detailu vybraného device</i>	44
<i>Obrázek 30: Ukázka kódu skriptu device_view</i>	44

<i>Obrázek 31: Okno detailu vybraného nodu</i>	45
<i>Obrázek 32: Okno detailu vybrané zóny</i>	45
<i>Obrázek 33: Ukázka kódu skriptu zones_view</i>	46
<i>Obrázek 34: Okno detailu vybraného zonesetu</i>	47
<i>Obrázek 35: Ukázka kódu skriptu zoneset_view</i>	47
<i>Obrázek 36: Relační model prototypu 2</i>	49
<i>Obrázek 37: Úvodní obrazovka prototypu 2</i>	50
<i>Obrázek 38: Okno pro vyhledávání</i>	50
<i>Obrázek 39: Okno s výsledkem vyhledávání podle WWN</i>	51
<i>Obrázek 40: Kód skriptu pro vyhledávání podle WWN</i>	51
<i>Obrázek 41: Okno Device_View s doplněným atributem</i>	52
<i>Obrázek 42: Okno Nodes_View s doplněným odkazem In Zones</i>	52
<i>Obrázek 43: Zdrojový kód</i>	53
<i>Obrázek 44: Okno Nodes_View_Zone</i>	53
<i>Obrázek 45: Relační model prototypu 3</i>	55
<i>Obrázek 46: Okno Nodes v prototypu 3</i>	56
<i>Obrázek 47: Okno detailu vybraného nodu typu smyčka</i>	57
<i>Obrázek 48: Model – Metoda Tří prototypů</i>	59

SEZNAM POUŽITÝCH ZKRATEK A POJMŮ

FC (Fibre Channel)

Technologie propojení výkonných úložných zařízení a serverů optickými kanály, která umožňuje bezpečně přenášet data vysokou rychlostí na omezenou vzdálenost (max. 10 km) a to v rámci datového centra typicky odděleného od podnikové sítě.

LAN (Local Area Network)

Lokální (místní) počítačová síť, zpravidla v rámci firmy, budovy apod.

SAN (Storage Area Network)

Síť nezávislá na podnikovém intranetu, specificky zaměřená na propojení úložných zařízení. SAN je optimalizovaná na sběr, ukládání a výběr bloků dat. SAN používají otevřené technologie jako Fibre Channel, nebo řešení orientovaná na IP (iSCSI, FCIP).

SPOF (Single Point Of Failure)

Prvek nebo část systému, jehož vyřazení způsobí nefunkčnost celého systému.

WAN (Wide Area Network)

Počítačová síť pokrývající geograficky rozlehlé území (překračující hranice města, regionu).

WWN (World Wide Name)

Unikátní 64-bitová adresa (obdoba MAC adresy) jednoznačně identifikující zařízení připojené do SAN pomocí HBA.

ÚVOD

Informační a komunikační technologie (ICT – Information and Communication Technologies) zahrnují veškeré technologie pro komunikaci a práci s informacemi. Nedílnou součástí ICT jsou počítačové sítě. Počítačové sítě dělíme na jednotlivé typy podle různých kritérií – velikosti sítě, její topologie, použité technologie spojení, účelu apod. Jednou z významných množin počítačových sítí jsou sítě datové – SAN (Storage Area Network). Datové sítě jsou fyzicky oddělené od běžných komunikačních počítačových sítí typu LAN, WAN. Slouží výhradně ke čtení, zápisu a přesouvání bloků dat mezi zařízeními určenými k ukládání dat a servery. Konfigurace datových sítí se od konfigurace komunikačních sítí typu LAN velmi liší. Stejně tak identifikace zařízení připojených do sítě.

Ve firmě Povodí Labe, státní podnik, pracuji mimo jiné i jako administrátor firemní datové sítě. Protože konfigurace SAN se děje na aktivních prvcích sítě (SAN switche), informace o aktuálním nastavení a stavu sítě lze získat pouze administrátorským přístupem do managementu těchto prvků. Tyto informace ovšem občas potřebují i kolegové z odboru informatiky, případně i další uživatelé (např. vývojáři). Specializovaný software, který by administrátorům sítě a dalším uživatelům zprostředkoval informace o konfiguraci SAN na trhu chybí, resp. bývá spojen s dodávkou velmi drahých zařízení. Software používaný pro monitorování a dokumentaci komunikačních sítí LAN typu Zabbix, LibreNMS či NetXMS nelze použít k požadovanému účelu pro specifické postupy konfigurace a specifické atributy SAN (zónování, WWN apod.).

Cílem práce je vývoj webové aplikace pro dokumentaci sítě SAN v topologii Switched Fabric. Na základě charakteristiky možných konfigurací tohoto typu sítě bude navržena a vytvořena aplikace pro potřeby správce sítě SAN. Využity budou nástroje datového modelování a nástroje pro vývoj webových aplikací.

1 STORAGE AREA NETWORK (SAN)

Storage area network (SAN) představuje dedikovanou síť (oddělenou od komunikační sítě LAN, WAN), propojující veškerá zařízení určená k ukládání dat (disková pole, páskové knihovny a jiná zálohovací zařízení) a servery. [9]

1.1 Funkce SAN

Tím, že je SAN oddělena od komunikační sítě podniku a funguje jako sekundární síť, ulevuje hlavní síti, protože veškeré objemné přesuny dat se odehrávají mezi systémy připojenými k SAN. Zátěž je rozložena mezi více serverů s různým úložným prostorem, případně se realizuje přímo na páskové jednotky (server-less), takže neexistuje jediné problematické místo potenciálního selhání jako v případě přímo připojených úložných jednotek. [8]

SAN je optimalizovaná na sběr, ukládání a výběr bloků dat. Vznikla hlavně kvůli narůstajícím potřebám na zabezpečení a konsolidaci dat. Díky poměrně vysokým pořizovacím nákladům se SAN budují hlavně ve větších společnostech (bankovní sektor, automobilový průmysl, média), které vyžadují vysokou dostupnost svých služeb, rychlé odezvy v transakčně orientovaných aplikacích a co největší škálovatelnost (rozšiřitelnost). V poslední době se však díky klesající ceně začínají malé SAN rozšiřovat i do stále menších společností. Spíše než problém s vysokou cenou jde tedy o problém přesvědčit management o potřebnosti takové investice, což při obecné neochotě firem investovat do informačních technologií není jednoduché. [5]

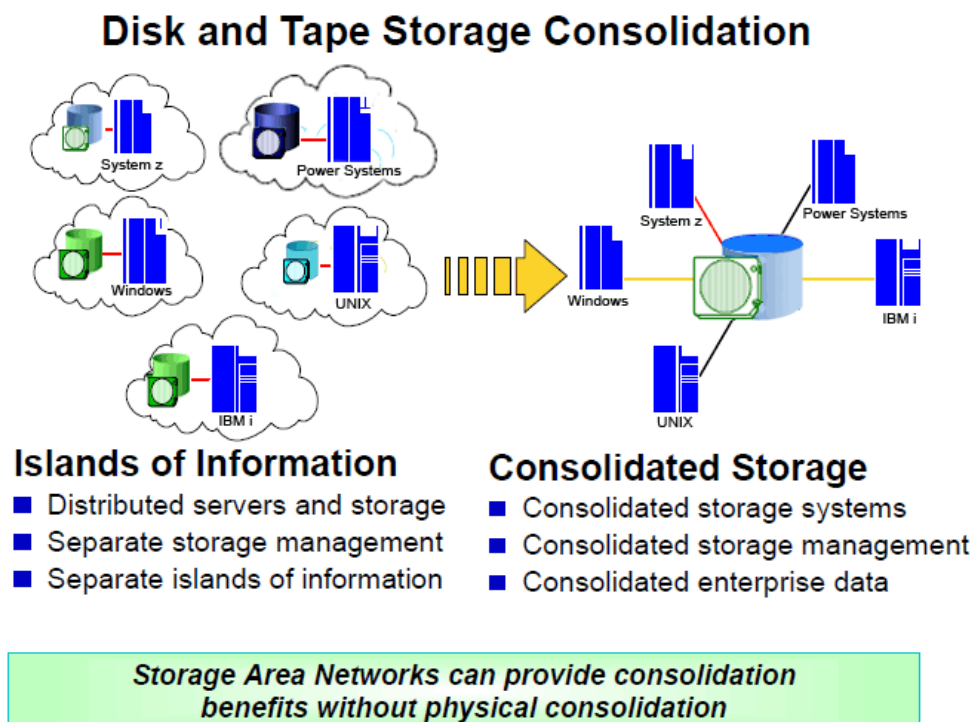
1.2 Přínosy SAN

SAN nejenže vyžaduje citelně vyšší investice, ale v neposlední řadě i odpovídajícím způsobem proškolený personál. Jde totiž o několikanásobně složitější záležitost a odpovědní pracovníci se musí bez problémů orientovat v její topologii, terminologii i ve všech charakteristických vlastnostech. Současně však nabízí i „něco navíc“ a právě tyto bonusy by měly výše uvedené komplikace dostatečně vykompenzovat. [10]

Konsolidace

Proces konsolidace zahrnuje tři aspekty. Jednak je to konsolidace datových úložišť a zálohovacích zařízení jako takových: různá zařízení zpravidla svázaná se servery distribuovanými v různých lokalitách nahradí jeden systém s jednotnou platformou. S tím

souvisí konsolidace storage managementu. Třetím aspektem je konsolidace dat a efektivnější získávání informací pro řízení společnosti a manažerské aplikace (viz Obrázek 1). [5]



Obrázek 1: Konsolidace úložišť

Zdroj: [5]

Virtualizace

Virtualizace je jednoduchý nástroj pro komunikaci mezi servery a diskovými poli. Virtualizace umožňuje zpřístupnit serverům veškerou diskovou kapacitu v rámci jediného prostoru – SAN. Službu virtualizace obstarává virtualizační server (controller), který je zpravidla v systému již implementován. Slouží k jednoduchému poskytování diskového prostoru serverům prostřednictvím virtuálních logických jednotek vytvořených napříč diskovým polem a zároveň spravuje veškerá datová úložiště v síti. Tato funkcionalita umožňuje pracovat s různými logickými jednotkami na různých diskových polích, pracovat s daty a přesouvat je, rozšiřovat logické jednotky o další kapacity se zapojením nových datových úložišť. [7]

Automatizace

Činnosti a úlohy, jejichž smyslem je ochrana dat a zajištění jejich vysoké dostupnosti, lze v moderní SAN zpravidla plně zautomatizovat. A to ať se jedná o automatické zpracování události výpadku některého z prvků systému (varovné hlášení, nahrazení vadného prvku

náhradním, přepnutí na redundantní cestu apod.) nebo automatizaci úloh typu replikace dat, zálohování na pásku a jiné. Pokud jsou například denní úlohy prováděny automaticky, mohou administrátoři věnovat více času složitějším úlohám nebo se mohou věnovat úlohám pro danou společnost unikátním. [5]

Integrace

Integrace SAN do informačního systému zjednodušuje správu a zlepšuje bezpečnost. Zabezpečený přístup všech serverů ke všem datům zvyšuje schopnost informačního systému rychle obsloužit veškeré požadavky uživatelů. [5]

Bezpečnost a vysoká dostupnost

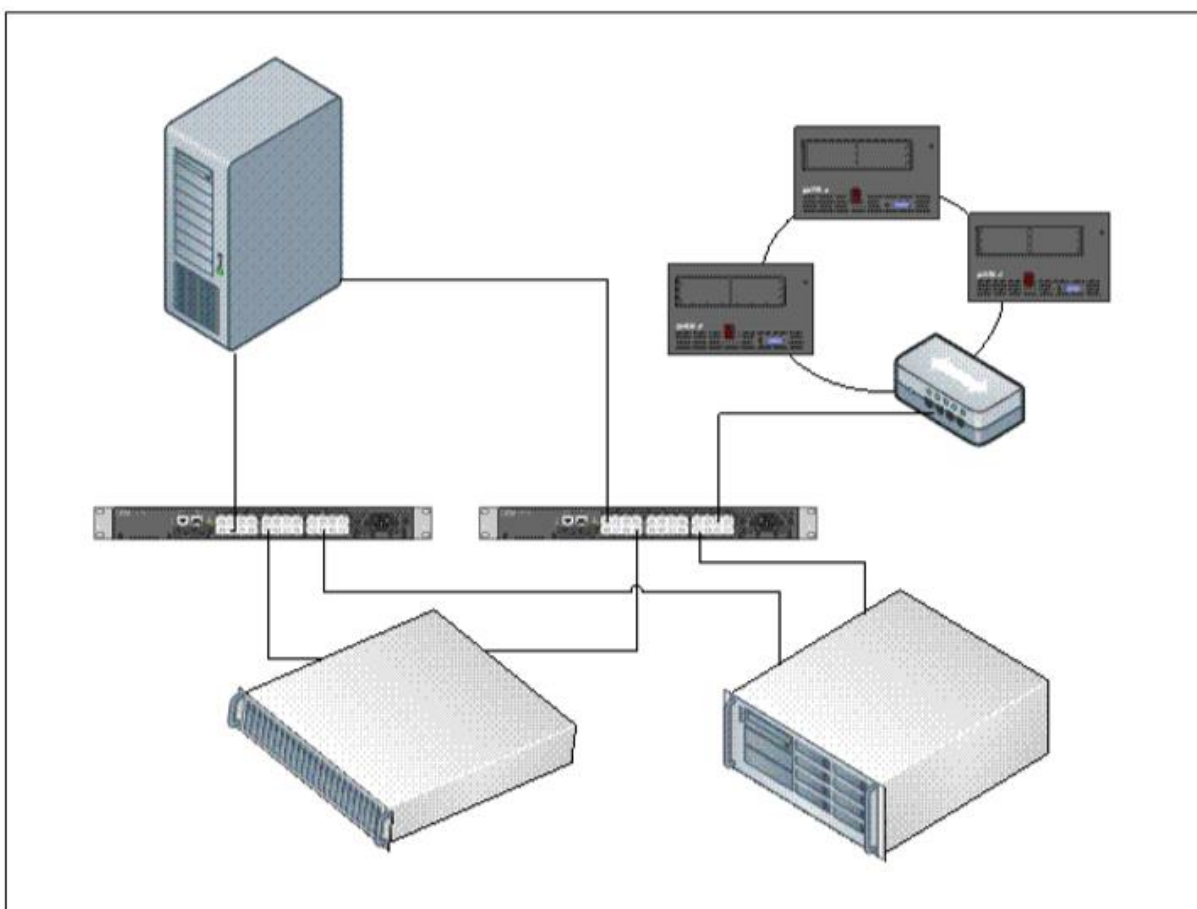
Zabezpečení dat a jejich vysoká dostupnost bývá nejsilnější motivací pro implementaci SAN do informačního systému společnosti. SAN v tomto ohledu nabízí několik bonusů:

- Fyzické oddělení dat a serverů
- Vyšší propustnost
- Definici redundantních cest ke zdrojům
- Podporu pro clusterová řešení a architekturu "no single point of failure" [9]

Pojmem Single point of failure (SPOF) bývá označován prvek nebo část systému, jehož vyřazení způsobí nefunkčnost celého systému. Architektura "no single point of failure" zcela obecně znamená, že nějaké konkrétní zařízení s tímto označením (např. síťové přepínače IP) by mělo být alespoň teoreticky zabezpečené vůči výpadkům, útočnickům (pak překládáme vysloveně jako „žádná skulina v systému“) atd. [10]

2 SWITCHED FABRIC STORAGE AREA NETWORK

Topologie Switched Fabric patří v současnosti k nejvíce používaným typům uspořádání oddělené datové sítě postavené na technologii Fibre Channel. Další dvě základní topologie – Point To Point a Arbitrated Loop – jsou používány zřídka, zpravidla interně v zařízeních typu datové úložiště nebo jako dílčí část rozlehlejších sítí typu Switched Fabric. Obrázek 2 ukazuje příklad topologie sítě typu Switched Fabric. Lze vidět dvě fabric (dva nezávislé SAN switche), vícecestně připojená koncová zařízení i zařízení zapojená do smyčky.

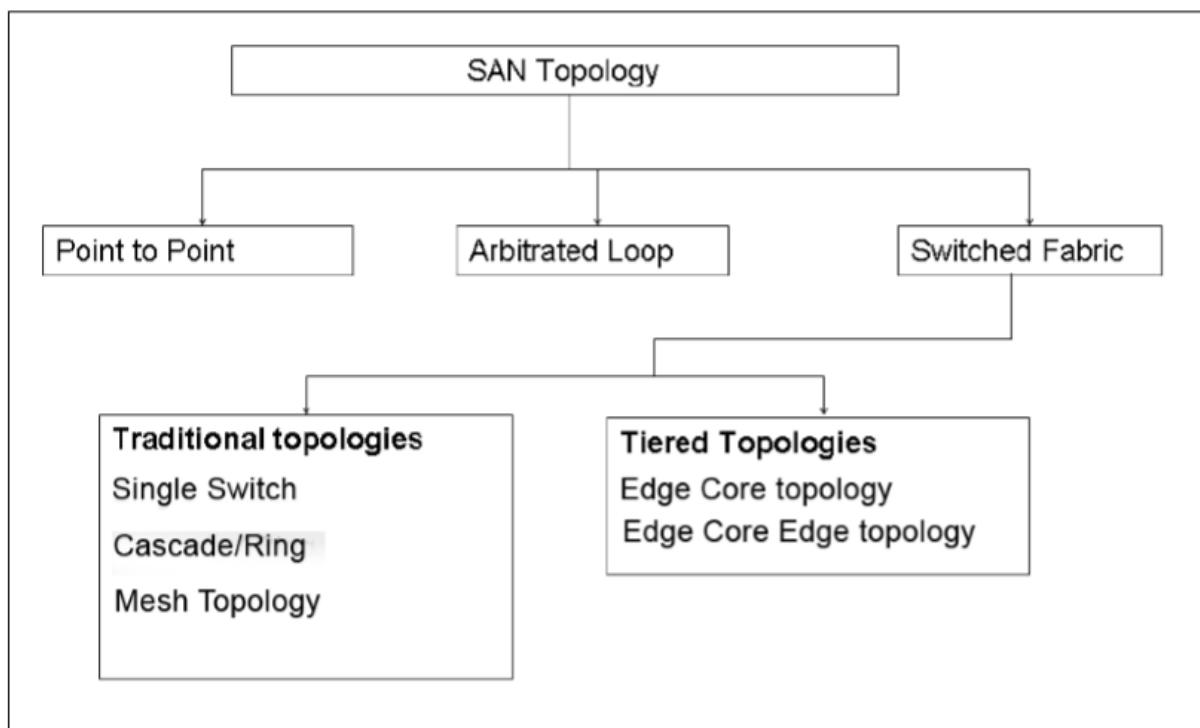


Obrázek 2: Příklad topologie Fibre Channel SAN

Zdroj: [5]

Technologie Fibre Channel (FC) využívá pro komunikaci optická vlákna s rychlostí přenosu až 20 Gbps a protokol FCP. Proto jsou pro provoz takové sítě nutné speciální FC switche (přepínače) a každé koncové zařízení musí být připojené do SAN pomocí tzv. HBA (Fibre Channel Host Bus Adapter). Každý HBA má svoji unikátní 64-bitovou adresu, pomocí které je zařízení (port zařízení) v SAN jednoznačně identifikováno. Jedná se o WorldWide Name (WWN), obdobu MAC adresy u síťové karty.

Druhy topologie sítě SAN založené na technologii Fibre Channel ukazuje přehledně Obrázek 3.



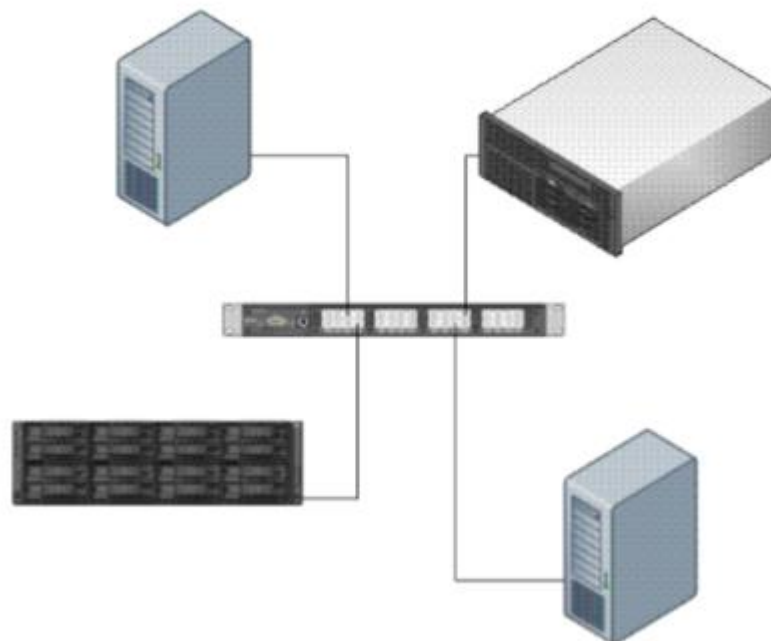
Obrázek 3: Topologie Fibre Channel SAN

Zdroj: [5]

Jak ukazuje obrázek, podtypy sítě Switched Fabric SAN se dělí na topologie tradiční (Single Switch, Cascade/Ring, Mesh) a topologie vrstvené. V dalším textu budou tyto topologie představeny podrobněji.

2.1 Single Switch topologie

Pro topologii Single Switch je charakteristická přítomnost pouze jednoho přepínače (switche) a tedy žádná propojení typu přepínač – přepínač. Jedná se o nejjednodušší topologii Switched Fabric SAN, která neobsahuje žádnou redundanci propojení. Protože tato topologie obsahuje SPOF (Single Point Of Failure - prvek nebo část systému, jehož vyřazení způsobí nefunkčnost celého systému), bývá zřídka používána. Obrázek 4 ukazuje topologii Single Switch, kde všechna zařízení jsou připojena do jednoho přepínače.



Obrázek 4: Topologie Single Switch

Zdroj: [5]

2.2 Cascade/Ring topologie

Kaskádová a kruhová topologie je zobrazena na Obrázku 5. Přepínače jsou propojeny ve frontě za sebou, v případě kruhové topologie se tato fronta vrací k prvnímu přepínači a propojení se uzavírá v kruh.

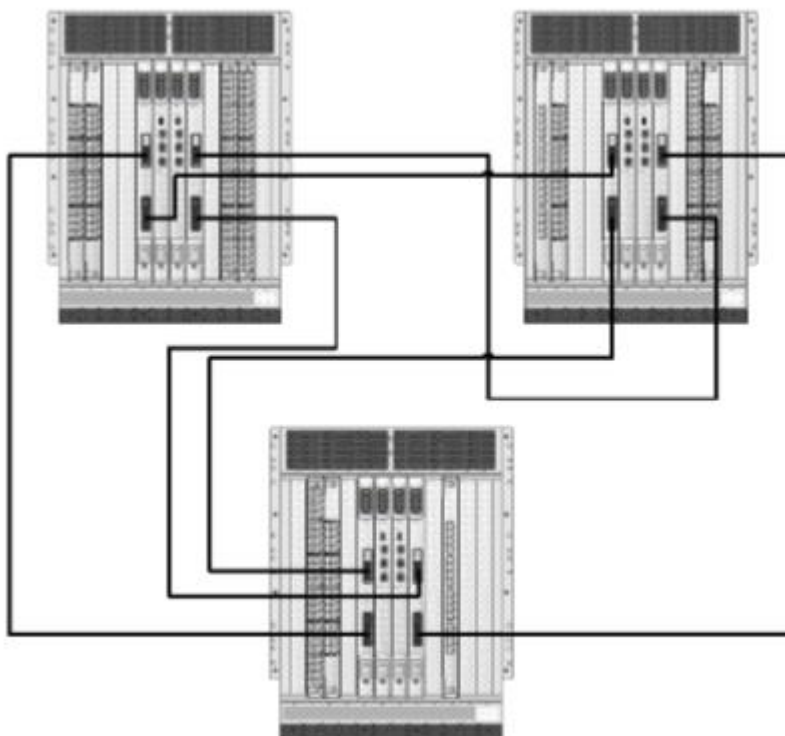


Obrázek 5: Topologie Cascade a Ring

Zdroj: [5]

2.3 Mesh topologie

Mesh topologie se vyznačuje propojením každého přepínače se všemi ostatními přepínači v dané fabric. Díky redundanci každého spojení se tato topologie vyznačuje vysokou spolehlivostí. Mesh topologii ukazuje Obrázek 6.



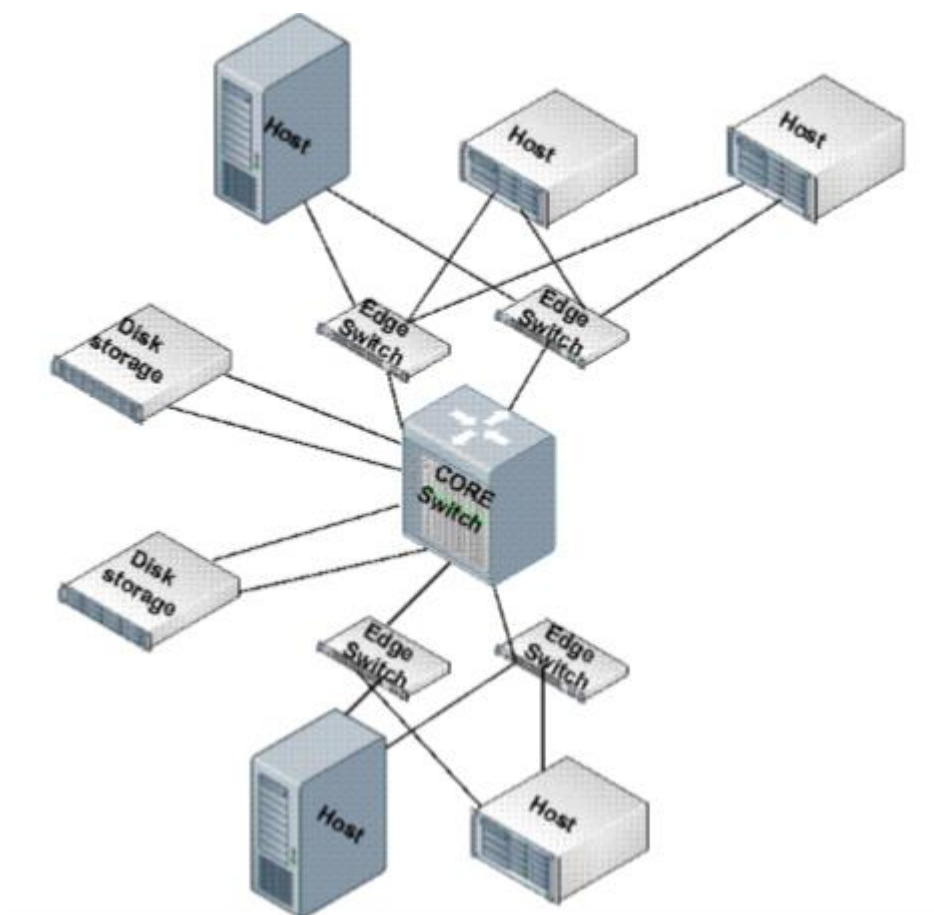
Obrázek 6: Topologie Mesh

Zdroj: [5]

2.4 Vrstvené topologie

Vrstvené topologie se vyznačují použitím tzv. Core switchu jako hlavního prvku sítě. Aby Core switch nepředstavoval Single point of failure (SPOF), jsou použity různé technologie, např. stohování switchů.

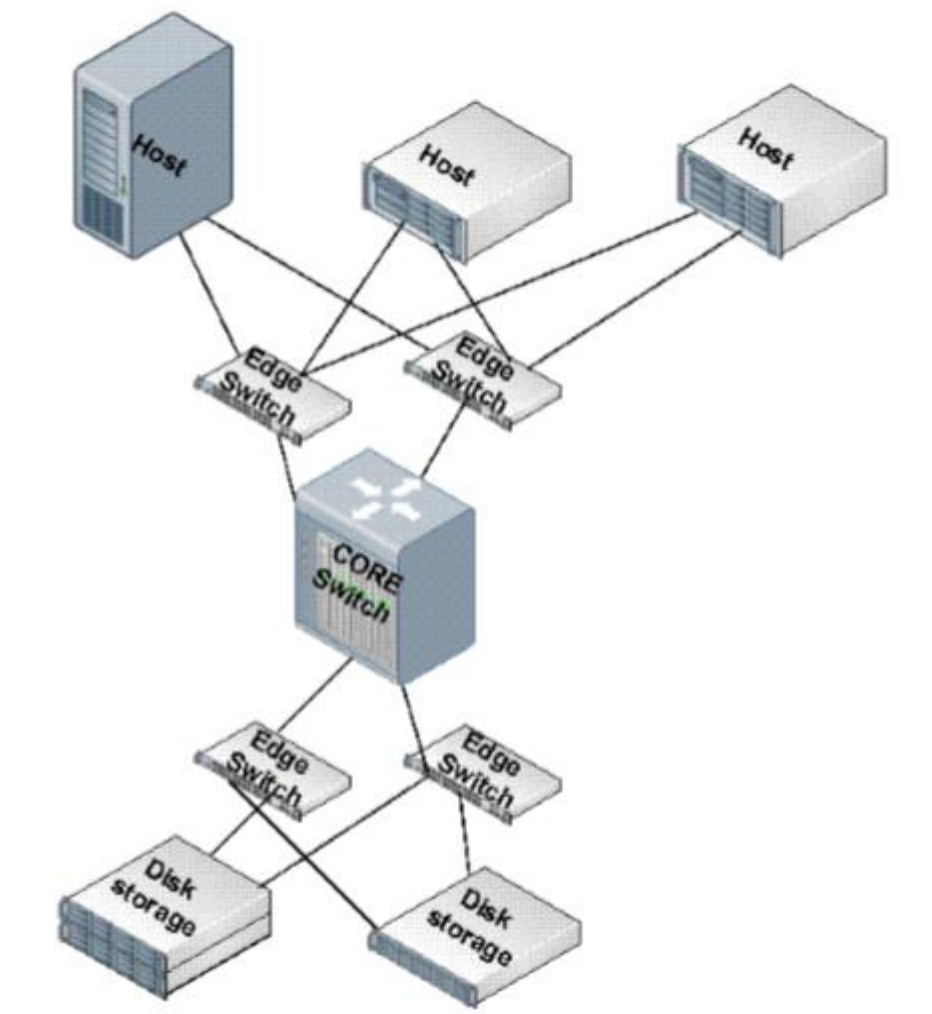
Topologie Edge Core představuje síť, kde koncová zařízení typu server jsou do sítě připojena přes další vrstvu switchů. Datová úložiště jsou naopak připojena přímo ke Core switchi (viz Obrázek 7).



Obrázek 7: Topologie Edge Core

Zdroj: [5]

Naproti tomu topologie Edge Core Edge umožňuje do Core switchu připojovat pouze další SAN switchy, přes které se do SAN připojují veškerá koncová zařízení včetně datových úložišť. Zpravidla existuje skupina SAN switchů připojujících do SAN koncová zařízení typu server a skupina SAN switchů připojujících datová úložiště (viz Obrázek 8).



Obrázek 8: Topologie Edge Core Edge

Zdroj: [5]

Tabulka 1: Switched Fabric SAN a její varianty topologie

Switched Fabric SAN	
Tradiční topologie	Single Switch Topology
	Cascade/Ring Topology
	Mesh Topology
Vrstvené topologie	Edge Core Topology
	Edge Core Edge Topology

Zdroj: vlastní zpracování

2.5 Konfigurace Switched Fabric

SAN se skládá z mnoha typů prvků (optické kabely, přepínače, servery, disková pole, převodníky, optické vany a další). Z pohledu možnosti jejich konfigurace lze ovšem určit dva hlavní typy. V terminologii SAN se jedná o devices (zpravidla přepínače) a nody (koncová zařízení - servery, disková pole, zálohovací knihovny a jiné). [5]

Příkladem device je SAN switch Cisco MDS 9148 (viz Obrázek 9).

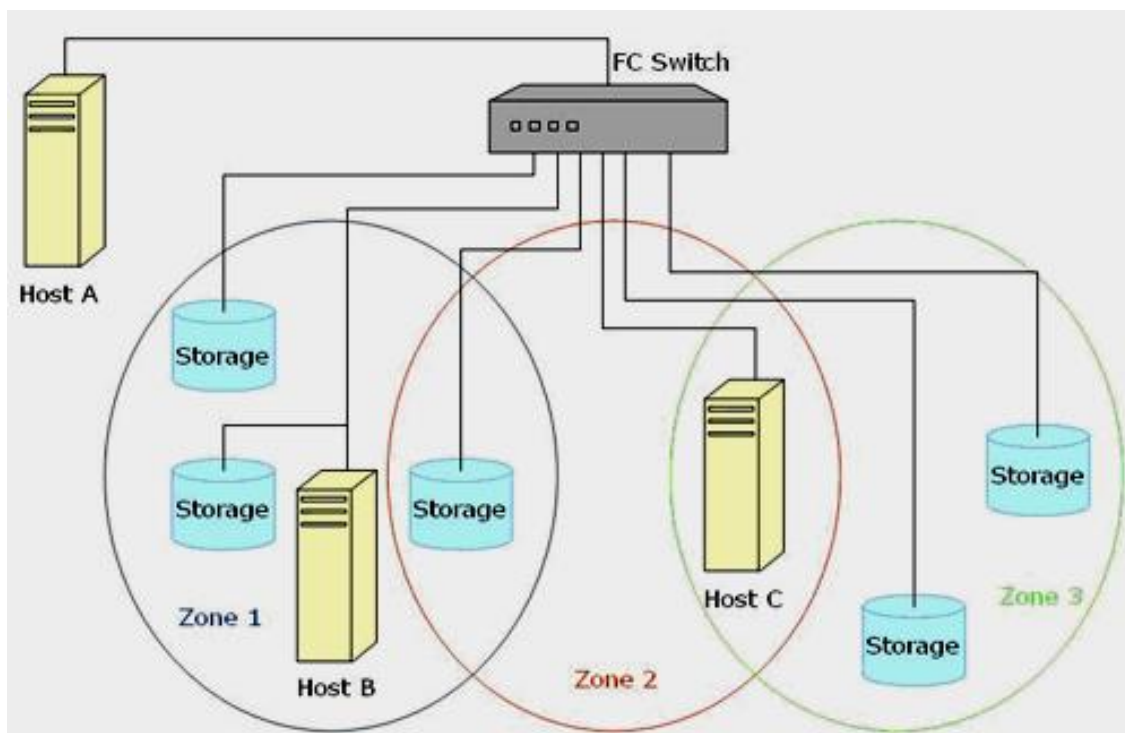


Obrázek 9: Cisco MDS 9148 Fabric Switch

Zdroj: [5]

Konfigurace SAN je vlastně souhrn konfigurací jednotlivých devices. Ty obsahují nastavení parametrů konkrétního přepínače (switche), např. určení IP adres pro management (adresa switche, maska sítě, adresa brány), IP adresy poskytovatele času, nastavení a popis jednotlivých Fibre Channel portů atd. Především ale obsahují konfiguraci Fabric, obsahující výčet identifikovaných koncových zařízení podle WWN a jejich pojmenování (příkaz *fcalias*), výčet definovaných zón a zonesetů.

Zóny zajišťují vzájemnou viditelnost vybraných zařízení. V rámci zóny jsou vyjmenovány aliasy (nebo přímo WWN) těch zařízení, která mají spolu komunikovat. WWN identifikují připojená zařízení do SAN. 64-bitový hexadecimální řetězec ovšem není příliš vhodný pro opakované zápisy do konfigurace, proto se zpravidla na začátku konfigurace definuje pro každou WWN tzv. alias – slovní pojmenování definující připojené zařízení. Tyto aliasy se pak dále používají při vytváření zón. Například pro komunikaci zálohovací knihovny se zálohovacím serverem není potřeba, aby zálohovací knihovna viděla jiné zařízení než pouze daný server. Nadefinujeme tedy zónu pouze pro tato dvě zařízení. Naproti tomu zálohovací server bude muset přistupovat na většinu datových úložišť, což zařídíme nadefinováním další zóny. Příklad použití zón znázorňuje Obrázek 10. [11]



Obrázek 10: Příklad zónování

Zdroj: [2]

Zonesety sdružují vybrané nadefinované zóny. Aktivní v dané konfiguraci může být pouze jeden zoneset. Zóna, která není v aktivním zonesetu, se neuplatní. Je tedy možné mít nadefinováno více zonesetů s různými zónami a v případě potřeby aktivní zoneset měnit. [11]

V praxi může mít definice více zonesetů význam například při stěhování daného aktivního prvku mezi různými SAN nebo při dočasném rozšiřování SAN.

2.6 Dílčí souhrn

V této kapitole byly představeny možné topologie sítě SAN Switched Fabric. Charakteristiky a zvláštnosti těchto topologií budou zohledněny při tvorbě aplikace tak, aby aplikace byla použitelná pro všechny uvedené topologie. Principy konfigurace SAN byly uvedeny pro představu čtenáře této práce, jaké prvky SAN bude aplikace dokumentovat (fabric, device, nod, zone, zoneset).

3 NÁVRH POSTUPU PRÁCE

Tato kapitola představuje jednotlivé kroky předpokládaného postupu práce.

3.1 Rozhodnutí o použití aplikace a jejím formátu

Aplikace pro dokumentaci sítě SAN bude poskytovat informace o dokumentované síti určitému okruhu uživatelů. Tento okruh uživatelů by měl být popsán. Stejně tak je nutné se rozhodnout pro určitý formát aplikace a zvolit technologie a technické prostředky, pomocí nichž bude vývoj aplikace probíhat. Toto je zpracováno v kapitole 4.

3.2 Rozhodnutí o způsobu vývoje aplikace a vlastní vývoj

Metod pro vývoj aplikací existuje více. V této fázi bude třeba vybrat jakým způsobem bude aplikace vyvíjena a následně tuto aplikaci zvoleným způsobem vytvořit.

Návrhu postupu práce, nezávisle na zvoleném formátu aplikace, předcházela myšlenka začít řešit dané zadání vytvořením požadované aplikace pro jednu konkrétní síť SAN, jejíž jsem správcem. Praktická znalost a možnost správy jedné konkrétní sítě SAN umožňuje vytvoření a odladění aplikace „ušité na míru“ pro tuto síť. Od této myšlenky se odvíjí způsob vývoje aplikace představený v kapitole 5.

3.3 Vyhodnocení použitého postupu vývoje aplikace

Zvolený způsob vývoje aplikace a jeho případné použití při řešení podobných problémů je zhodnoceno v kapitole 6.

4 ÚČEL APLIKACE A TECHNICKÉ PROSTŘEDKY VÝVOJE APLIKACE

4.1 Účel aplikace

Aplikace pro dokumentaci sítí SAN bude poskytovat informace o dokumentované síti uživatelům aplikace. Těmi budou především správci těchto sítí. Předpokládá se, že budou zároveň administrátory aplikace a budou vytvářet data aplikace. Aplikace jim pak umožní rychle se orientovat v konfiguraci dané SAN a přehledně tuto konfiguraci zobrazit bez nutnosti přístupu na jednotlivé devices a vyčítání informací o SAN z konfigurace každého SAN switchu.

Okruh uživatelů aplikace nemusí tvořit pouze správci dokumentovaných sítí. Informace o konfiguraci SAN poskytnuté aplikací mohou být užitečné např. i pro systémové inženýry, projektanty nových systémů či aplikací a správce ostatních částí informačního systému organizace.

4.2 Formát aplikace a technické prostředky vývoje aplikace

Klasický formát desktopové aplikace pozvolna přepouští své místo aplikacím webovým. Desktopový formát se stále používá tam, kde má své opodstatnění, např. při práci s velkými objemy dat jako zpracování obrazu či videa, vytváření složitých grafických výstupů apod. Aplikace pro dokumentaci sítě SAN bude pracovat s malým množstvím dat. Taktéž na grafickou prezentaci aplikace nevyplývají z jejího účelu vysoké nároky, proto bude vhodné vytvořit ji jako aplikaci webovou.

Webové aplikace jsou v dnešní době velice populární. Na rozdíl od klasických desktopových aplikací přinášejí webové aplikace řadu výhod především z pohledu provozu, distribuce a samotné správy aplikace. U webových aplikací odpadá nutnost instalace aplikace na každý počítač zvlášť, hlídat a provádět samostatně aktualizace či do detailu studovat kompatibilitu prostředí, ve kterém chceme aplikaci provozovat. Lze si vystačit pouze s webovým prohlížečem, který funguje jako tenký klient a který zprostředkovává kontakt mezi uživatelem a webovým serverem poskytujícím webovou aplikaci. Přístup k aplikaci je možný prakticky odkudkoliv. Z těchto důvodů se řada tradičních desktopových systémů nově implementuje jako webové aplikace. [3]

Pro vývoj webových aplikací existuje velké množství programovacích prostředků. Tyto mohou být součástí větších systémů nebo mohou existovat jako samostatný produkt. Profesionální nástroje práci usnadní, ale jsou zpravidla placené. Open source prostředky jsou volně dostupné, vyžadují však poněkud širší znalosti programování. Pro vývoj aplikace zmiňované v této práci byly využity dále popsané prostředky.

4.2.1 JavaScript

JavaScript je událostmi řízený programovací jazyk, kterým lze oživit webové stránky či webové aplikace. Umožňuje vytvořit aktivní prostředí reagující na uživatelskou činnost (změna tlačítek odkazů po přejetí kurzorem myši apod.).

JavaScript je objektově orientovaný jazyk. Pracuje s objekty internetového prohlížeče, kterými mohou být okna, formuláře nebo prvky formuláře, třeba tlačítka a zaškrťávací pole. Tyto objekty pak mají své vlastnosti a metody. [6]

Ve vyvíjené webové aplikaci bylo pomocí tohoto prostředku vytvořeno menu aplikace sestávající z grafických tlačítek reagujících na pohyb kurzoru myši.

4.2.2 PHP

Jazyk PHP je jedním z nejrozšířenějších jazyků pro tvorbu webových aplikací. Jeho obliba je doložena nejen širokou vývojářskou komunitou, ale i faktem, že více jak osmdesát procent všech webových stránek na internetu je vytvořeno v tomto jazyce. Neustálý vývoj jazyka a jeho zvyšující se kvalita je jedním z dalších faktorů, který významným způsobem ovlivňuje jeho celosvětovou popularitu. [3]

PHP je skriptovací jazyk na straně serveru, který byl navržen speciálně pro web. Do stránky HTML je možno vkládat kód jazyka PHP, jenž se provede při každé návštěvě této stránky. Kód jazyka PHP je interpretován webovým serverem, přičemž generuje HTML nebo jiný výstup, který uvidí návštěvník.

PHP je projekt s otevřeným kódem, což znamená otevřený přístup k jeho zdrojovému kódu s možností jeho používání, změny a distribuce. [1]

PHP tvoří podstatnou část zdrojového kódu vyvíjené aplikace. Je použitý také z důvodu jeho podpory databázových systémů. PHP nativně podporuje více databázových systémů, mezi jinými i MySQL.

4.2.3 MySQL

MySQL je rychlý a robustní systém pro správu relačních databází. Databáze umožňuje aplikaci efektivně ukládat, vyhledávat, řadit a načítat data. Zajišťuje, aby k datům mohlo

přístupovat více uživatelů současně, nabízí rychlý přístup a dohlíží na to, aby přístup získali pouze oprávnění uživatelé (aplikace). MySQL je veřejně dostupný. Používá SQL (Structured Query Language) – standardní databázový dotazovací jazyk. [1]

4.3 Dílčí souhrn

Aplikace pro dokumentaci sítě SAN bude koncipována jako webová aplikace s provozováním v lokální síti. Podstatná část aplikace bude napsána pomocí skriptovacího jazyka PHP. Na vytvoření menu aplikace bude použit událostmi řízený skriptovací jazyk JavaScript. Jako databázový systém bude použit MySQL nebo jeho varianta (např. MariaDB). Tabulka 2 shrnuje použité technické prostředky použité pro vývoj aplikace.

Tabulka 2: Technické prostředky použité pro vývoj aplikace

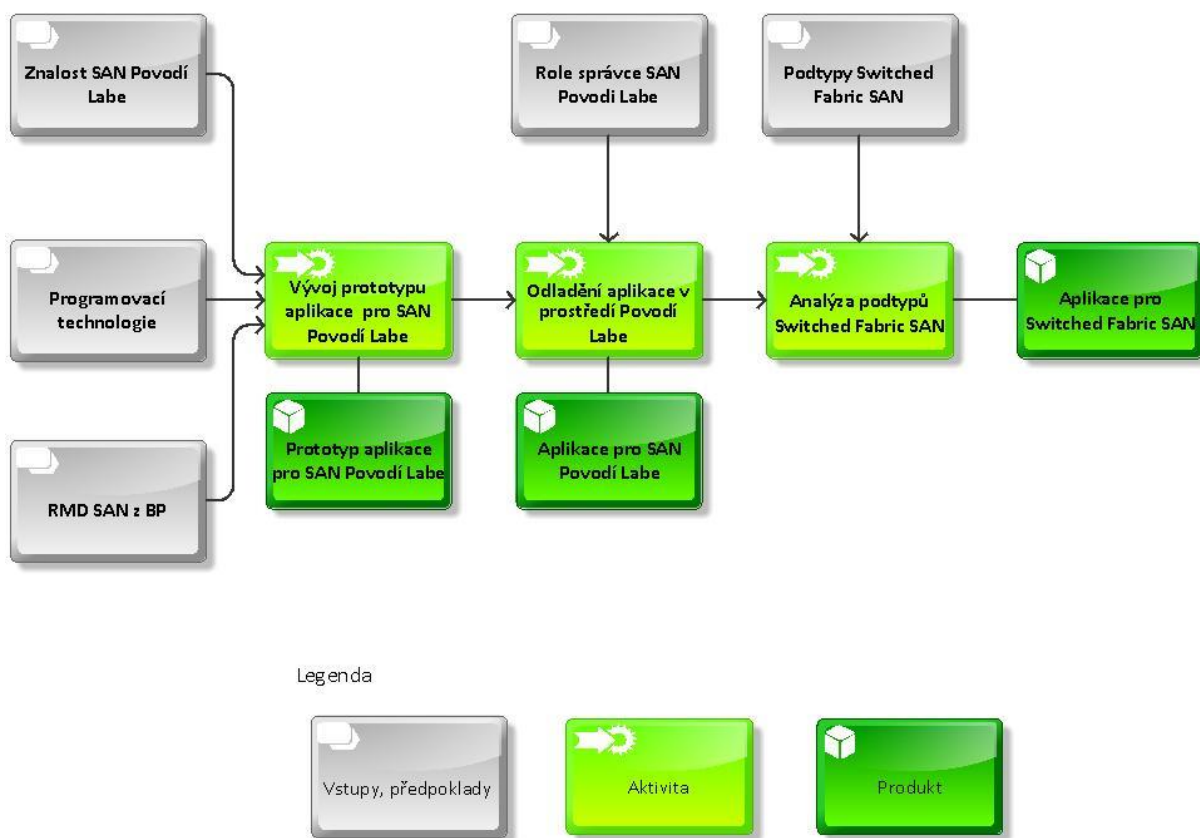
PHP	Tělo aplikace
JavaScript	Menu aplikace
MySQL	Databázový systém

Zdroj: vlastní zpracování

5 NÁVRH A VÝVOJ APLIKACE

Provozovaných SAN sítí je o několik řádů méně než např. lokálních počítačových sítí. Přístup k nim je zpravidla velmi omezený, neboť se jedná o kritickou infrastrukturu IS. Vyvíjet aplikaci pro celou skupinu sítí SAN bez možnosti testování aplikace se jeví jako velmi teoretická činnost. Proto rozhodnutí o způsobu vývoje aplikace předcházela myšlenka začít řešit dané zadání vytvořením požadované aplikace pro jednu konkrétní síť SAN, jejíž jsem správcem. Praktická znalost a možnost správy jedné konkrétní sítě SAN umožňuje vytvoření a odladění aplikace „ušité na míru“ pro tuto síť. Na základě této myšlenky jsem se rozhodl vyvíjet aplikaci prostřednictvím řady prototypů vedoucí k finální verzi požadované aplikace. První prototypy budou vytvořeny a odladěny pro konkrétní topologii sítě SAN, pro obecnější použití aplikace pak bude aplikace rozšířena o prvky a funkcionality vyskytující se v ostatních topologiích Switched Fabric SAN. Navržený postup vývoje aplikace přehledně ukazuje Obrázek 11.

Vývoj aplikace pomocí prototypů



Obrázek 11: Model postupu při vývoji aplikace

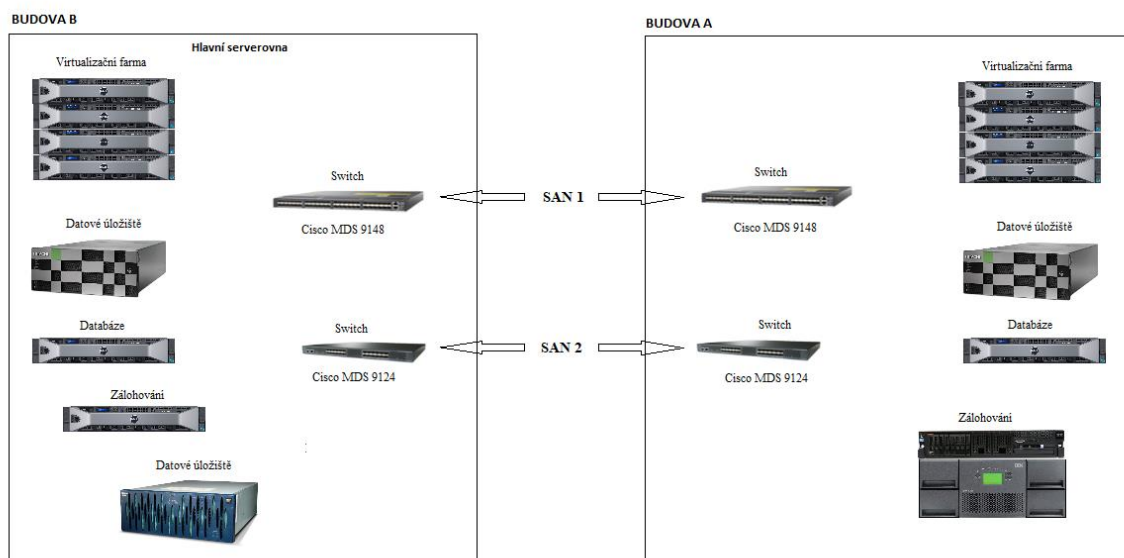
Zdroj: vlastní zpracování

Model postupu při vývoji aplikace byl vytvořen pomocí SW ARIS Express – nástroj Business proces.

5.1 Prototyp 1

Návrh prvního prototypu aplikace (Prototyp 1) byl vytvořen pro konkrétní síť SAN, provozovanou ve společnosti Povodí Labe, státní podnik.

Datová síť Povodí Labe je umístěna na generálním ředitelství v Hradci Králové. Z bezpečnostních důvodů je fyzicky umístěna ve dvou oddělených lokalitách, ve dvou samostatných budovách. Obě lokality jsou propojeny optickými a ethernetovými kabely. Datovou síť tvoří z důvodu zajištění redundance propojení a vysoké dostupnosti dat dvě dvojice propojených SAN switchů představující dvě oddělené sítě (fabric). Z pohledu typů topologií Switched Fabric SAN představených v kapitole 2 se jedná o topologii kaskádovou. V každé lokalitě jsou zrcadlově umístěny farmy serverů používané virtualizační platformy, databázové servery a centrální datová úložiště. Tato koncová zařízení jsou připojena do každé z fabric a jsou tedy dostupná po více než jedné cestě. Tím je zajištěna jejich dostupnost při výpadku jednotlivého zařízení i při výpadku celé fabric. V lokalitě B je navíc umístěn ještě zálohovací server a datové úložiště sloužící pro zálohování. V lokalitě A je navíc umístěn další zálohovací server spolu s páskovou zálohovací knihovnou. Schematicky znázorňuje síť SAN firmy Obrázek 12.

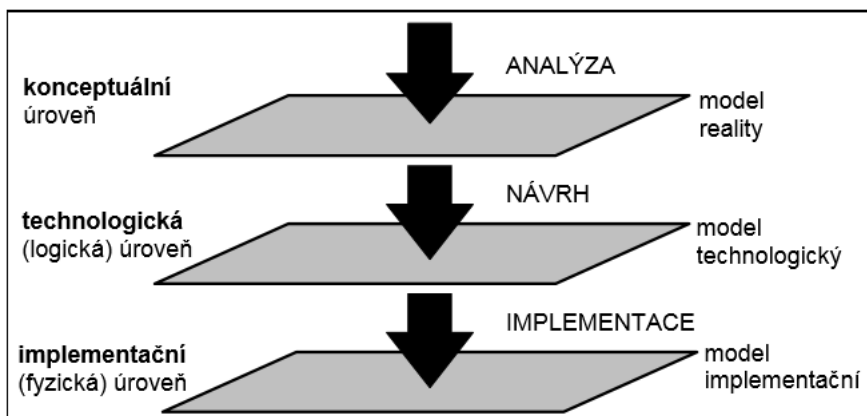


Obrázek 12: Síť SAN společnosti Povodí Labe, státní podnik

Zdroj: vlastní zpracování

5.1.1 Datový model

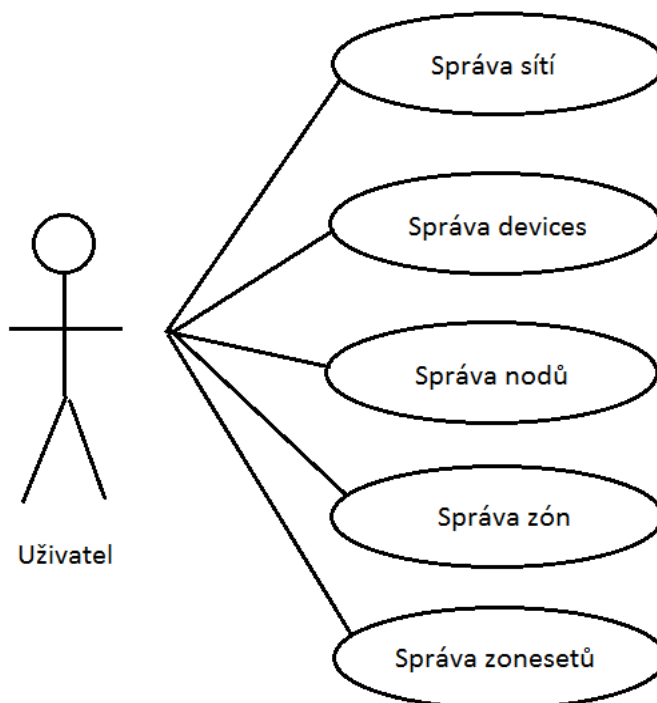
Datový model prototypu 1 odpovídá relačnímu modelu z mé bakalářské práce z roku 2017 Dokumentace sítí SAN. Návrh datových struktur vycházel z konceptu tří úrovní a strukturovaného přístupu (viz Obrázek 13).



Obrázek 13: Koncept tří úrovní

Zdroj: [2]

V rámci konceptuální úrovně byl určen aktér a jeho případy užití (viz Obrázek 14).

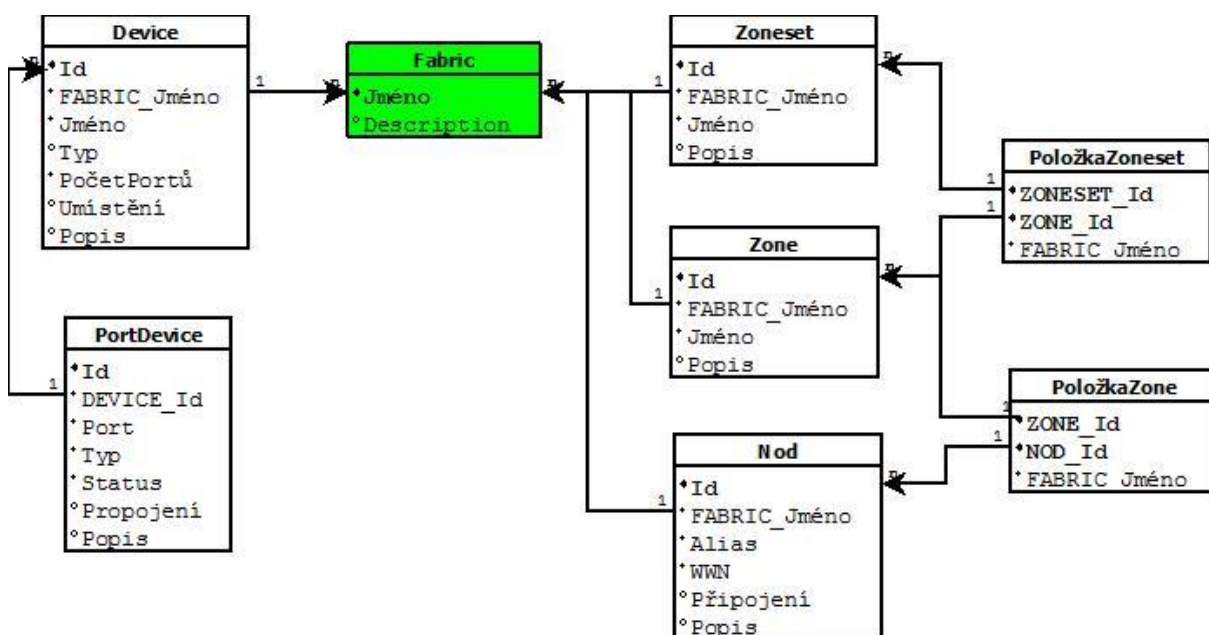


Obrázek 14: Use Case diagram

Zdroj: vlastní zpracování

Dále v rámci této úrovně modelování byly navrženy entity vytvářeného datového modelu aplikace a jejich atributy. Vztahy mezi jednotlivými entitami včetně kardinality a parcuality byly popsány pomocí ER diagramu.

V rámci druhé (technologické) úrovně modelování byla provedena transformace entit a vztahů mezi nimi na návrhy relací. Navržené relace prošly procesem normalizace, a to prostřednictvím pravidel používaných pro odstraňování anomálií v datovém modelu, tzv. normálních forem (1NF, 2NF a 3NF). Výsledkem byl relační model, který jsem nyní použil při vývoji prototypu 1 aplikace (viz Obrázek 15).



Obrázek 15: Relační model

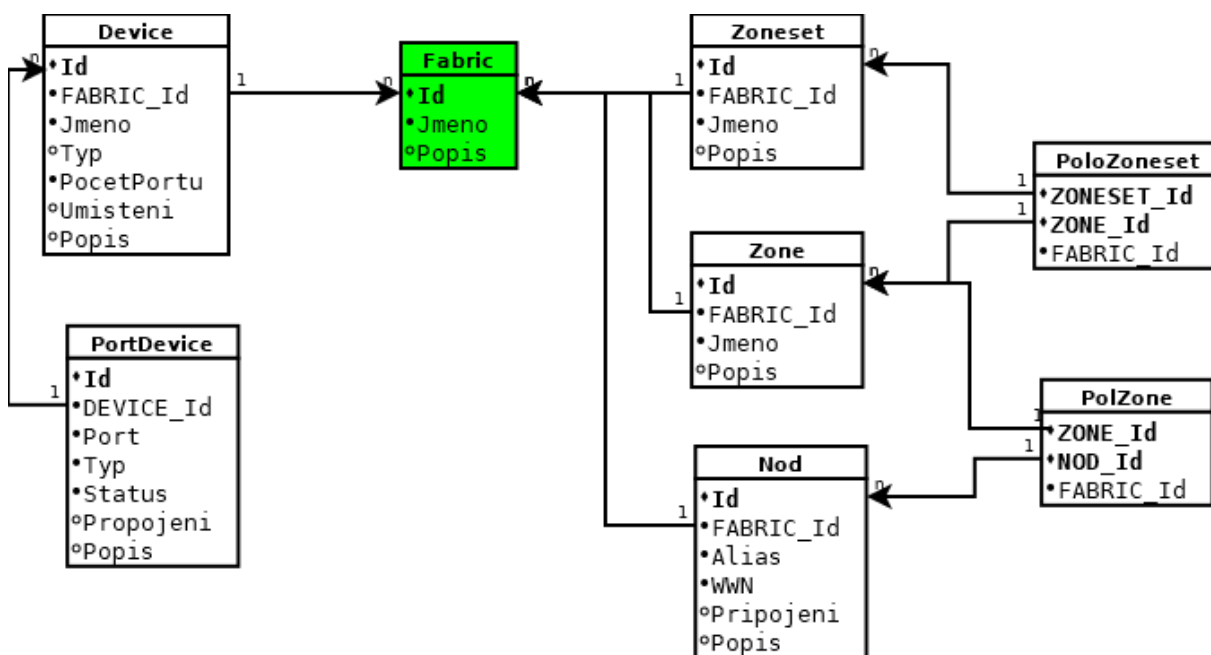
Zdroj: vlastní zpracování

V rámci třetí úrovně (implementační) byl zmíněný relační model implementován do konkrétního prostředí. Pro implementaci datového modelu byl zvolen programový prostředek MariaDB. Jedná se o fork MySQL (varianta MySQL), který vytvořil sám původní autor systému MySQL Monty Widenius. [4]

Vlastní implementace probíhala pomocí volně dostupného nástroje phpMyAdmin určeného k administraci relačních databází typu MySQL nebo MariaDB.

Definice datového modelu v relační databázi MariaDB (konkrétně při využití databázového serveru MariaDB 5) začíná vytvořením nové databáze, jejím pojmenováním a určením defaultní znakové sady. Databáze představující výše zmíněný datový model byla

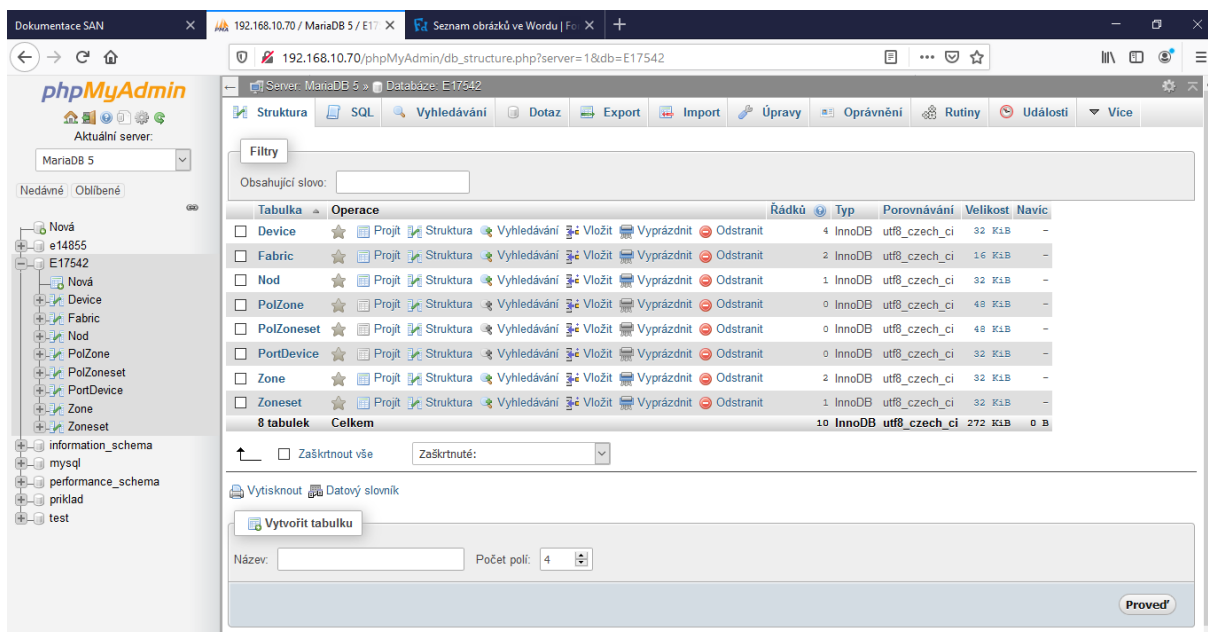
pojmenována E17542, jako defaultní znaková sada byla zvolena sada utf8_czech_ci. Dále bylo potřeba v dané databázi vytvořit databázové tabulky odpovídající jednotlivým relacím datového modelu: entitním relacím Fabric, Device, PortDevice, Zoneset, Zone, Nod (stejnomené databázové tabulky) a vztahovým relacím PoložkaZoneset a PoložkaZone. Názvy těchto dvou relací jsem při implementaci změnil na PolZoneset a PolZone, stejně tak názvy některých atributů u ostatních relací, abych se vyhnul používání českých znaků v databázi a tím preventivně vyloučil případné možné problémy při nasazení v různých systémech. Relační model prototypu 1 pro implementaci ukazuje Obrázek 16.



Obrázek 16: Relační model prototypu 1

Zdroj: vlastní zpracování

Implementovanou databázi E17542 v prostředí Maria DB 5 včetně souhrnného přehledu tabulek ukazuje prostřednictvím nástroje phpMyAdmin Obrázek 17.



Obrázek 17: Přehled tabulek datového modelu

Zdroj: vlastní zpracování

U každé tabulky byla vytvořena struktura dle v datovém modelu uvedených atributů a určeny klíče. U každého atributu musel být definován jeho název, datový typ, v případě typu text znaková sada (ponechávána defaultní utf8_czech_ci), zda je daný atribut u vytváření nového záznamu nulový, případně jeho výchozí hodnota.

V následujícím textu jsou představeny atributy jednotlivých databázových tabulek včetně definujících parametrů. Databázová tabulka Fabric obsahuje definované sítě v rámci dokumentované SAN. Atributy databázové tabulky Fabric a jejich vlastnosti shrnuje Tabulka 3.

Tabulka 3: Struktura databázové tabulky Fabric

Databázová tabulka Fabric				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Jmeno	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

Databázová tabulka Device obsahuje seznam aktivních prvků (SAN switchů) vyskytujících se v dokumentovaných sítích. Atributy databázové tabulky Device a jejich vlastnosti shrnuje Tabulka 4.

Tabulka 4: Struktura databázové tabulky Device

Databázová tabulka Device				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Fabric_Id	Int (11)		žádná	cizí
Jmeno	Text	utf8_czech_ci	žádná	
Typ	Text	utf8_czech_ci	žádná	
PocetPortu	Int (11)		žádná	
Umisteni	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

Tabulka PortDevice obsahuje seznam portů k jednotlivým device (switchům), pokud jsou uživatelem aplikace vygenerovány. Atributy databázové tabulky PortDevice a jejich vlastnosti shrnuje Tabulka 5.

Tabulka 5: Struktura databázové tabulky PortDevice

Databázová tabulka PortDevice				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Device_Id	Int (11)		žádná	cizí
Port	Int (11)		žádná	
Typ	Text	utf8_czech_ci	žádná	
Status	Text	utf8_czech_ci	žádná	
Propojeni	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

V tabulce Zoneset je uveden seznam definovaných zonesetů používaných v konfiguracích SAN switchů a tedy v konfiguraci SAN. Atributy databázové tabulky Zoneset a jejich vlastnosti shrnuje Tabulka 6.

Tabulka 6: Struktura databázové tabulky Zoneset

Databázová tabulka Zoneset				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Fabric_Id	Int (11)		žádná	cizí
Jmeno	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

Databázová tabulka Zone sdružuje zóny definované v konfiguracích SAN switchů. Atributy databázové tabulky Zone a jejich vlastnosti shrnuje Tabulka 7.

Tabulka 7: Struktura databázové tabulky Zone

Databázová tabulka Zone				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Fabric_Id	Int (11)		žádná	cizí
Jmeno	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

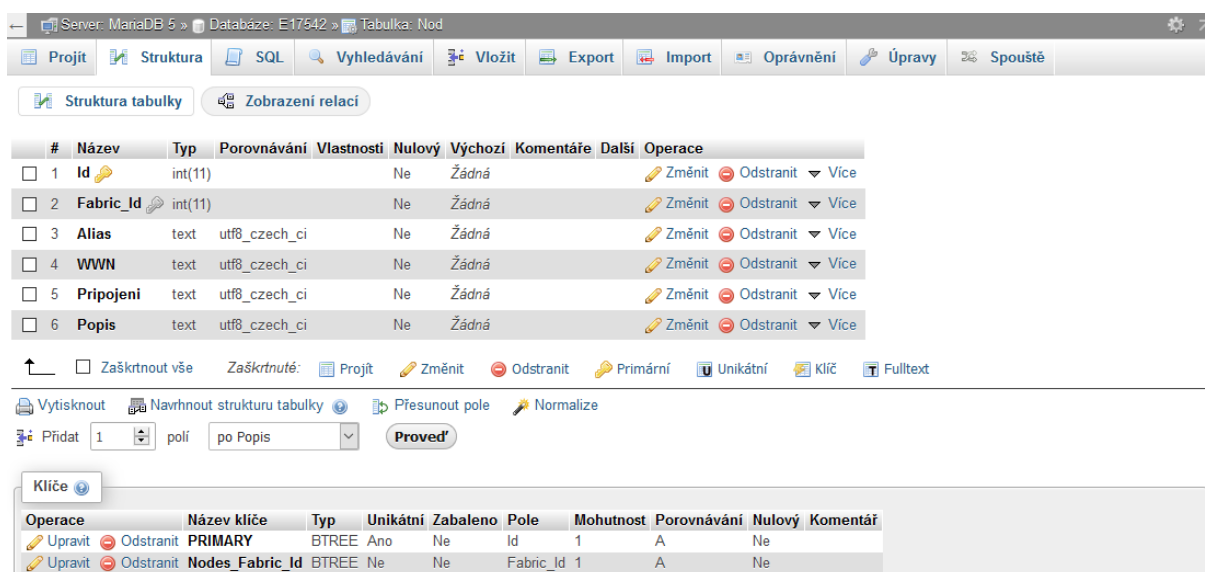
V tabulce Nod jsou zaznamenána všechna připojovaná koncová zařízení do SAN. Každá WWN má v této tabulce svůj záznam. V praxi to znamená, že pokud do SAN připojovaný server má z důvodu zajištění vysoké dostupnosti dva HBA adaptéry, tj. dvě jedinečné WWN, musí mít v této tabulce dva záznamy – např. port 1 a port 2. Datové úložiště mívá zpravidla čtyři nebo osm takovýchto připojení, pak pro toto úložiště bude v tabulce existovat příslušný počet záznamů (čtyři nebo osm). Atributy databázové tabulky Nod a jejich vlastnosti shrnuje Tabulka 8.

Tabulka 8: Struktura databázové tabulky Nod

Databázová tabulka Nod				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Fabric_Id	Int (11)		žádná	cizí
Alias	Text	utf8_czech_ci	žádná	
WWN	Text	utf8_czech_ci	žádná	
Pripojeni	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

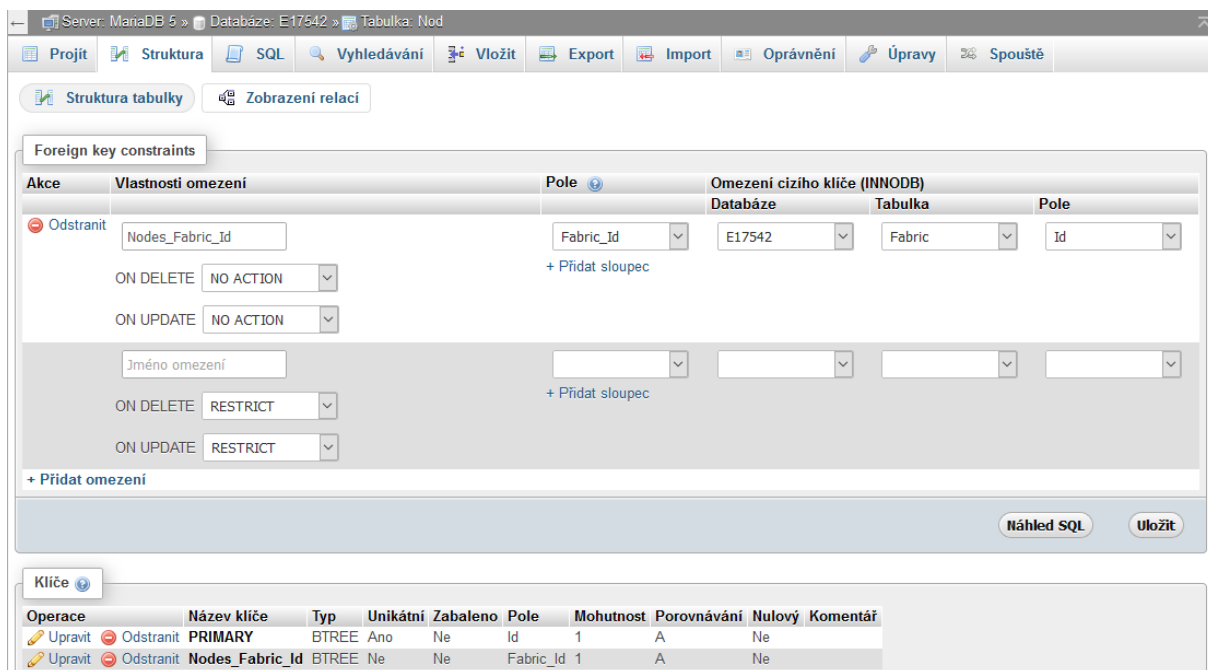
Pro konkrétní představu definování databázové tabulky Nod v databázovém serveru MariaDB 5 jsou dále uvedeny screenshoty administračního nástroje phpMyAdmin. Strukturu databázové tabulky Nod po implementaci v MariaDB ukazuje Obrázek 18.



Obrázek 18: Struktura databázové tabulky Nod

Zdroj: vlastní zpracování

Obrázek 19 ukazuje definované vazby či relace v rámci tabulky Nod.



Obrázek 19: Vazby databázové tabulky Nod

Zdroj: vlastní zpracování

Poslední dvě databázové tabulky odpovídají vztahovým relacím datového modelu. Od předchozích tabulek se liší především svými primárními klíči, které jsou složené ze dvou cizích klíčů.

Tabulka PolZoneset (relace PoložkaZoneset v původním relačním datovém modelu) zaznamenává příslušnost existujících zón k některému z definovaných zonesetů. Atributy databázové tabulky PolZoneset a jejich vlastnosti shrnuje Tabulka 9.

Tabulka 9: Struktura databázové tabulky PolZoneset

Databázová tabulka PolZoneset				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Zoneset_Id	Int (11)		žádná	cizí
Zone_Id	Int (11)		žádná	cizí
Fabric_Id	Int (11)		žádná	cizí

Zdroj: vlastní zpracování

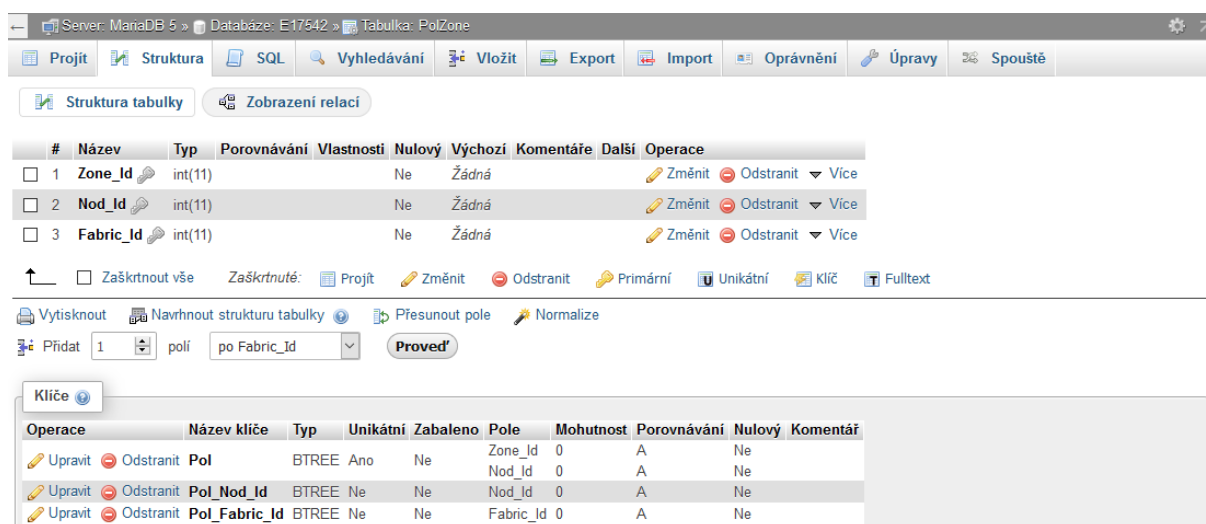
Tabulka PolZone (relace PoložkaZone v původním relačním datovém modelu) zaznamenává příslušnost jednotlivých WWN (nodů) k definovaným zónám. Atributy databázové tabulky PolZone a jejich vlastnosti shrnuje Tabulka 10.

Tabulka 10: Struktura databázové tabulky PolZone

Databázová tabulka PolZone				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Zone_Id	Int (11)		žádná	cizí
Nod_Id	Int (11)		žádná	cizí
Fabric_Id	Int (11)		žádná	cizí

Zdroj: vlastní zpracování

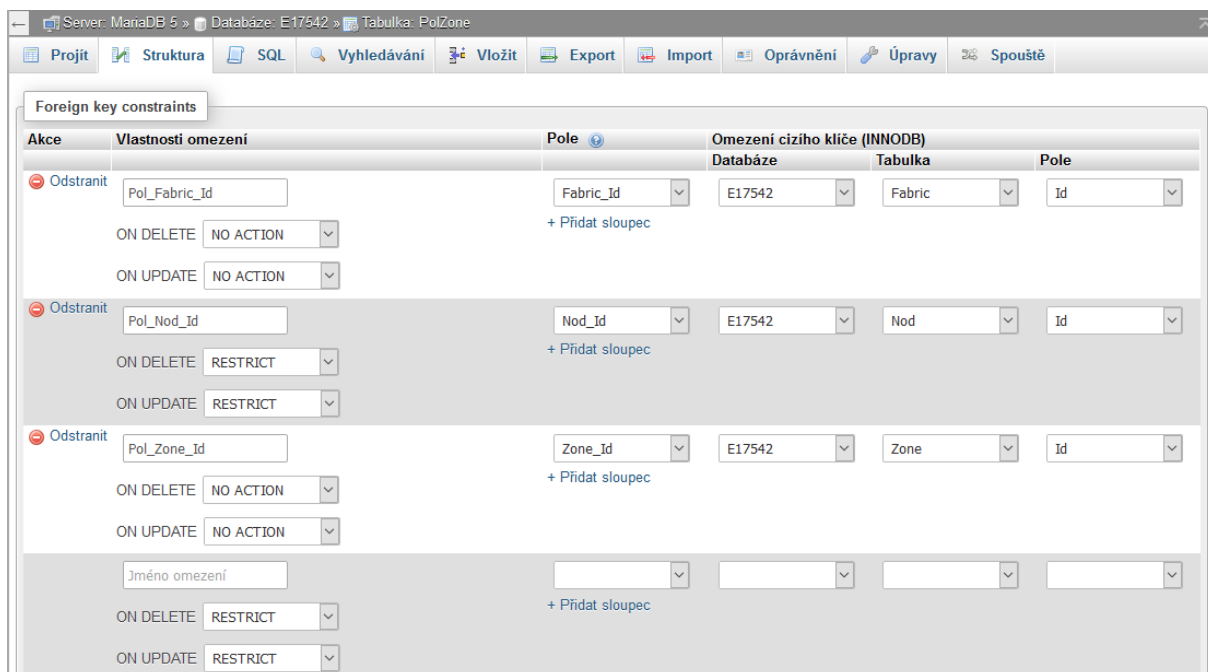
Pro konkrétní představu definování databázové tabulky PolZone v databázovém serveru MariaDB 5 jsou dále uvedeny screenshoty administračního nástroje phpMyAdmin. Strukturu databázové tabulky PolZone po implementaci v MariaDB ukazuje Obrázek 20.



Obrázek 20: Struktura databázové tabulky PolZone

Zdroj: vlastní zpracování

Obrázek 21 ukazuje definované vazby či relace v rámci tabulky PolZone.



Obrázek 21: Vazby databázové tabulky PolZone

Zdroj: vlastní zpracování

Zálohování dat používaných aplikací a uložených v databázi není řešeno v rámci aplikace. Předpokládá se zálohování na úrovni databáze, např. pravidelnými exporty nebo pomocí plánovacích úloh v rámci systému, na kterém bude databázový server provozován.

5.1.2 Aplikace – prototyp 1

Konfigurace sítě SAN je vlastně konfigurací devices. Správce sítě SAN bude tedy zároveň administrátorem aplikace, bude v ní vytvářet data stejnými postupy jako při konfiguraci devices – především pojmenování jednotlivých fabric, evidenci nodů prostřednictvím jejich WWN a aliasů, vytváření zón a jejich sdružování do zonesetů. Aby tato data mohla být k dispozici i ostatním uživatelům a zároveň bylo ošetřeno, že nedojde ze strany uživatelů k zásahu do datových struktur, měla by být aplikace provozována ve dvou modech.

Aplikace v uživatelském módu bude umožňovat:

- Prohlížení seznamu sítí (+ výpis příslušných devices a nodů k dané síti)
- Prohlížení seznamu devices (+ náhled detailu vybraného device)
- Prohlížení seznamu nodů (+ náhled detailu vybraného nodu)
- Prohlížení seznamu zón (+ náhled detailu vybrané zóny)
- Prohlížení seznamu zonesetů (+ náhled detailu vybraného zonesetu)

V aplikaci v administrátorském módu bude možné především:

- Správa sítí (vytvoření, zrušení)
- Správa devices (vytvoření, editace vlastností – příslušnost k fabric, vygenerování portů, detailní náhled, popis,..., zrušení)
- Správa nodů (vytvoření, editace vlastností - WWN, alias, popis, propojení nodu a portu device,..., zrušení)
- Správa zón (vytvoření, editace vlastností - jméno, přiřazení koncových zařízení, popis,..., zrušení)
- Správa zonesetů (vytvoření, editace vlastností - jméno, přiřazení zón, aktivace, popis,..., zrušení)

K provozování aplikace v různých módech na základě uživatelských oprávnění je zapotřebí vyřešit autentifikaci uživatelů. Protože téma autentifikace uživatelů je velmi široké a konkrétní řešení závisí na lokálních podmínkách, vyvíjená aplikace bude provozována s plným přístupem pro každého uživatele. Konkrétní řešení autentifikace a následné omezení některých funkcí může být do aplikace doplněno při nasazení v konkrétním prostředí.

5.1.3 Menu aplikace

Menu aplikace tvoří pět tlačítek umístěných vertikálně v levé části obrazovky – Fabrics, Devices, Nodes, Zones, Zonesets. Pro oživení aplikace je menu napsáno s využitím událostmi řízeného skriptovacího jazyku Javascript. Kombinace funkcí zajišťuje reakci tlačítek menu na pohyb kursoru myši, resp. na jeho umístění nad zobrazené tlačítko. Prakticky se jedná o výměnu obrázku daného tlačítka. Menu v rámci úvodní obrazovky ukazuje Obrázek 22.

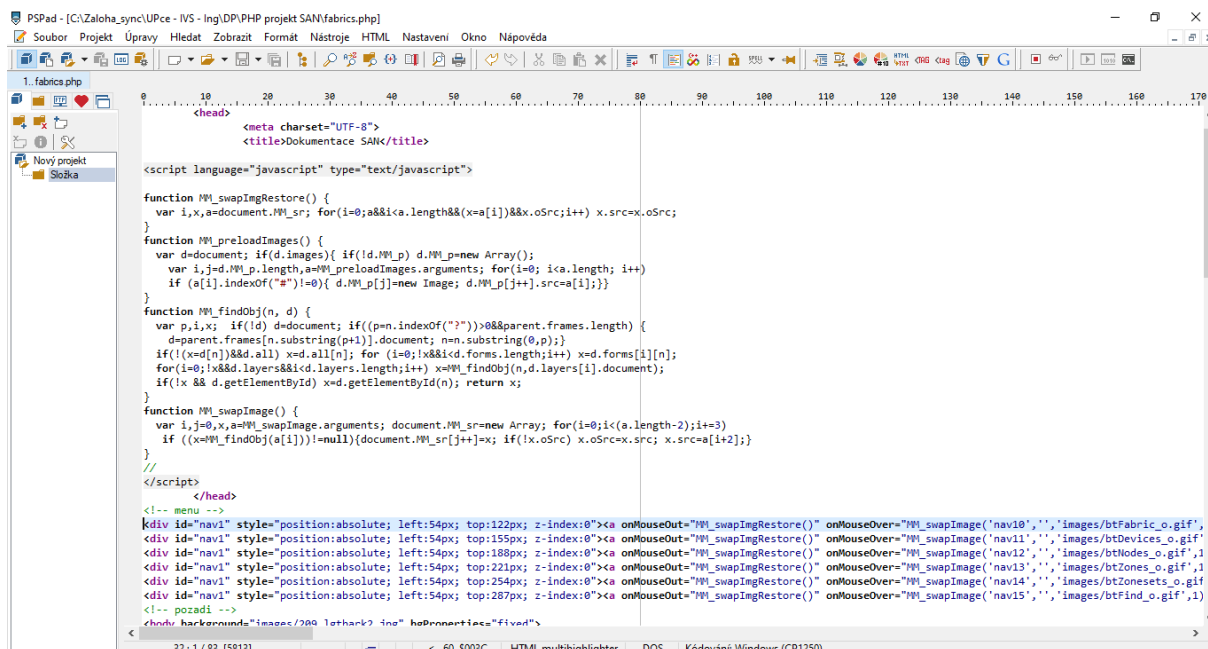


Obrázek 22: Menu aplikace – úvodní okno aplikace

Zdroj: vlastní zpracování

Výměna obrázků tlačítek při pohybu kursoru myši nad tlačítkem budí dojem stlačení tlačítka a interakce aplikace vzhledem k uživateli.

Obrázek 23 ukazuje kód menu aplikace. Tento kód se opakuje v kódu každé stránky aplikace. Editace zdrojových kódů probíhala pomocí volně dostupného textového editoru PSPad. Kompletní zdrojový kód aplikace a databázová struktura jsou přílohou této práce.



```
<head>
<meta charset="UTF-8">
<title>Dokumentace SAN</title>

<script language="javascript" type="text/javascript">

function MM_swapImgRestore() {
  var i,x,a=document.MM_sr; for(i=0;a&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}
function MM_preloadImages() {
  var d=document; if(d.images){ if(!d.MM_p) d.MM_p=new Array();
  var i,j=d.MM_p.length,a=MM_preloadImages.arguments; for(i=0; i<a.length; i++)
  if (a[i].indexOf("#")!=0){ d.MM_p[j]=new Image; d.MM_p[j++].src=a[i];}}
}
function MM_findObj(n, d) {
  var p,i,x; if(!d) d=document; if((p=n.indexOf("?"))>0&&parent.frames.length) {
  d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d[n])&&d.all) x=d.all[n]; for (i=0;i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;i<d.layers.length;i++) x=MM_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}
function MM_swapImage() {
  var i,j=0,x,a=MM_swapImage.arguments; document.MM_sr=new Array; for(i=0;i<a.length-2;i+=3)
  if ((x=MM_findObj(a[i]))!=null){document.MM_sr[j++]=x; if(!x.oSrc) x.oSrc=x.src; x.src=a[i+2];}
}
//
</script>
</head>
<!-- menu -->
<div id="nav1" style="position:absolute; left:54px; top:122px; z-index:0"><a onMouseEvent="MM_swapImgRestore()" onmouseover="MM_swapImage('nav10','','images/btFabric_o.gif',1"
<div id="nav1" style="position:absolute; left:54px; top:155px; z-index:0"><a onMouseEvent="MM_swapImgRestore()" onmouseover="MM_swapImage('nav11','','images/btDevices_o.gif',1"
<div id="nav1" style="position:absolute; left:54px; top:188px; z-index:0"><a onMouseEvent="MM_swapImgRestore()" onmouseover="MM_swapImage('nav12','','images/btNodes_o.gif',1"
<div id="nav1" style="position:absolute; left:54px; top:221px; z-index:0"><a onMouseEvent="MM_swapImgRestore()" onmouseover="MM_swapImage('nav13','','images/btZonesets_o.gif',1"
<div id="nav1" style="position:absolute; left:54px; top:254px; z-index:0"><a onMouseEvent="MM_swapImgRestore()" onmouseover="MM_swapImage('nav14','','images/btZonesets_o.gif',1"
<div id="nav1" style="position:absolute; left:54px; top:287px; z-index:0"><a onMouseEvent="MM_swapImgRestore()" onmouseover="MM_swapImage('nav15','','images/btFind_o.gif',1"
<!-- pozadí -->
<hr style="background-color: #cccccc; border: 1px solid #000; height: 1px; width: 100%; margin: 0;"/>
32: 1 / 83 [58131]
```

Obrázek 23: Kód menu aplikace v JavaScriptu


Zdroj: vlastní zpracování

Názvy a funkce tlačítek odpovídá prostředkům, které správce SAN spravuje a konfiguruje. Jedná se o síť (Fabric), aktivní prvky SAN, především přepínače (Devices), koncová zařízení připojovaná do SAN, především disková pole a servery (Nodes), zóny sdružující koncová zařízení a zajišťující jejich vzájemnou viditelnost (Zones) a zonesety sdružující jednotlivé zóny (Zonesets).

5.1.4 Napojení na databázi

Jazyk PHP obsahuje řadu příkazů a knihoven pro práci s databází. Základní knihovna pro připojení k serveru MySQL se jmenuje *mysql*. Připojení k databázi E17542 se uskutečňuje téměř v každé volané stránce aplikace, a to pomocí procedurální syntaxe příkazů knihovny. Obrázek 24 ukazuje část zdrojového kódu úvodní obrazovky představující připojení k MySQL serveru (příkaz *mysql_connect*), výběr databáze E17542 (příkaz *mysql_select_db*) a výpis obsahu databázové tabulky Fabric (příkaz *mysql_query*). Na konci práce s databází je připojení ukončeno a uvolněn tak použitý přístup (příkaz *mysql_close*).

Odkaz Devices spustí skript, který vypíše z databázové tabulky Device všechny záznamy náležící zvolené Fabric. Podobně odkaz Nodes spustí skript, který vypíše z databázové tabulky Nodes všechna koncová zařízení náležící zvolené Fabric. Obrázek 26 ukazuje výpis koncových zařízení definovaných ve Fabric_SAN1.



Dokumentace SAN

Fabric_SAN1 : Nodes

N	Fabric	Alias	WWN	Pripojeni	Popis
1	1	HK9TE-21_p1	5002.77a4.100c.4e21		
3	1	Oracle_p2			
6	1	HK9VM-002_p1			
8	1	HK9VM-003_p1			
9	1	HK9VM-004_p1			
13	1	HK9TE-21_p2	5002.77a4.100c.4e22		
14	1	Oracle_SA_p1			
16	1	Oracle_SB_p1			
18	1	HK9TE-21_p3	5002.77a4.100c.4e23		
20	1	Dell_p1			
22	1	HK9VM-001_p1			

Obrázek 26: Výpis koncových zařízení definovaných ve Fabric_SAN1

Zdroj: vlastní zpracování

Pomocí odkazu Edit lze daný záznam upravit, pomocí odkazu Delete lze daný záznam z databázové tabulky smazat. Pod tabulkou lze ještě najít odkaz Add Fabric pro vytvoření nového záznamu. Přidání a editace záznamu probíhá pomocí formuláře, pro přidání se zobrazí prázdný formulář, pro editaci se zobrazí formulář s vyplněnými existujícími hodnotami (viz Obrázek 27).



Dokumentace SAN

Fabric - uprava záznamu

Odeslat
(Stisknete az po vyplneni formulare)

Uprava záznamu c. 1

Nazev:
Fabric_SAN1

Popis:
dvojice SAN switchu Cisco MDS9148

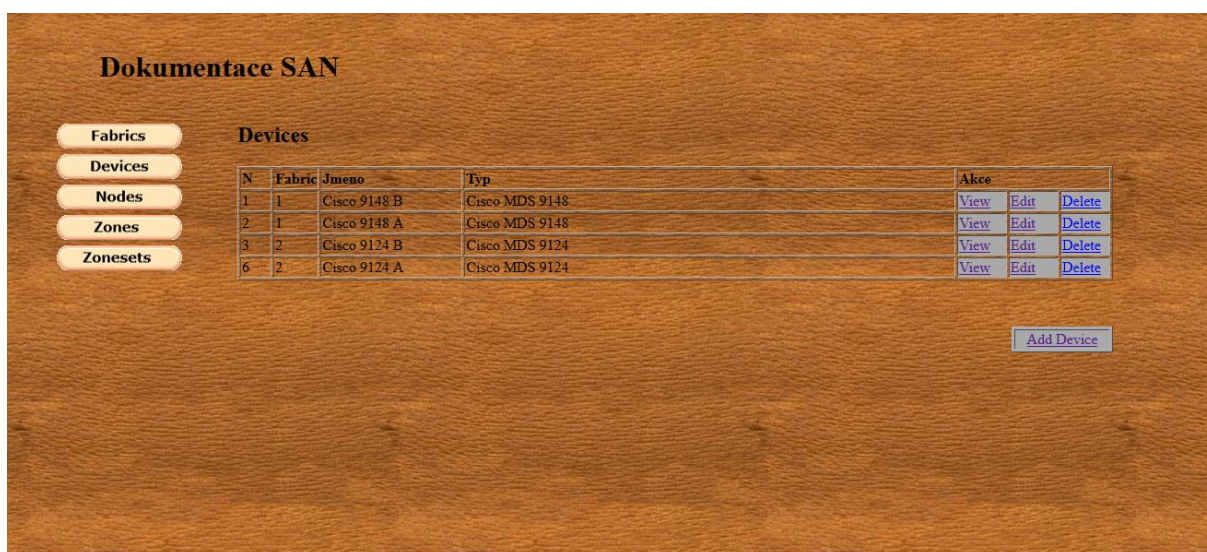
* povinná položka

Obrázek 27: Úprava záznamu

Zdroj: vlastní zpracování

Odeslání formuláře spustí skript, který uloží hodnoty z formuláře do databáze. Pro každý typ formuláře (přidání záznamu, úprava, fabric, device, nod, zone, zoneset, portdevice) existuje specifický ukládací skript.

Další čtyři základní okna (Devices, Nodes, Zones, Zonesets) mají podobnou strukturu. Vypisují seznamy z příslušných databázových tabulek. Od okna Fabrics se liší odkazy umístěnými v tabulce za každým záznamem – zde jsou odkazy View, Edit a Delete. Odkaz Edit a odkaz Delete mají stejnou funkci jako v okně Fabrics, odkaz View spouští skript, který zobrazí detail vybraného záznamu (viz Obrázek 28).



Obrázek 28: Okno Devices

Zdroj: vlastní zpracování

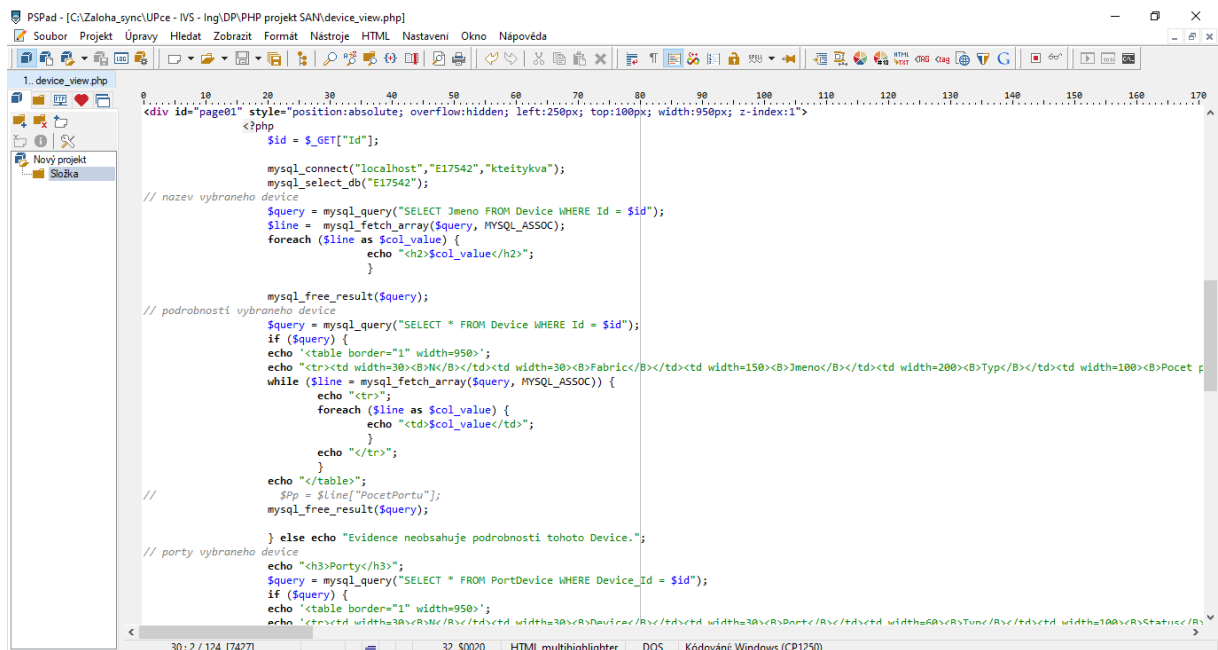
V případě okna Devices odkaz View spustí skript `device_view.php`. Skript `device_view.php` zobrazí všechny dostupné informace k danému záznamu a také porty náležící k vybranému device. Ty jsou zobrazeny ve formě tabulky s možností editace každého z nich - odkaz Edit (viz Obrázek 29). Záznamy představující jednotlivé porty vybraného zařízení jsou uloženy v databázové tabulce PortDevice. Pokud k vybranému zařízení tyto záznamy chybí, lze je vygenerovat pomocí odkazu Generate uvedeným na konci stránky. Vygenerování příslušných portů probíhá na základě hodnoty počtu portů uvedené v záznamu vybraného device.



Obrázek 29: Okno detailu vybraného device

Zdroj: vlastní zpracování

Obrázek 30 ukazuje na části kódu skriptu `device_view.php` jaké dotazy do jakých databázových tabulek obsahuje. Jedná se o dotaz do databáze na jméno vybraného device, dotaz na podrobnosti o vybraném zařízení a dotaz na seznam portů definovaných pro dané zařízení.



Obrázek 30: Ukázka kódu skriptu `device_view`

Zdroj: vlastní zpracování

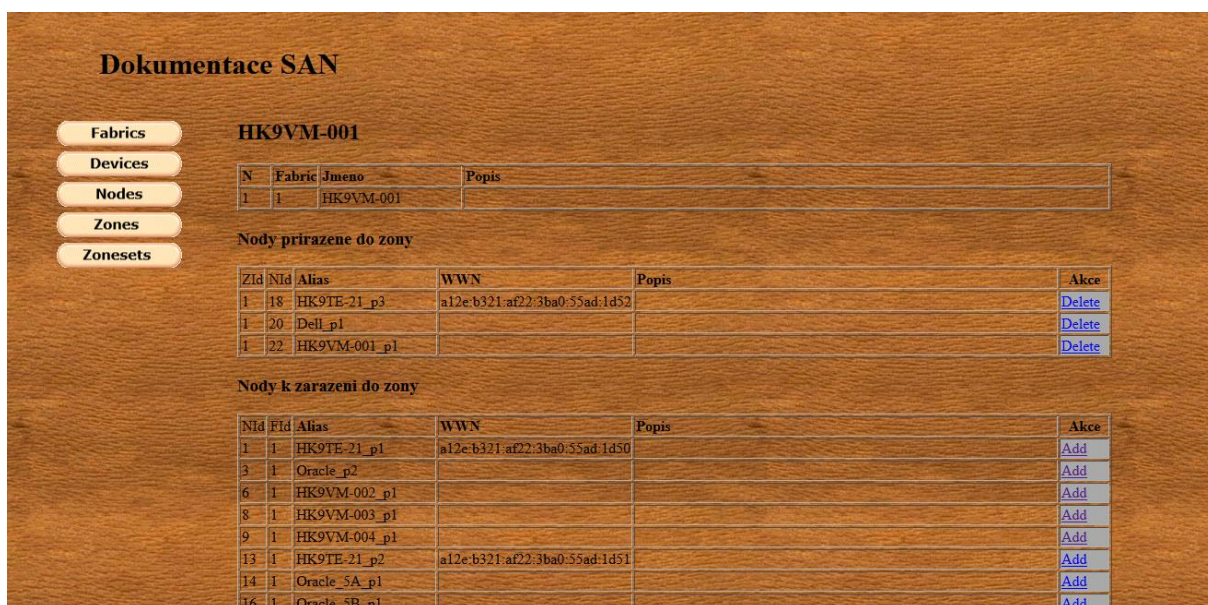
V případě okna Nodes odkaz View spustí skript nodes_view.php, který zobrazí podrobnosti k vybranému koncovému zařízení (viz Obrázek 31). Jedná se o jednoduchý dotaz do databázové tabulky Nodes.



Obrázek 31: Okno detailu vybraného nodu

Zdroj: vlastní zpracování

Detail Zóny zobrazí odkaz View v okně Zones pomocí skriptu zones_view.php. Kromě údajů k vybrané zóně se v okně zobrazí obsah zóny, tj. která koncová zařízení (nody) jsou do zóny zařazeny, s možností pomocí odkazu Delete je z dané zóny odstranit. Zároveň se zobrazí seznam koncových zařízení registrovaných ve stejné Fabric, s možností zařazení do prohlížené zóny pomocí odkazu Add. Vzhled okna je představen na Obrázku 32.



Obrázek 32: Okno detailu vybrané zóny

Zdroj: vlastní zpracování

Dotazy do databáze v tomto skriptu už nejsou tak jednoduché. Pro výpis obsahu zóny stejně jako výpisu možných koncových zařízení k zařazení do zóny je potřeba použít spojení více databázových tabulek. Obrázek 33 ukazuje část zdrojového kódu skriptu zones_view.php, který zajišťuje výpis obsahu zóny a výpis koncových zařízení ze stejné Fabric, ve které je registrována prohlížená zóna. Pro výpis obsahu zóny spojuje v databázovém dotazu databázové tabulky PolZone a Nod, pro výpis koncových zařízení pak databázové tabulky Zone a Nod.

```

<!-- pozadí -->
<body background="images/209_lgtbark2.jpg" bgProperties="fixed">

<!-- nadpis -->
<div id="text01" style="position:absolute; overflow:hidden; left:100px; top:20px; width:650px; height:100px; z-index:0">
<H1>Dokumentace SAN</H1</div>

<!-- menu -->
<div id="nav1" style="position:absolute; left:54px; top:122px; z-index:0"><a onMouseOut="WM_swapImgRestore()" onMouseOver="WM_swapImage('nav10','','images/btFabric_o.gif',1)" href="fabriccs.php"><img name="nav10" onLoad="WM_preload"
<div id="nav1" style="position:absolute; left:54px; top:155px; z-index:0"><a onMouseOut="WM_swapImgRestore()" onMouseOver="WM_swapImage('nav11','','images/btDevices_o.gif',1)" href="devices.php"><img name="nav11" onLoad="WM_preload"
<div id="nav1" style="position:absolute; left:54px; top:188px; z-index:0"><a onMouseOut="WM_swapImgRestore()" onMouseOver="WM_swapImage('nav12','','images/btNodes_o.gif',1)" href="nodes.php"><img name="nav12" onLoad="WM_preload"
<div id="nav1" style="position:absolute; left:54px; top:221px; z-index:0"><a onMouseOut="WM_swapImgRestore()" onMouseOver="WM_swapImage('nav13','','images/btZones_o.gif',1)" href="zones.php"><img name="nav13" onLoad="WM_preload"
<div id="nav1" style="position:absolute; left:54px; top:254px; z-index:0"><a onMouseOut="WM_swapImgRestore()" onMouseOver="WM_swapImage('nav14','','images/btZonesets_o.gif',1)" href="zonesets.php"><img name="nav14" onLoad="WM_preload"
<div id="nav1" style="position:absolute; left:54px; top:287px; z-index:0"><a onMouseOut="WM_swapImgRestore()" onMouseOver="WM_swapImage('nav15','','images/btFind_o.gif',1)" href="find.php"><img name="nav15" onLoad="WM_preload"

<div id="page01" style="position:absolute; overflow:hidden; left:250px; top:100px; width:950px; z-index:1">
<?php
    $id = $_GET["id"];
    mysql_connect("localhost","E17542","kteitykva");
    mysql_select_db("E17542");

// nazev vybrane zony
    $query = mysql_query("SELECT Zmeno FROM Zone WHERE Id = '$id'");
    $line = mysql_fetch_array($query, MYSQL_ASSOC);
    foreach ($line as $col_value) {
        echo "<h2>$col_value</h2>";
    }
    mysql_free_result($query);

// podrobnosti vybrane zony
    $query = mysql_query("SELECT * FROM Zone WHERE Id = '$id'");
    echo <table border="1" width=950>;
    echo <tr><td width=30><B>K</B></td><td width=30><B>Fabric</B></td><td width=150><B>Zmeno</B></td><td><B>Popis</B></td></tr>;
    while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
        echo <tr>;
        foreach ($line as $col_value) {
            echo <td>$col_value</td>;
        }
        echo </tr>;
    }
    echo </table>;
    mysql_free_result($query);
    } else echo "evidence neobsahuje podrobnosti teto Zone.";

// obsah vybrane zony
    echo <h3>Nody prirazene do zony</h3>;
    $query = mysql_query("SELECT PolZone.Zone_Id, PolZone.Nod_Id, Nod.Alias, Nod.MWn, Nod.Popis FROM PolZone, Nod WHERE PolZone.Zone_Id = $id AND PolZone.Nod_Id = Nod.Id AND PolZone.Fabric_Id = Nod.Fabric_Id");
    if ($query) {
        echo <table border="1" width=950>;
        echo <tr><td width=20>Zid</td><td width=20>Mid</td><td width=150><B>Alias</B></td><td width=150><B>MWn</B></td><td width=50><B>Popis</B></td><td width=50 align="center"><B>Akce</B></td></tr>;
        while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
            echo <tr>;
            foreach ($line as $col_value) {
                echo <td>$col_value</td>;
            }
            echo <td width=50 bgcolor=#A9A9A9><a href="polzone_delete.php?Zid=".$line["Zone_Id"]."&Mid=".$line["Nod_Id"]."&Id=".$line["Nod_Id"]."&Delete=1">Delete</a></td>;
        }
        echo </table>;
        mysql_free_result($query);
    } else echo "Zone neobsahuje zadne Nody.";

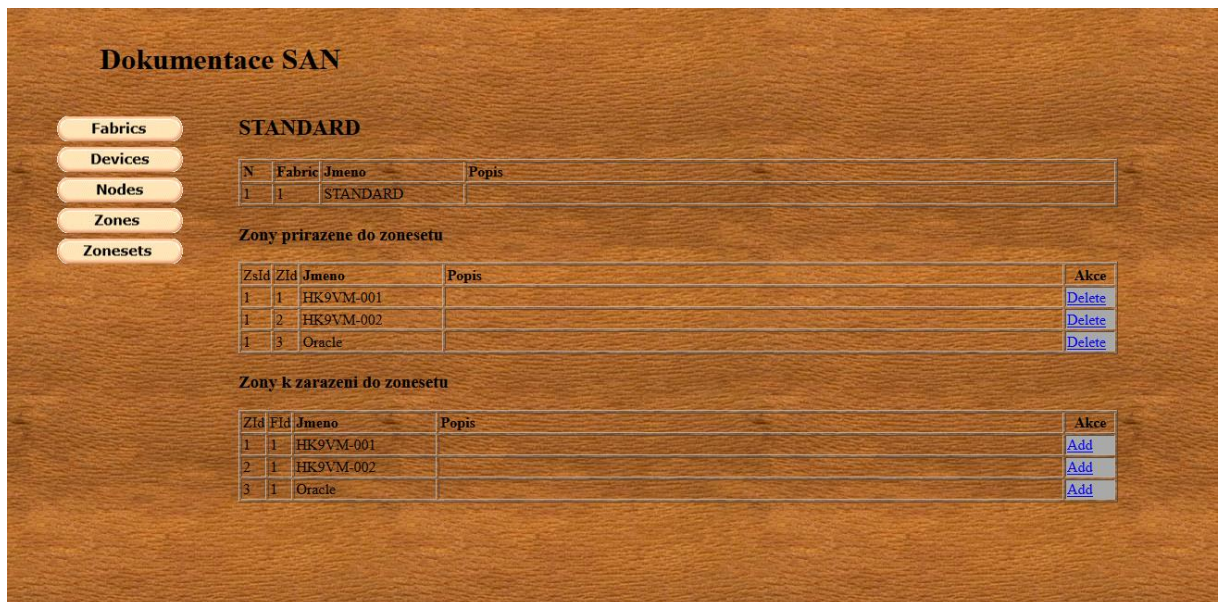
// nody dostupne k zarazení do zony
    echo <h3>Nody k zarazení do zony</h3>;
    $query = mysql_query("SELECT Nod.Id, Nod.Fabric_Id, Nod.Alias, Nod.MWn, Nod.Popis FROM Zone, Nod WHERE Zone.Id = $id AND Zone.Fabric_Id = Nod.Fabric_Id");
    if ($query) {
        echo <table border="1" width=950>;
        echo <tr><td width=20>Mid</td><td width=20>Fid</td><td width=150><B>Alias</B></td><td width=150><B>MWn</B></td><td width=50><B>Popis</B></td><td width=50 align="center"><B>Akce</B></td></tr>;
        while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
            echo <tr>;
            foreach ($line as $col_value) {
                echo <td>$col_value</td>;
            }
            echo <td width=50 bgcolor=#A9A9A9><a href="polzone_add.php?Zid=".$id."&Mid=".$line["Id"]."&Fid=".$line["Fabric_Id"]."&Add=1">Add</a></td>;
        }
        echo </table>;
        mysql_free_result($query);
    } else echo "Nesou k dispozici zadne Nody.";
    mysql_close();
?>

```

Obrázek 33: Ukázka kódu skriptu zones_view

Zdroj: vlastní zpracování

Odkaz View v okně Zonesets spouští skript zoneset_view.php. Podobně jako v případě okna Zones zobrazí podrobnosti vybraného zonesetu, výčet zón obsažených v zonesetu i zóny, které je možné do zonesetu zařadit. Opět je možné zóny ze zonesetu odstraňovat pomocí odkazu Delete i přidávat pomocí odkazu Add (viz Obrázek 34).



Obrázek 34: Okno detailu vybraného zonesetu

Zdroj: vlastní zpracování

Ve skriptu zoneset_view.php je znovu použito spojení více databázových tabulek. Obrázek 34 ukazuje část zdrojového kódu zajišťující výpis obsahu zonesetu a výpis zón ze stejné Fabric, které lze případně do prohlíženého zonesetu zařadit. Pro výpis obsahu zonesetu spojuje v databázovém dotazu databázové tabulky PolZoneset a Zone, pro výpis zón pak databázové tabulky Zoneset a Zone (viz Obrázek 35).

```

<?php
$id = $_GET["id"];
mysql_connect("localhost","E17542","kteilykva");
mysql_select_db("E17542");
// nazev vybraného zonesetu
$query = mysql_query("SELECT Jmeno FROM Zoneset WHERE Id = '$id'");
while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
    echo "<td>$col_value</td>";
}
mysql_free_result($query);
// podrobnosti vybraného zonesetu
$query = mysql_query("SELECT * FROM Zoneset WHERE Id = '$id'");
if ($query) {
    echo "<table border='1' width='95%'>";
    echo "<tr><td width=30%>N</td><td width=30%>Fabric</td><td width=15%>Jmeno</td><td width=15%>Popis</td></tr>";
    while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
        echo "<tr>";
        foreach ($line as $col_value) {
            echo "<td>$col_value</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
    mysql_free_result($query);
} else echo "Evidence neobsahuje podrobnosti tohoto Zoneset.";
// obsah vybraného zonesetu
echo "<h3>Zony prirazene do zonesetu</h3>";
$query = mysql_query("SELECT PolZoneset.Zoneset_Id, PolZoneset.Zone_Id, Zone.Jmeno, Zone.Popis FROM PolZoneset, Zone WHERE PolZoneset.Zoneset_Id = '$id' AND PolZoneset.Zone_Id = Zone.Id AND PolZoneset.Fabric_Id = Zone.Fabric_Id");
if ($query) {
    echo "<table border='1' width='95%'>";
    echo "<tr><td width=20%>ZsId</td><td width=20%>Zid</td><td width=15%>Jmeno</td><td width=15%>Popis</td><td width=50% align='center'><a href='\"#\"'>Delete</a></td></tr>";
    while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
        echo "<tr>";
        foreach ($line as $col_value) {
            echo "<td>$col_value</td>";
        }
        echo "<td width=50% bgcolor=#A9A9A9><a href='\"polzoneset_delete.php?ZsId=$line[Zoneset_Id]&Zid=$line[Zone_Id]&\"'>Delete</a></td>";
        echo "</tr>";
    }
    echo "</table>";
    mysql_free_result($query);
} else echo "Zoneset neobsahuje zadne Zony.";
// zony dostupne k zarazení do zonesetu
echo "<h3>Zony k zarazení do zonesetu</h3>";
$query = mysql_query("SELECT Zone.Id, Zone.Fabric_Id, Zone.Jmeno, Zone.Popis FROM Zoneset, Zone WHERE Zoneset.Id = '$id' AND Zoneset.Fabric_Id = Zone.Fabric_Id");
if ($query) {
    echo "<table border='1' width='95%'>";
    echo "<tr><td width=20%>Zid</td><td width=20%>Fid</td><td width=15%>Jmeno</td><td width=15%>Popis</td><td width=50% align='center'><a href='\"#\"'>Add</a></td></tr>";
    while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
        echo "<tr>";
        foreach ($line as $col_value) {
            echo "<td>$col_value</td>";
        }
        echo "<td width=50% bgcolor=#A9A9A9><a href='\"polzoneset_add.php?ZsId=$id.&Zid=$line[Zid]&Fid=$line[Fabric_Id]&\"'>Add</a></td>";
        echo "</tr>";
    }
    echo "</table>";
    mysql_free_result($query);
} else echo "Nejsou k dispozici zadne Zony.";
mysql_close();
?>

```

Obrázek 35: Ukázka kódu skriptu zoneset_view

Zdroj: vlastní zpracování

5.2 Prototyp 2

Prototyp 2 je vylepšenou verzí prototypu 1, doplněnou o zapracované požadavky a připomínky uživatelů aplikace. Postřehy a podněty pro dopracování byly sbírány v rámci testovacího provozu, kdy aplikace ve verzi prototypu 1 byla nasazena a používána v reálném prostředí společnosti, pro jejíž datovou síť byl prototyp 1 vytvořen.

5.2.1 Doplnkové požadavky

Sběr doplňkových požadavků byl prováděn v průběhu nasazení aplikace ve verzi prototypu 1 do testovacího provozu. Vzhledem k mé pozici správce SAN ve společnosti Povodí Labe mi bylo umožněno aplikaci v tomto prostředí nasadit a zpřístupnit ji ostatním uživatelům, především kolegům inženýrům, kteří mne ve správě SAN zastupují. Z jejich strany pak vzešly podněty a připomínky, ze kterých byly definovány reálné doplňující požadavky na aplikaci.

Doplňující požadavky na aplikaci shrnuje Tabulka 11.

Tabulka 11: Seznam doplňkových požadavků na aplikaci

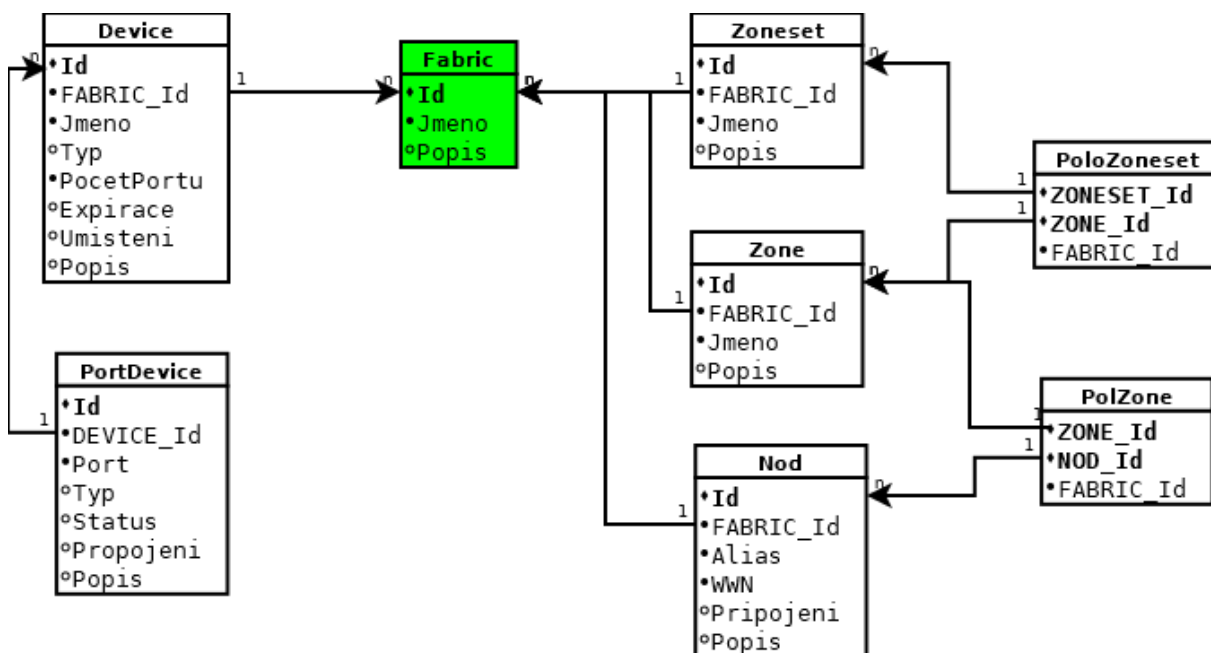
	Seznam požadavků na aplikaci
Vyhledávací požadavky	Vyhledání zařízení podle WWN
	Vyhledání zařízení podle názvu (Aliasu)
Ostatní požadavky	Doplnit k device atribut datum ukončení podpory (expirace)
	Atribut datum ukončení podpory zobrazit v přehledovém okně
	Doplnit možnost u detailu nodu vypsát jeho členství v zónách

Zdroj: vlastní zpracování

5.2.2 Změny v datovém modelu

Doplňující požadavky na prototyp 1 vyvolaly v datovém modelu pouze jedinou změnu – doplnění atributu Expirace do databázové tabulky Device.

Upravený relační datový model pro prototyp 2 ukazuje Obrázek 36.



Obrázek 36: Relační model prototypu 2

Zdroj: vlastní zpracování

Atributy databázové tabulky Device rozšířené o atribut Expirace a jejich vlastnosti shrnuje Tabulka 12.

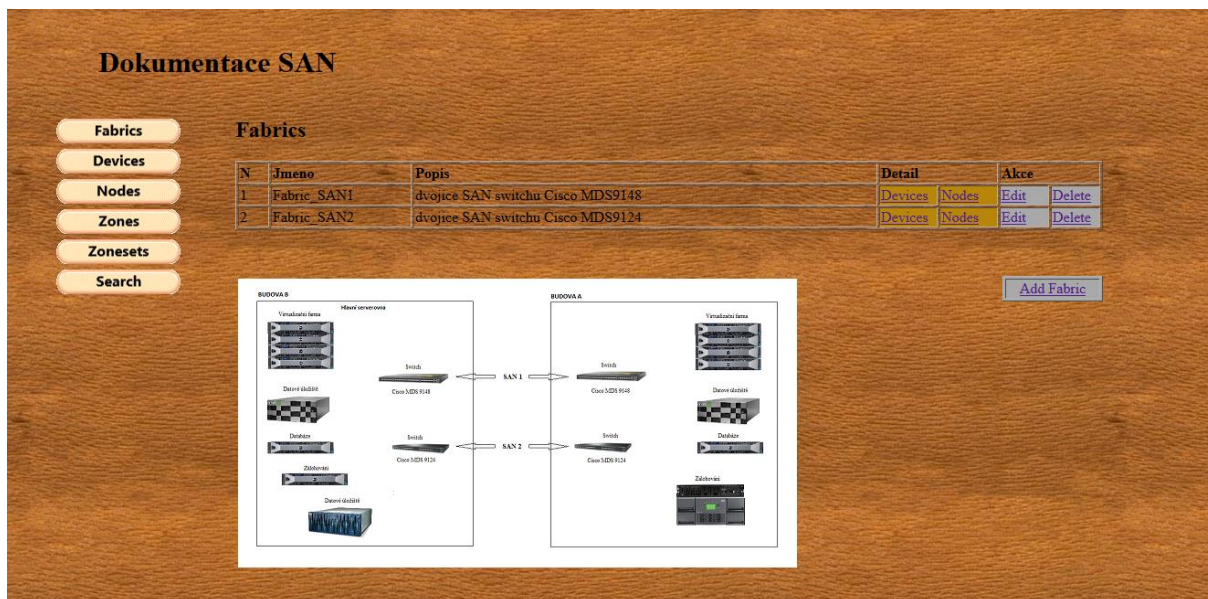
Tabulka 12: Struktura databázové tabulky Device

Databázová tabulka Device				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Fabric_Id	Int (11)		žádná	cizí
Jmeno	Text	utf8_czech_ci	žádná	
Typ	Text	utf8_czech_ci	žádná	
PocetPortu	Int (11)		žádná	
Expirace	Date		žádná	
Umisteni	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

5.2.3 Změny v aplikaci

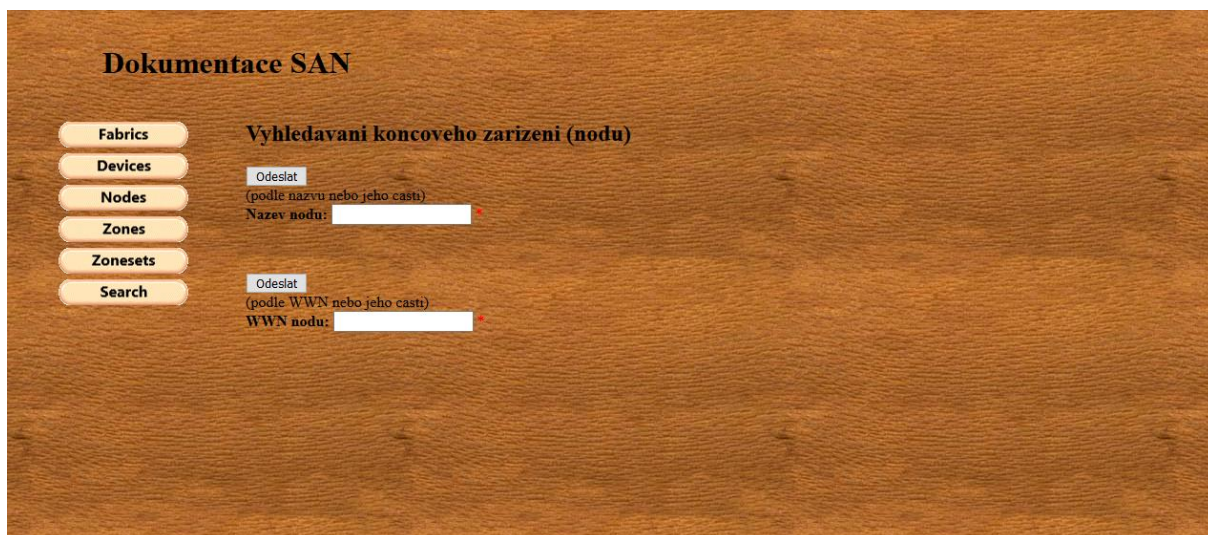
Na základě vyhledávacích požadavků bylo menu aplikace rozšířeno o tlačítko Search. Úvodní okno aplikace s rozšířeným menu ukazuje Obrázek 37.



Obrázek 37: Úvodní obrazovka prototypu 2

Zdroj: vlastní zpracování

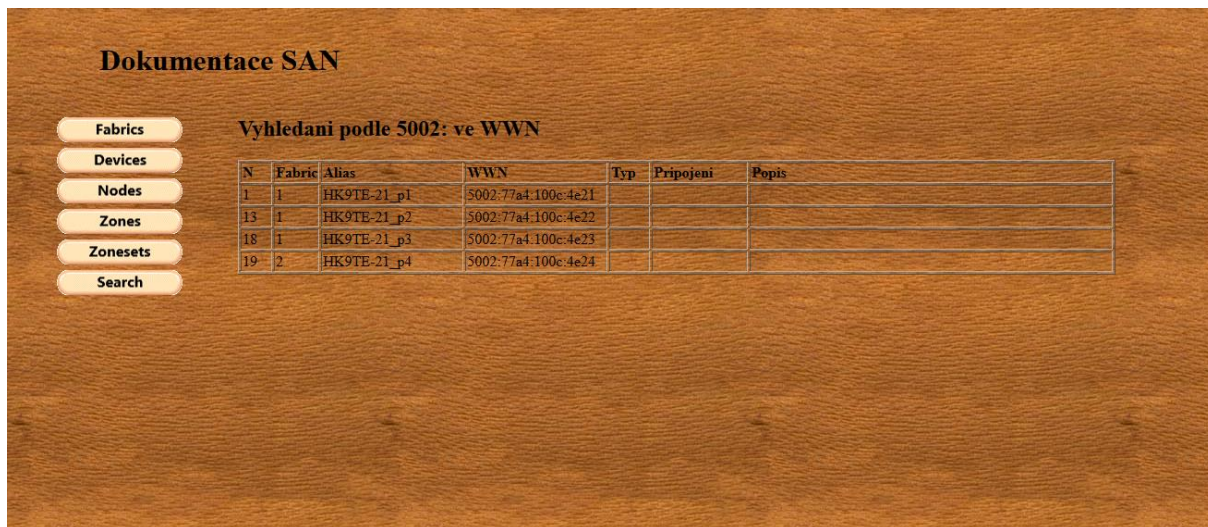
Tlačítko Search spouští skript find.php. Ten pomocí formuláře dává uživateli možnost vyhledávání zařízení podle WWN či podle názvu (aliasu). Okno aplikace pro vyhledávání ukazuje Obrázek 38. Okno obsahuje dvě tlačítka, jedno pro vyhledávání podle WWN, druhé pro vyhledávání podle názvu. Každé spouští jiný vyhledávací skript.



Obrázek 38: Okno pro vyhledávání

Zdroj: vlastní zpracování

Jako parametr vyhledávání lze uvést pouze část hledaného řetězce, skript vrátí seznam všech zařízení, v jejichž příslušných parametrech (alias, WWN) se tato část řetězce vyskytuje. Obrázek 39 ukazuje výsledek vyhledávání podle WWN, Obrázek 40 pak kód celého skriptu.



Obrázek 39: Okno s výsledkem vyhledávání podle WWN

Zdroj: vlastní zpracování

```

<head>
  <meta charset="UTF-8">
  <title>Dokumentace SAN</title>

  <script language="javascript" type="text/javascript">
function MW_swapImgRestore() {
  var i,x,a=document.MW_sr; for(i=0;&&i<a.length&&(x=a[i])&&x.oSrc;i++) x.src=x.oSrc;
}

function MW_preloadImages() {
  var d=document; if(d.images){ if(!d.MW_p) d.MW_p=new Array();
  var i,j=d.MW_p.length,a=MW_preloadImages.arguments; for(i=0; i<a.length; i++)
  if (a[i].indexOf("#")!=""&&d.MW_p[j]=new Image; d.MW_p[j].src=a[i];}}

function MW_findObj(n, d) {
  var p,i,j; if(!d) d=document; if((p=n.indexOf("?"))&&parent.frames.length) {
  d=parent.frames[n.substring(p+1)].document; n=n.substring(0,p);}
  if(!(x=d.n)&&d.all) x=d.all[n]; for (i=0;&&i<d.forms.length;i++) x=d.forms[i][n];
  for(i=0;&&d.layers&&i<d.layers.length;i++) x=MW_findObj(n,d.layers[i].document);
  if(!x && d.getElementById) x=d.getElementById(n); return x;
}

function MW_swapImage() {
  var i,j=0,x,a=MW_swapImage.arguments; document.MW_sr=new Array; for(i=0;i<a.length-2;i+=3)
  if ((x=MW_findObj(a[i]))!=null)(document.MW_sr[j++]x; if(!x.oSrc) x.oSrc=a[i+2]);
}

//
</script>
</head>

<!-- pozadi -->
<body background="images/209_lgtbar42.jpg" bgProperties="fixed">

<!-- nadpis -->
<div id="text01" style="position:absolute; overflow:hidden; left:100px; top:20px; width:650px; height:100px; z-index:0">
<h1>Dokumentace SAN</h1></div>

<!-- menu -->
<div id="nav1" style="position:absolute; left:54px; top:122px; z-index:0"><a onmouseover="MW_swapImgRestore()" onmouseout="MW_swapImage('nav1','','images/btFabric_o.gif','1') href="fabrics.php"><img name="nav18" onload="MW_preloadImages('images/btFabric_o.gif','1') href="fabrics.php"></a>
<div id="nav1" style="position:absolute; left:54px; top:155px; z-index:0"><a onmouseover="MW_swapImgRestore()" onmouseout="MW_swapImage('nav11','','images/btDevices_o.gif','1') href="devices.php"><img name="nav11" onload="MW_preloadImages('images/btDevices_o.gif','1') href="devices.php"></a>
<div id="nav1" style="position:absolute; left:54px; top:188px; z-index:0"><a onmouseover="MW_swapImgRestore()" onmouseout="MW_swapImage('nav12','','images/btNodes_o.gif','1') href="nodes.php"><img name="nav12" onload="MW_preloadImages('images/btNodes_o.gif','1') href="nodes.php"></a>
<div id="nav1" style="position:absolute; left:54px; top:221px; z-index:0"><a onmouseover="MW_swapImgRestore()" onmouseout="MW_swapImage('nav13','','images/btZones_o.gif','1') href="zones.php"><img name="nav13" onload="MW_preloadImages('images/btZones_o.gif','1') href="zones.php"></a>
<div id="nav1" style="position:absolute; left:54px; top:254px; z-index:0"><a onmouseover="MW_swapImgRestore()" onmouseout="MW_swapImage('nav14','','images/btZonesets_o.gif','1') href="zonesets.php"><img name="nav14" onload="MW_preloadImages('images/btZonesets_o.gif','1') href="zonesets.php"></a>
<div id="nav1" style="position:absolute; left:54px; top:287px; z-index:0"><a onmouseover="MW_swapImgRestore()" onmouseout="MW_swapImage('nav15','','images/btFind_o.gif','1') href="find.php"><img name="nav15" onload="MW_preloadImages('images/btFind_o.gif','1') href="find.php"></a>

<div id="page01" style="position:absolute; overflow:hidden; left:250px; top:100px; width:950px; z-index:1">
  <pre>
$wm = trim($_POST['wm']);
echo "<h2>Vyhledani podle $wm ve MM</h2>";

mysql_connect("localhost","E17542","krtelkyva");
mysql_select_db("E17542");

// podrobnosti vyhledaneho device
$query = mysql_query("SELECT * FROM Nod WHERE MMN LIKE '%$wm%'");
if ($query) {
echo "<table border='1' width='950'>";
echo "<tr><td width='30'>BMM</td><td width='30'>BFabric</td><td width='150'>BAlias</td><td width='150'>BMM</td><td width='40'>BTyp</td><td width='100'>BPripojeni</td><td><td>BPopis</td></tr>";
while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
echo "<tr>";
foreach ($line as $col_value) {
echo "<td>$col_value</td>";
}
echo "</tr>";
}
echo "</table>";
mysql_free_result($query);
} else echo "Evidence neobsahuje Nod s timto MMN.";
}
  </pre>
</div>
</body>

```

Obrázek 40: Kód skriptu pro vyhledávání podle WWN

Zdroj: vlastní zpracování

Na základě požadavku doplnění atributu Expirace a požadavku jeho zobrazení v přehledu Devices došlo ke změně v okně Devices, Device_View (detail vybraného device) a v okně Fabric_Devices (výpis devices v dané fabric). Obrázek 41 ukazuje druhé ze zmíněných oken.

Dokumentace SAN

Fabrics Cisco 9124 B

N	Fabric	Jmeno	Typ	Porty	Expirace	Umístění	Popis
3	2	Cisco 9124 B	Cisco MDS 9124	24	2019-01-31	Budova B	expirace záruky 1/2019

Porty

N	Device	Port	Typ	Status	Propojení	Popis	Akce
1	3	1	E	Aktiv		propojení switchu	Edit
2	3	2	N	Aktiv			Edit
3	3	3	N	Aktiv			Edit
4	3	4	N	Aktiv			Edit
5	3	5	E	Aktiv		propojení switchu	Edit
6	3	6	N	Aktiv			Edit
7	3	7	N	Aktiv			Edit
8	3	8	N	Aktiv			Edit
9	3	9					Edit
10	3	10					Edit

Obrázek 41: Okno Device_View s doplněným atributem

Zdroj: vlastní zpracování

Změn doznaly díky těmto požadavkům nejen skripty vykreslující uvedená okna, ale i skripty pro úpravu (editaci) vybraného device, pro přidání nového a skripty pro uložení dat do databáze.

Na základě požadavku dopracování funkcionality, která by zobrazila u vybraného koncového zařízení všechny zóny, ve kterých se dané zařízení (WWN, alias) vyskytuje, byl v okně detailu vybraného zařízení (okno Nodes_View) doplněn odkaz In Zones (viz Obrázek 42).

Dokumentace SAN

Fabrics HK9TE-21_p1

N	Fabric	Alias	WWN	Typ	Pripojeni	Popis	Akce
1	1	HK9TE-21_p1	5002:77a4:100c:4e21				In Zones

Obrázek 42: Okno Nodes_View s doplněným odkazem In Zones

Zdroj: vlastní zpracování

Odkaz In Zones spouští skript nodes_inzones.php, který vypíše zóny obsahující vybrané zařízení. Podobně jako u výpisu obsahu zóny nebo obsahu zonesetu je použito při SQL dotazu spojování tabulek. V tomto případě se spojují v databázovém dotazu databázové tabulky PolZone a Zone. Obrázek 43 ukazuje část zdrojového kódu tohoto skriptu obsahující zmíněné dotazování.

```

<div id="page01" style="position:absolute; overflow:hidden; left:250px; top:100px; width:950px; z-index:1">
  <?php
    $id = $_GET["Id"];
    mysql_connect("localhost","E17542","kteitykva");
    mysql_select_db("E17542");
    // nazev vybraného nodu
    $query = mysql_query("SELECT Alias FROM Nod WHERE Id = $id");
    $line = mysql_fetch_array($query, MYSQL_ASSOC);
    foreach ($line as $col_value) {
      echo "<h2>Zony obsahující $col_value</h2>";
    }
    mysql_free_result($query);
    // vypis zon
    $query = mysql_query("SELECT PolZone.Nod_Id, PolZone.Zone_Id, Zone.Jmeno, Zone.Popis FROM PolZone, Zone WHERE PolZone.Nod_Id = $id AND PolZone.Zone_Id = Zone.Id");
    if ($query) {
      echo "<table border='1' width=950'>";
      echo "<tr><td width=20>Nid</td><td width=20>Zid</td><td width=150><B>Jmeno</B></td><td><B>Popis</B></td></tr>";
      while ($line = mysql_fetch_array($query, MYSQL_ASSOC)) {
        echo "<tr>";
        foreach ($line as $col_value) {
          echo "<td>$col_value</td>";
        }
        echo "</tr>";
      }
      echo "</table>";
      mysql_free_result($query);
    } else echo "Nod není v zadne zone.";
    mysql_close();
  ?>
</div>

```

Obrázek 43: Zdrojový kód

Zdroj: vlastní zpracování

Výsledkem provedení skriptu nodes_inzones.php je tabulkový výpis zón obsahujících vybraný nod. Okno s požadovaným výpisem zón ukazuje Obrázek 44.



Obrázek 44: Okno Nodes_View_Zone

Zdroj: vlastní zpracování

5.3 Prototyp 3

Aby aplikace mohla být použitelná pro všechny podtypy a topologie Switched Fabric, bylo třeba prototyp 2 rozšířit tak, aby aplikace mohla zdokumentovat libovolnou topologickou variantu. Prototyp 3 je rozšířenou verzí prototypu 2, doplněnou o případné zvláštnosti jednotlivých topologií, které by prototyp 2 nebyl schopen dostatečně zdokumentovat. Potřebná rozšíření aplikace byla určena analýzou podtypů sítě Switched Fabric představených v kapitole 2.

5.3.1 Analýza podtypů

Podtypy sítě Switched Fabric představené v kapitole 2 byly přezkoumány z pohledu možnosti jejich dokumentace pomocí aplikace ve verzi prototypu 2. Single Switch topologie je jednoduchá topologie s jedním SAN switchem. Jako taková je jistě dokumentovatelná prototypem 2. Pro Cascade/Ring topologii byl prototyp 2 (prototyp 1) přímo vyvíjen. Mesh topologie vyznačující se oproti předchozím topologiím pouze více propojeními mezi switchi je také jistě dokumentovatelná prototypem 2. Vrstvené topologie obsahují hlavní switch (Core switch) a ostatní switche (Edge switche). Některé Edge switche mohou být vyhrazeny pouze pro připojení serverů nebo pouze pro připojení datových úložišť. Toto lze zdokumentovat v prototypu 2 buď pomocí názvů devices (switchů) nebo využitím atributu Popis u těchto prvků. Všechny podtypy sítě Switched Fabric tedy lze dokumentovat pomocí prototypu 2.

Součástí sítě Switched Fabric může být ale i další základní typ SAN – Arbitrated Loop. Obrázek 2 ukazuje topologii s touto smyčkou v úvodu kapitoly 2. Jedná se v podstatě o koncové zařízení, které má za sebou připojená další zařízení do smyčky. To prototyp 2 zdokumentovat neumí, proto je třeba aplikaci rozšířit o dokumentaci koncových zařízení typu smyčka. Tabulka 13 shrnuje rozšiřující požadavky.

Tabulka 13: Seznam rozšiřujících požadavků na aplikaci

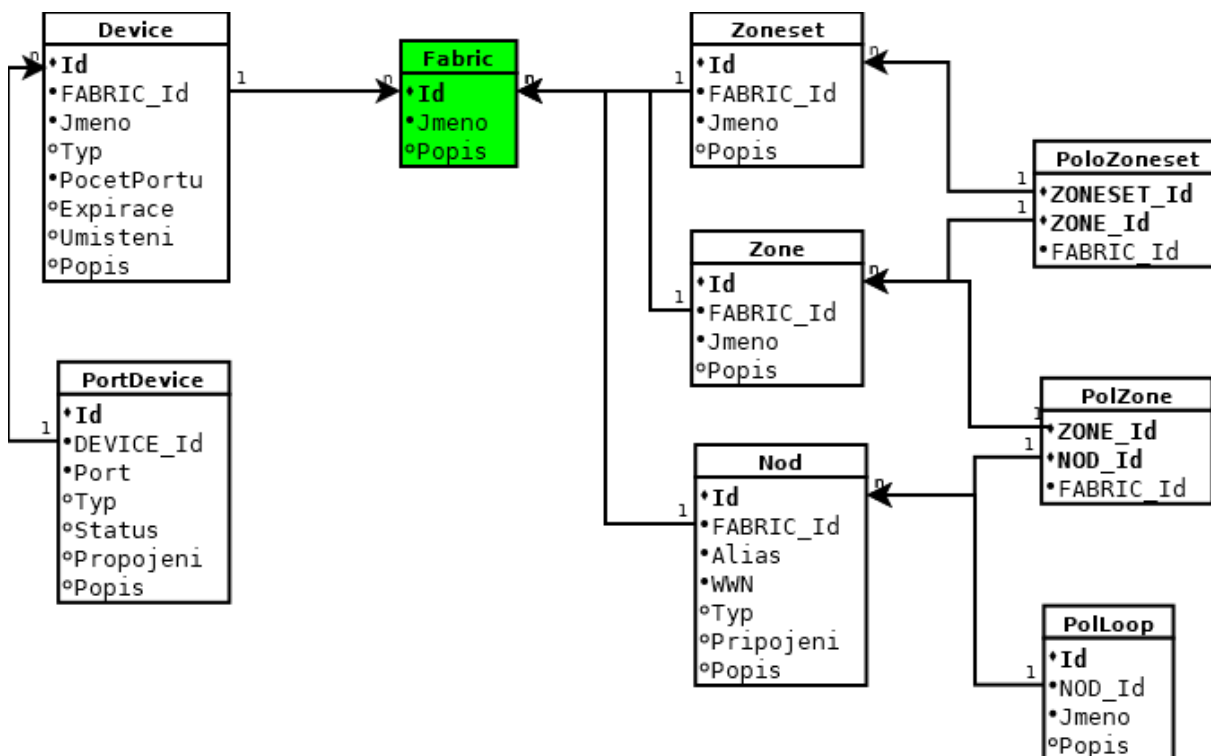
	Seznam rozšiřujících požadavků
Smyčky (loop) v topologii	Doplnění atributu Typ v databázové tabulce Nod
	Nová tabulka PolLoop pro evidenci zařízení ve smyčkách

Zdroj: vlastní zpracování

5.3.2 Změny v datovém modelu

Pro možnost zařazení smyčky (Loop) do topologie SAN byl relaci Nod přidán atribut Typ, označující, zda se jedná o jedno koncové zařízení nebo více zařízení ve smyčce. Zároveň byla

přidána relace PolLoop představující seznam zařízení zařazených do smyček (loop). Upravený relační datový model pro prototyp 3 ukazuje Obrázek 45.



Obrázek 45: Relační model prototypu 3

Zdroj: vlastní zpracování

Strukturu databázové tabulky Nod rozšířenou o atribut Typ ukazuje Tabulka 14.

Tabulka 14: Struktura databázové tabulky Nod

Databázová tabulka Nod				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Fabric_Id	Int (11)		žádná	cizí
Alias	Text	utf8_czech_ci	žádná	
WWN	Text	utf8_czech_ci	žádná	
Typ	Text	utf8_czech_ci	žádná	
Pripojeni	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

Strukturu nové databázové tabulky PolLoop sloužící k evidenci koncových zařízení zapojených ve smyčce ukazuje Tabulka 15.

Tabulka 15: Struktura nové databázové tabulky PolLoop

Databázová tabulka Nod				
Název	Typ	Znaková sada	Výchozí hodnota	Klíč
Id	Int (11)		žádná	primární
Nod_Id	Int (11)		žádná	cizí
Jmeno	Text	utf8_czech_ci	žádná	
Popis	Text	utf8_czech_ci	žádná	

Zdroj: vlastní zpracování

5.3.3 Změny v aplikaci

Na základě doplnění atributu Typ v tabulce Nod a vytvoření nové tabulky PolLoop došlo ke změně v okně Nodes, Nodes_View (detail vybraného koncového zařízení) a v okně Fabric_Nodes (výpis nodů v dané fabric).

Přehledové okno Nodes bylo doplněno o zobrazení atributu Typ, určující, zda daný nod je smyčkou či nikoli. Obrázek 46 ukazuje okno Nodes v prototypu 3, kde nod ZALOHY je typu smyčka (loop).

N	Fabric	Alias	WWN	Typ	Popis	Akce
1	1	HK9TE-21_p1	5002.77a4.100c.4a21			View Edit Delete
2	2	Oracle_p1				View Edit Delete
3	1	Oracle_p2				View Edit Delete
6	1	HK9VM-002_p1				View Edit Delete
7	2	HK9VM-002_p2				View Edit Delete
9	1	HK9VM-004_p1				View Edit Delete
11	2	HK9VM-004_p2				View Edit Delete
13	1	HK9TE-21_p2	5002.77a4.100c.4a22			View Edit Delete
18	1	HK9TE-21_p3	5002.77a4.100c.4a23			View Edit Delete
19	2	HK9TE-21_p4	5002.77a4.100c.4a24			View Edit Delete
20	1	Dell_p1				View Edit Delete
21	2	Dell_p2				View Edit Delete
22	1	HK9VM-001_p1				View Edit Delete
23	1	ZALOHY		loop		View Edit Delete

Obrázek 46: Okno Nodes v prototypu 3

Zdroj: vlastní zpracování

Větších změn doznalo okno Nodes_View zobrazující detaily vybraného koncového zařízení. V případě, že vybrané zařízení má v atributu Typ uvedeno hodnotu „loop“, skript nodes_view.php se pokusí vypsát z databázové tabulky PolLoop položky přiřazené vybranému zařízení. Případné nalezené záznamy jsou vypsány formou tabulky s možností odstranění jednotlivých záznamů pomocí odkazu Delete. Jako poslední prvek na stránce detailu koncového zařízení typu Loop je odkaz Add pro možnost vytvoření záznamu v databázové tabulce PolLoop svázaného s prohlíženým koncovým zařízením (viz Obrázek 47).



Obrázek 47: Okno detailu vybraného nodu typu smyčka

Zdroj: vlastní zpracování

Změn doznaly díky těmto požadavkům nejen skripty vykreslující uvedená okna, ale i skripty pro úpravu (editaci) vybraného nodu, pro přidání nového a skripty pro uložení dat do databáze.

5.4 Dílčí souhrn

Aplikace byla vyvinuta metodou prototypování, konkrétně byly během vývoje aplikace vytvořeny tři prototypy. Prototyp 1 představoval aplikaci pro konkrétní SAN, vytvořenou podle znalostí a zkušeností správce této SAN. Prototyp 2 byla vyladěná verze prototypu 1, doplněná o reálné požadavky uživatelů aplikace na základě jejího testovacího provozu. Prototyp 3 je rozšířená verze prototypu 2, zohledňující charakteristiky různých topologií Switched Fabric SAN uvedených v kapitole 2 a umožňující dokumentaci tohoto typu sítě v libovolné topologii. Prototyp 3 představuje finální verzi vyvíjené aplikace.

6 VYHODNOCENÍ APLIKACE A POUŽITÉHO POSTUPU VÝVOJE

V této kapitole je stručně zhodnocena vyvinutá aplikace a postup, který byl při jejím vývoji použitý.

6.1 Využití, význam aplikace

Aplikace pro dokumentaci sítě SAN je navržena pro evidenci a poskytování informací o dokumentované síti SAN. Aplikace umožňuje rychle se orientovat v konfiguraci dané SAN a přehledně tuto konfiguraci zobrazit bez nutnosti administrátorského přístupu na jednotlivé prvky sítě a vyčítání informací o SAN z konfigurací jednotlivých prvků. Je alternativou k obdobným aplikacím velkých výrobců datových úložišť či SAN switchů, které jsou bez současného nákupu drahého hardwaru nedostupné.

6.2 Metoda Tří prototypů

Postup vývoje aplikace řadou prototypů, který byl použit pro vytvoření finální aplikace, vycházel ze specifických podmínek a možností, které jsem jako řešitel měl k dispozici. Především to byla možnost práce (správy) s konkrétní SAN a z toho vyplývající zkušenosti s touto sítí a znalost potřeb správce takové sítě. Úvodní prototyp (prototyp 1) byl tedy vytvořen pro tuto SAN. Dále to byla možnost nasadit aplikaci do testovacího provozu a získat potřebné podněty a náměty od testovacích uživatelů pro definování reálných doplňujících požadavků na aplikaci (prototyp 2). Pro zobecnění a rozšíření aplikace na možnost nasazení v libovolné topologii Switched Fabric SAN (prototyp 3) bylo z důvodu nedostupnosti jiných datových infrastruktur použito informací o možných podtypech Switched Fabric SAN, které byly shromážděny z dostupné literatury a technické dokumentace. Ve výsledku tedy pro vývoj finální aplikace bylo zapotřebí vytvoření pouze třech prototypů. Prototypy a předpoklady pro jejich vznik shrnuje Tabulka 16.

Tabulka 16: Vývoj aplikace pomocí tří prototypů

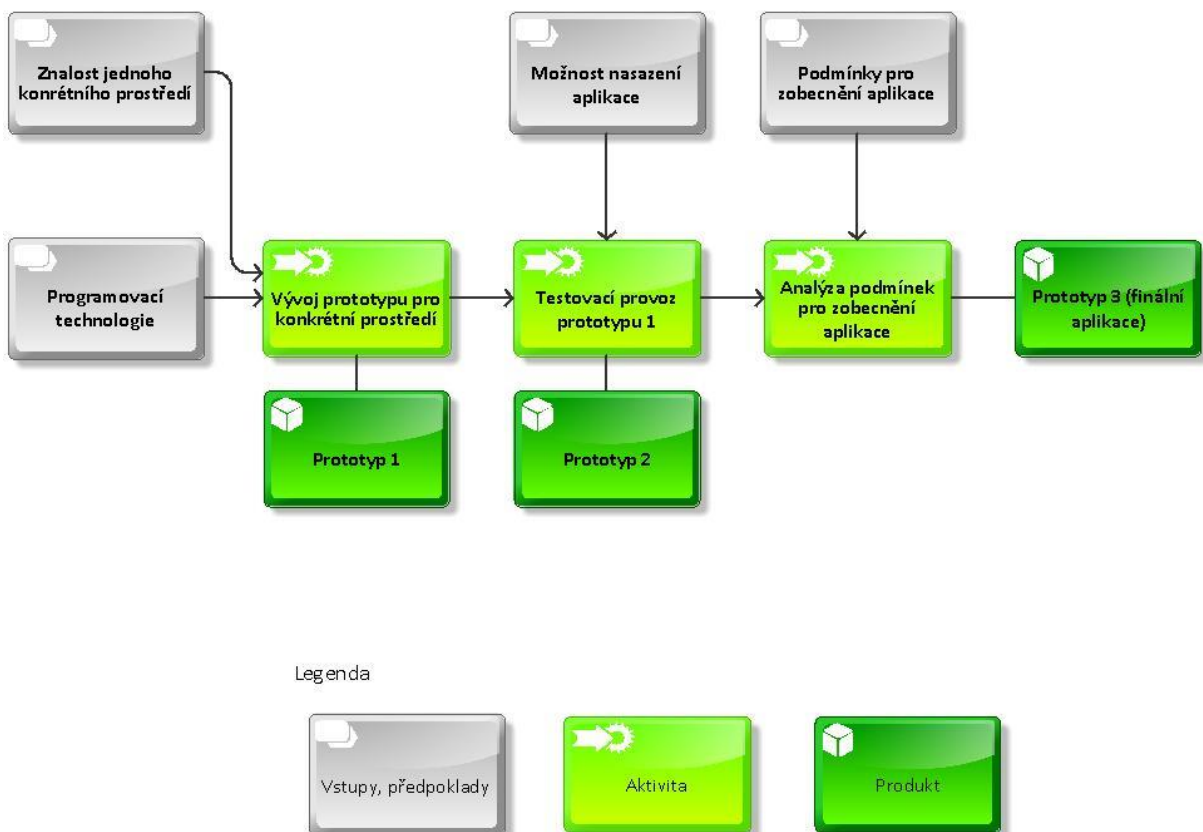
Prototyp 1	Aplikace vytvořená pro konkrétní prostředí
Předpoklady	Znalost konkrétního prostředí
	Znalost požadavků na práci s aplikací

Prototyp 2	Rozšíření aplikace o doplňkové funkce navržené uživateli
Předpoklady	Možnost nasazení aplikace do testovacího provozu
	Definování reálných doplňujících požadavků
Prototyp 3	Zobecnění aplikace pro určité varianty prostředí
Předpoklady	Omezený ale smysluplný počet variant prostředí
	Znalost specifik jednotlivých prostředí

Zdroj: vlastní zpracování

Obrázek 48 ukazuje model Metody Tří prototypů, který byl vytvořen pomocí SW ARIS Express – nástroj Business proces.

Metoda Tří prototypů



Obrázek 48: Model – Metoda Tří prototypů

Zdroj: vlastní zpracování

7 ZÁVĚR

Cílem práce byl vývoj webové aplikace pro dokumentaci sítě SAN v topologii Switched Fabric. Na základě charakteristiky možných konfigurací tohoto typu sítě byla navržena a vytvořena aplikace pro potřeby správce sítě SAN. Využity byly nástroje datového modelování a nástroje pro vývoj webových aplikací.

Aktuální a přesná dokumentace sítě SAN je nutným předpokladem pro výkon administrátorských činností typu správa či monitoring. Rychlá orientace v konfiguraci dané SAN a její přehledné zobrazení bez nutnosti administrátorského přístupu na jednotlivé prvky sítě a vyčítání informací z dílčích konfigurací je žádoucí nejen pro správce takové sítě, ale i pro systémové inženýry, projektanty nových systémů či aplikací apod. Tato práce předkládá aplikaci, která je alternativou k obdobným, avšak bez vysokých investic nedostupným aplikacím.

V úvodu práce byla připomenuta funkce SAN v IS a představeny možné topologie sítě SAN Switched Fabric, které byly zkoumány při vytváření finální verze aplikace. Zmíněny byly principy konfigurace SAN pro objasnění obsahu vyvíjené aplikace. Další zpracování této práce probíhalo podle předem navržených kroků. V prvním kroku byl definován účel aplikace a okruh uživatelů. Dále byl zvolen formát aplikace a technické prostředky použité pro její vývoj. V dalším kroku byl zvolen způsob vývoje aplikace – prototypování – a následoval vlastní vývoj aplikace. Úvodní prototypy byly vytvořeny pro známé konkrétní prostředí, následující pak pro obecné použití na libovolnou topologii SAN Switched Fabric. V posledním kroku byla stručně zhodnocena výsledná aplikace a postup, který byl při vývoji aplikace použit. Použitý postup byl zformulován jako Metoda Tří prototypů.

Finální verze aplikace pro dokumentaci sítě SAN představuje cíl této práce. Aplikace umožňuje dokumentaci sítě SAN Switched Fabric v libovolné topologii. Kompletní zdrojový kód aplikace a vyexportovaná databázová struktura je přílohou této práce v digitální formě na přiloženém CD. Dílčím výsledkem této práce je i formulace postupu, použitého při vývoji této aplikace, nazvaného Metoda Tří prototypů.

Lze konstatovat, že cíl práce byl splněn.

POUŽITÁ LITERATURA A ZDROJE

- [1] WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL*. Přeložil Ondřej BAŠE. Brno: Computer Press, 2017. ISBN 978-80-251-4892-1.
- [2] ŠIMONOVÁ, Stanislava. *Databázové systémy I*. Pardubice: Univerzita Pardubice, 2013. ISBN 978-80-7395-702-5.
- [3] ČEGAN, Lukáš. *Vývoj webových aplikací v PHP a NetBeans*. Pardubice: Univerzita Pardubice, 2015. ISBN 978-80-7395-858-9.
- [4] BORONCZYK, Tim. *MySQL okamžitě*. Přeložil Milan DANĚK. Brno: Computer Press, 2016. ISBN 978-80-251-4737-5.
- [5] TATE, Jon, Pall BECK, Hector Hugo IBARRA, Shanmuganathan KUMARAVEL a Libor MIKLAS. *Introduction to Storage Area Networks*. IBM International Technical Support Organization, 2016. ISBN 0738441430.
- [6] NEGRINO, Tom a Dori SMITH. *JavaScript pro World Wide Web*. Praha: SoftPress, c2001. Praktická vizuální příručka. ISBN 80-86497-09-7.
- [7] MATOUŠEK, Jakub. *Virtualizace dat* [online]. [cit. 2016-10-29]. Dostupné z: <http://www.systemonline.cz/sprava-it/virtualizace-dat.htm>
- [8] PUŽMANOVÁ, Rita. *DAS, SAN, NAS: Varianty řešení ukládání a zálohování dat* [online]. 2004 [cit. 2016-10-12]. Dostupné z: <http://www.systemonline.cz/clanky/das-san-nas.htm>
- [9] BLAHA, Ondřej. *Celková správa sítě SAN* [online]. IBM Tivoli Productivity Storage Center, 2016 [cit. 2017-03-03]. Dostupné z: http://www-05.ibm.com/cz/events/iif2010/resources/Komplexni_sprava_SAN_infrastruktury_-_novy_Tivoli_Storage_Productivity_Center_-_Ondrej_Blah.pdf
- [10] ČERNÝ, Jiří. *NAS vs. SAN - jak na správu dat?* [online]. 2009 [cit. 2016-10-12]. Dostupné z: <http://www.svethardware.cz/nas-vs-san-jak-na-spravu-dat/27556>
- [11] *Cisco MDS 9020 Fabric Switch Configuration Guide and Command Reference: Cisco MDS 9000 FabricWare Release 2.1(3)* [online]. 2006 [cit. 2016-10-16]. Dostupné z:

http://www.cisco.com/en/US/docs/storage/san_switches/mds9000/sw/rel_2_x/fabricware/configuration/guide/cliguide.html

PŘÍLOHY

Příloha A – Struktura přiloženého CD

CD přiložené k této práci používá souborový systém ISO 9660. Obsah adresářové struktury je následující:

- *Application/*
 - *SAN.zip* – komprimovaný archiv zdrojových kódů (skriptů) aplikace včetně podadresáře *images*
- *Database/*
 - *E17542.sql* – vyexportovaná databázová struktura aplikace