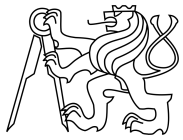
ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ



# ZADÁNÍ BAKALÁŘSKÉ PRÁCE

| | |
|---|---|
| **Název:** | Webová aplikace pro zpracování inventarizačních dat |
| **Student:** | Martin Pavelek |
| **Vedoucí:** | Ing. Viktor Černý |
| **Studijní program:** | Informatika |
| **Studijní obor:** | Web a multimédia |
| **Katedra:** | Katedra softwarového inženýrství |
| **Platnost zadání:** | Do konce letního semestru 2016/17 |

## Pokyny pro vypracování

Proces inventarizace představuje především přesun velmi jednoduchých dat mezi zaměstnanci. V mnoha společnostech je inventura samotná prováděna buď papírově, nebo se data předávají formou tabulkového dokumentu. Za pomoci webových technologií vytvořte informační systém, který dokáže tento proces zautomatizovat a nabídne uživatelsky komfortní rozhraní. Postupujte podle následujících pokynů.

1. Analyzujte proces inventarizace dat v existující společnosti.
2. Identifikujte potřeby osob ve firemní hierarchii, které s inventárními daty pracují.
3. Zaměřte se především na požadavky na uživatelské rozhraní.
4. Na základě předchozí analýzy vyberte vhodnou kombinaci webových technologií pro implementaci uživatelského rozhraní.
5. Pro ukládání tabulky s inventárními daty použijte databázi MySQL.
6. Proveďte alespoň základní uživatelské testování.

## Seznam odborné literatury

Dodá vedoucí práce.

L.S.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
děkan

V Praze dne 10. ledna 2016

České vysoké učení technické v Praze
Fakulta informačních technologií
Katedra Softwarového Inženýrství

Bakalářská práce

# Webová aplikace pro zpracování inventarizačních dat

***Martin Pavelek***

Vedoucí práce: Ing. Viktor Černý

16. května 2016

# Poděkování

Tímto bych chtěl poděkovat vedoucímu této práce Ing. Viktorovi Černému za možnost podnětných konzultací a ochotu při řešení problémů. Dále bych rád poděkoval zaměstancům firmy Vermont Holding a. s. za spolupráci a aktivní zájem. V neposlední řadě bych chtěl poděkovat své rodině a přátelům za podporu během celého studia.

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 16. května 2016 .....

### Odkaz na tuto práci

Pavelek, Martin. *Webová aplikace pro zpracování inventarizačních dat.* Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2016.

# Abstrakt

Tato bakalářská práce se zabývá archivací výsledků inventur prováděných na prodejnách s oděvním zbožím firmy Vermont Holding a. s. Autor se v úvodu věnuje průzkumu postupů na jednotlivých prodejnách i kancelářských odděleních a sjednocení těchto procesů pro jednotlivé země (Česká republika, Slovenská republika, Maďarsko). Po nasazení do provozu bude aplikace využita jako online archiv.

Po důkladné analýze dostupných řešení byl navržen informační systém jako webová aplikace, která nabídne komfortní uživatelské rozhraní.

Výstupem práce je aplikace dostupná přes webové rozhraní v jazyce PHP, která bude sloužit ke vkládání výsledků přímo z prodejen. V závěru se autor věnuje vyhodnocení naplnění cílů a možnostem, jak řešení dále rozšířit.

**Klíčová slova** informační systém, inventura, webová aplikace, PHP, archivace, bezpečnost

# Abstract

This thesis considers archiving the results of stock-takings, carried by stores of clothes selling company Vermont Holding a. s. In the introduction, the author survey procedures at individual stores and office departments and unify these processes for all countries (Czech republic, Slovakia, Hungary). After the application is put into operation, it will be used as an online archive.

After a detail analysis of available solutions, an information system has been designed as a web application, which offers a comfortable user interface.

The output of this work is the application available via the web interface in PHP language, that will be used to insert the stock-taking results directly from stores. In conclusion, the author evaluates the objectives and opportunities for further expansion of the solution.

**Keywords** information system, inventory, web application, PHP, archiving, security

# Obsah

# Seznam obrázků

# Seznam tabulek

# Úvod

Invetura je známý proces, pokud chceme zjistit stav majetku či jiných prostředků. Malým podnikatelům stačí většinou tužka a papír. Po sepsání přichází porovnání soupisu oproti dokladům a účetním záznamům. Z následných odchylek se okamžitě vyvodí důsledky a výsledek se například založí do složky.

Větší firmy s centralizovaným řízením jsou sice schopny jednorázově takovouto akci vykonat, ale nejsou schopny s takto nabytými informaceni dlouhodobě pracovat. Informace zachycené pouze v tištěné formě nebo v textových souborech je velmi těžké uschovávat. S takovými daty přibývají administrativní povinnosti, zdržování práce a hrozí nebezpečí manipulace z nepozornosti (vytracení originálního záznamu, smazání souboru, založení do špatné složky,...).

Tématem této bakalářské práce je zabývat se zefektivněním práce s inventárními daty. Cílem je vytvořit informační systém, který umožní automatizované zpracování inventárních dat. Výsledný informační systém zautomatizuje činnosti, které byly doposud prováděny manuálně a nabídne ucelený přehled o inventárních datech v závislosti na aktuální uživatelské roli.

Kapitola 1

# Cíl práce

Cílem práce je vytvořit uživatelsky komfortní informační systém, který bude sloužit ke zpracování a prezentaci inventárních dat. Systém bude také umožňovat ukládat soubory spojené s procesem inventury. Nedílným požadavkem pro využití takového systému je i příprava serveru, který bude připraven pro bezpečné zavedení do provozu.

Dílčími cíli bude identifikovat datové struktury potřebné pro ukládání dat, zjistit jaké úkony se s daty provádějí a osoby ve firemní hierarchii, které s daty pracují.

Výsledný informační systém bude implementován v jazyce PHP ve spolupráci s relační databází MySQL. Uživatelům bude dostupný přes webové rozhraní pomocí protokolu HTTPS.

Kapitola 2

# Analýza

## 2.1 Současný stav řešení problematiky

Práce je vytvořena pro nasazení ve firmě Vermont Holding a. s., která se zabývá prodejem značkového oblečení Gant, La Martina, Chaps a dalších. Aktuálně společnost provozuje 80 prodejen ve 3 zemích (CZ, SK, HU)[1] a čítá 500 zaměstanců (zaměstnanci prodejen i kanceláří dohromady). Obchodní jednotky jsou umístěny ve velkých obchodních centrech s dobrou dopravní dostupností.

Každá tato prodejna musí provádět inventury veškerého artiklu, a to většinou jednou měsíčně. Inventura zahrnuje fyzickou kontrolu zboží a hotovosti oproti pokladnímu systému a následný zápis výsledků. Výsledky se zapisují do souborů různých typů. Především se využívají soubory typu „.xls“ a „.doc“. Pravidelný měsíční audit provedený společností DANTEM s. r. o. obsahuje soubory „.csv“. Následně všechny soubory obdrží e-mailem Country manager. S výsledky inventur jsou z prodejen zasílány také soubory s přehledem závad, docházky a další.

Každý měsíc tedy obdrží Country manager e-mailem přibližně 5 souborů z každé prodejny ve své zemi a následně tyto přehledy předává na účetní oddělení opět e-mailem. Tím dochází k zahlcení schránek, nechtěným ztrátám dat a v neposlední řadě je téměř nemožné dohledat určitou informaci zpětně.

## 2.2 Možnosti řešení

Sdílení informací a souborů mezi pracovníky na různých pozicích ve firemní hierarchii je v rámci kanceláří řešeno sdílenými síťovými disky (NAS).[2] Bylo by možné prodejny propojit s úložištěm pomocí VPN[3] a sdílet soubory kopírováním mezi složkami. Alternativou by mohly být vlastní či hostované cloudové

[1] Česká republika, Slovenská republika, Maďarská republika

[2] Network Attached Storage – sdílené datové úložiště na síti

[3] Virtual Private Network – prostředek k propojení několika počítačů veřejnou sítí

služby, které by umožnily jednoduché sdílení i přístup odkudkoli, a to přes webové rozhraní či aplikaci.

## 2.3 Zvolené řešení

Ze zmíněných technologií vlastně žádná nepřináší výrazné výhody a není uživatelsky nijak komfortní. Takto řešená úložiště pouze vynechávají e-mail jako archiv. Uživatel ale stále musí být velmi opatrný. Není možné omezit jeho chování, kontrolovat vstup či rozšiřovat funkcionalitu. Z těchto důvodů jsou tato řešení nedostačující.

Vhodným řešením, které splní požadavky (zadefinované v sekci Analýza požadavků) je implementovat vlastní informační systém, který nabídne větší funkcionalitu i větší pohodlí uživatelům.

## 2.4 Zainteresované osoby

Pro přiblížení procesu inventury je potřeba se seznámit s lidmi, kteří s inventurami přicházejí do styku.

Nejvyšší osobou ve firemní hierarchii, která se zajímá o výsledky inventur je Country manager. Tento pracovník je zodpovědný za veškeré obchodní aktivity, budování a rozvoj obchodních kontaktů, firemní růst či vedení týmu.[1] Sleduje chod obchodních jednotek, zajišťuje spolupráci mezi prodejnou a ostatními odděleními.

Jeho podřízeným je Store manager. Jako vedoucí obchodní jednotky je odpovědný za komplexní řízení obchodu (jedné pobočky). Zabývá se komerční, logistickou i personální stránkou fungování obchodu. Zodpovídá také za ekonomické výsledky.[2]

Znát výsledky inventur potřebuje také účetní. Není podřízeným Country managera, ale úzce spolupracují. Účetní zpracovává podklady a zajišťuje vyrovnání finančních závazků. Jeho povinností je i postihování nebo odměňování zaměstnanců podle měsíčních obratů a inventur.

Každý systém potřebuje administrátora, osobu která bude plnit funkci supervizora (dohlížet na správné používání systému a spolehlivý chod). Tento správce systému nepotřebuje znát žádné z předávaných informací, ale musí vědět, jak aplikace funguje. Jeho úlohou je napravit problémy, které mohou uživatelům nastat a spravovat nastavení systému.

Dalšími jsou administrativní pracovníci, kteří vytvářejí přehledy pro vedení, srovnávají stavy skladů s pokladním systémem či objednávají nové zboží.

## 2.5 Popis procesu inventury

V procesu inventury jsou drobné nuance závislé na požadavcích Country managera dané země. Hlavní základ je ale stejný, liší se pouze odevzdávané sou-

bory nebo přesný postup inventury podle vnitřních předpisů.

Jednou za měsíc má každá prodejna povinnost nahlásit tržby a výsledky inventur. Tržba je uvedená v pokladním systému. Hodnoty z pokladního systému zaznamená do souboru „.pdf“[4] a odesílá obchodnímu centru. Inventuru hotovosti v pokladně si provádí zaměstnanci sami a za správnost údajů ručí svým podpisem. Inventuru artiklu na prodejní jednotce provádí společnost DANTEM s. r. o. skenováním EAN kódů ze štítků zboží a zaměstnanci pouze asistují při kontrole. Výstupem inventury provedené společností DANTEM s. r. o. (dále jen „Dantem inventura“) jsou 3 soubory typu „.csv“[5]. Soubory mají vždy stejný formát:

„NÁZEVPRODEJNY_KÓDPRODEJNY_YYYY-MM-DD_XXX.csv“

Místo „XXX“ jsou typy souboru. Prvním je „all“, který obsahuje všechny naskenované kódy a počet takovýchto naskenovaných artiklů. Druhým je „unknown“ a obsahuje naskenované kódy, které nebyly rozpoznány. To znamená, že daný kód nemá žádné přiřazené zboží a nebylo rozpoznáno. Nejdůležitějším je třetí soubor „differences“, který udává odchylky oproti očekávanému stavu, který byl nahlášen prodejnou. Tyto tři soubory jsou odeslány prodejně.

Prodejna je povinna si vést evidenci docházky, přesčasů a poruch. Tyto informace si udržuje samostatně.

Country manager české republiky vyžaduje odesílání všech výše zmíněných souborů každý měsíc (tržby, hotovost, tři soubory Dantem inventury, docházka, přesčasy, poruchy). Country manager slovenské a maďarské republiky si nechává zasílat pouze soubory, které vyžadují pozornost. Je tedy vždy na zvážení Store managera, které soubory a informace budou předány. Nepsaným pravidlem je, že se z prodejen posílají výsledky Dantem inventury, pokud obsahuje rozdíly a další soubory pouze v případě potřeby řešit nějakou nenadálou situaci.

V České republice jsou soubory posílány Country managerovi a ten sjednává nápravu dalším rozesíláním souborů kolegům, například na účetní oddělení. Na Slovensku a Maďarsku komunikují buď prodejny přímo s dalšími zaměstnanci (účetní, skladník,...) nebo Country manager kontaktuje kolegy sám na základě odevzdaných souborů.

Všechny zmíněné soubory jsou uloženy pouze v e-mailových schránkách a dochází ke ztrátám (nechtěné smazání, označení za spam), zahlcení schránek a je náročné zpětně dohledat nějakou informaci či ji předat.

## 2.6 Analýza požadavků

Tato sekce definuje funkční i nefunkční požadavky pro daný systém. Specifikace požadavků umožňuje popsat očekávanou funkčnost, kterou má systém

[4] Portable Document Format – univerzální formát pro přenos dokumentů

[5] Comma Separated Values – hodnoty oddělené čárkami pro výměnu tabulkových dat

splňovat, vyjasní zadání a zachytí omezení kladená na výsledné řešení. Požadavky vyplynuly z rozhovorů s uživateli na všech zmíněných pozicích.

### 2.6.1 Funkční požadavky

Výsledný informační systém by měl splňovat požadavky cílové skupiny.

#### F1 – Kontrola přístupu

Systém bude umožňovat přihlašování a odhlašování ze systému. Každý uživatel má své vlastní přístupové údaje. Systém bude dostupný pouze přihlášeným uživatelům.

#### F2 – Zadání měsíčního reportu

Systém bude uživateli na prodejně umožňovat přidání nového reportu. Bude moci zadat měsíční obrat a výsledek inventury hotovosti.

#### F3 – Ukládání souborů

Systém umožní uživatelům ukládat soubory vztahující se k danému reportu. Tyto soubory budou dostupné pro uživatele, kteří mají právo si daný report zobrazit. Systém také zobrazí výsledky z nahraného souboru rozdílů (differences.csv).

#### F4 – Zobrazení archivu výsledků

Systém bude uživateli zobrazovat archiv s výsledky inventur, na které se může podívat. Archiv bude tříděný podle země, období reportu a prodejny.

#### F5 – Administrátorské rozhraní

Systém umožní administrátorovi provádět nejběžnější akce přímo z webového rozhraní systému. Jedná se o operace: vytvoření nového uživatele, změna hesla uživateli, přiřazení práv uživateli.

#### F6 – Vzdálená správa serveru

Server, na kterém bude systém umístěn, bude umožňovat vzdálenou správu pomocí zabezpečené komunikace.

### 2.6.2 Nefunkční požadavky

#### N1 – Zabezpečený server

Server, na kterém bude systém umístěn, bude zabezpečen proti neoprávněným zásahům a útokům.

**N2 – Komunikace pomocí protokolu HTTPS**

Komunikace klientů přes webové rozhraní bude probíhat přes bezpečné spojení protokolem HTTPS.

**N3 – Aplikace dostupná přes webové rozhraní**

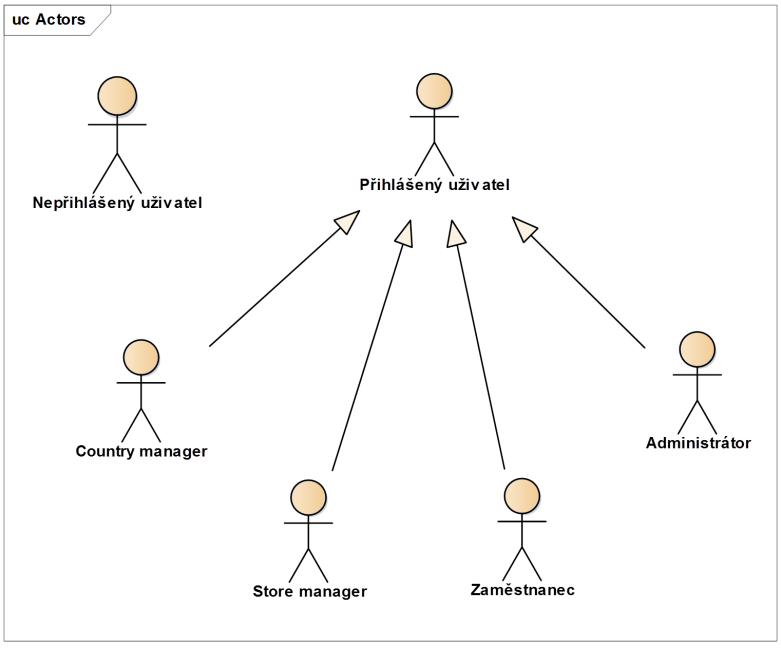
Výsledná aplikace bude dostupná uživatelům přes webové rozhraní pomocí webového prohlížeče.

## 2.7 Modelování případů užití

V této části jsou definovány případy užití (neboli use case), které mohou vykonávat uživatelé systému. Případy užití pokrývají funkční požadavky.

### 2.7.1 Seznam účastníků

Seznam účastníků vyplývá z definice osob účastnících se procesu inventarizace. Nepřihlášený uživatel se může pouze přihlásit do systému nebo si zobrazit kontakty. Přihlášený uživatel může vykonávat operace v systému, pokud mu to dovolí jeho uživatelská oprávnění (má-li k nim přiřazeny práva administrátorem).



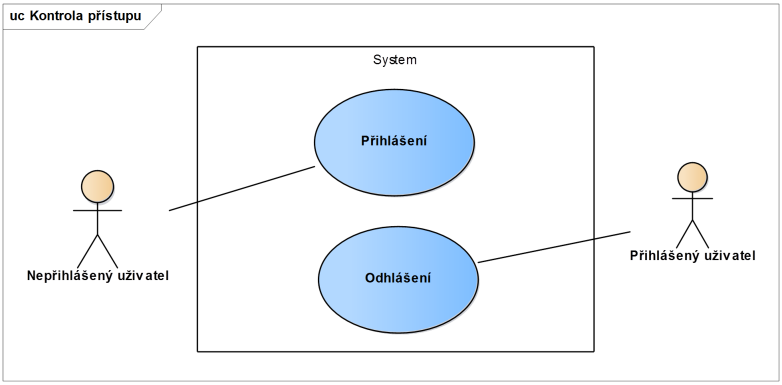

Obrázek 2.1: Seznam účastníků

### 2.7.2 Případy užití

Jednotlivé scénáře užití definují vztahy mezi aktéry a případy užití.

#### 2.7.2.1 Kontrola přístupu

Tento diagram plně pokrývá funkční požadavek F1 a definuje kontrolu přístupu uživatelů do aplikace.




Obrázek 2.2: Use case – Kontrola přístupu

**Přihlášení** Tento případ nastává vždy při startu práce se systémem. Uživatel, který stránku navštíví je nepřihlášeným uživatelem.

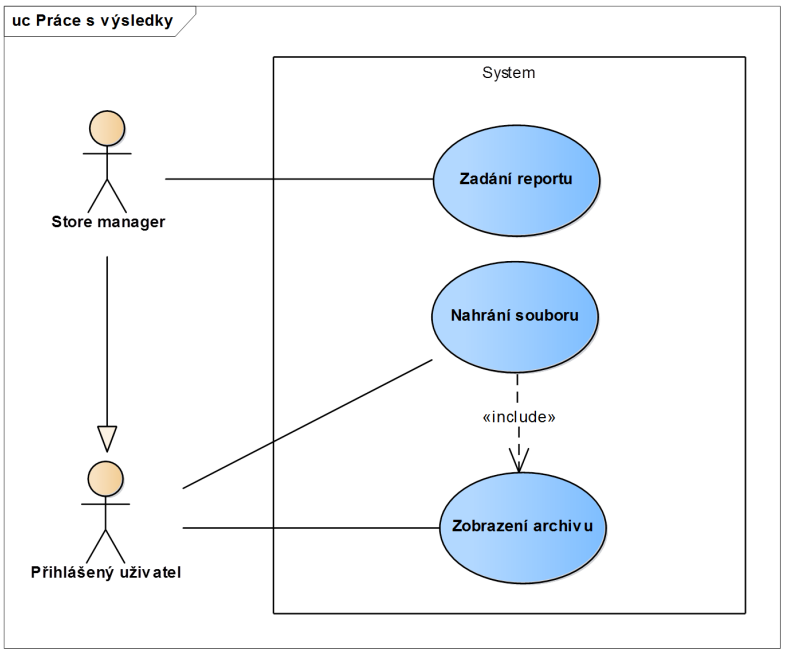
1. Systém zobrazí formulář pro zadání přístupového jména a hesla.
2. Uživatel vyplní údaje.
3. Systém ověří zadané údaje.
4. Uživatel se stane přihlášeným.

**Odhlášení** Tento případ užití začíná, pokud se uživatel rozhodne ukončit práci se systémem.

1. Uživatel zvolí volbu odhlásit v menu aplikace.
2. Systém odhlásí uživatele a zobrazí přihlašovací formulář.

#### 2.7.2.2 Práce s výsledky

Tento scénář pokrývá funkční požadavky F2 až F4 a definuje možné úkony prováděné uživatelem s výsledky inventur v systému.




Obrázek 2.3: Use case – Práce s výsledky

**Zadání reportu** Tento případ užití začíná ve chvíli, kdy Store manager potřebuje zadat měsíční hlášení o tržbě a stavu pokladny. Tento případ užití pokrývá funkční požadavek F2.

1. Uživatel zvolí v menu aplikace přidání nového reportu.

2. Systém zkontroluje oprávnění uživatele a zobrazí formulář pro vyplnění data a částek.

3. Uživatel vyplní formulář a odešle jej do systému.

4. Systém informace uloží a zobrazí uložený výsledek uživateli.

**Zobrazení archivu** Tento případ užití začíná, pokud si uživatel žádá vidět seznam uložených výsledků. Případ užití pokrývá funčkní požadavek F4.

1. Uživatel v menu aplikace zvolí položku archiv.

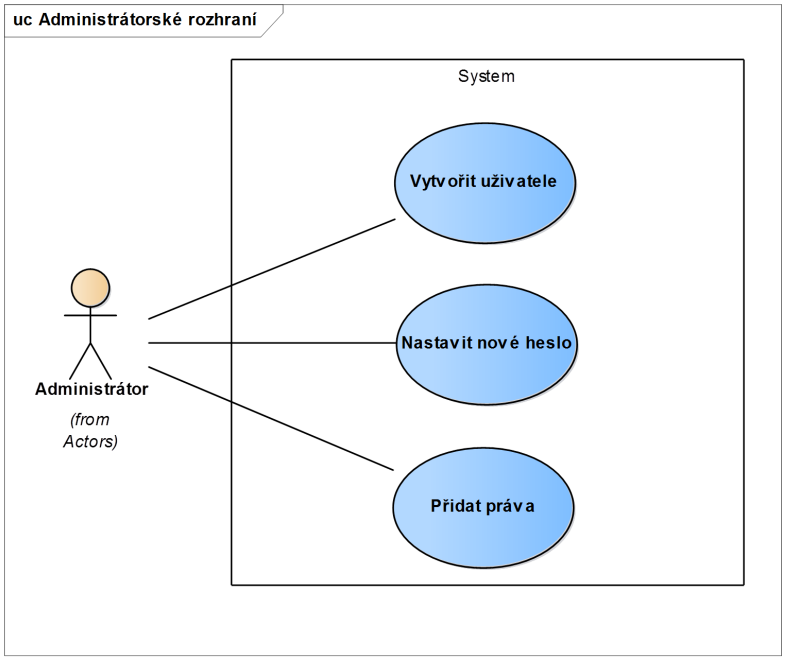
2. Systém zkontroluje práva uživatele a zobrazí výsledky, které má uživatel právo prohlížet. Výsledky se zobrazují ve stromové struktuře, řazeny podle země, roku, měsíce a prodejny.

**Nahrání souboru** Tento případ užití začíná ve chvíli, kdy uživatel chce nahrát soubor vztahující se k danému reportu. Případ užití pokrývá funkční požadavek F3.

1. Include (Zobrazení archivu)
2. Uživatel zvolí příslušné hlášení.
3. Systém zkontroluje práva uživatele a zobrazí informace reportu, ke kterým má uživatel oprávnění.
4. Uživatel zvolí možnost přidání souboru.
5. Systém zobrazí nahrávací formulář.
6. Uživatel zvolí soubor, který chce nahrát a odešle formulář.
7. Systém uloží soubor k reportu.

#### 2.7.2.3 Administrátorské rozhraní

Tento scénář pokrývá funkční požadavek F5 a definuje možné úkony prováděné administrátorem v systému.




Obrázek 2.4: Use case – Administrátorské rozhraní

**Vytvoření uživatele** Tento případ užití začíná ve chvíli, kdy je potřeba vytvořit nového uživatele systému. Například jde o nového zaměstnance.

1. Uživatel s administrátorskými právy zvolí v menu aplikace administrátorskou sekci.
2. Systém ověří práva uživatele a nabídne mu dostupné akce.
3. Administrátor zvolí možnost vytvoření nového uživatele.
4. Systém zobrazí formulář.
5. Administrátor vyplní požadované informace o nově vytvářeném uživateli a odešle formulář.
6. Systém ověří data a uloží výsledek. Nově vytvořený uživatel se může přihlásit do systému.

**Nastavení nového hesla** Tento případ užití začíná ve chvíli, kdy je potřeba upravit heslo existujícímu uživateli systému. Nejčastěji jde o případ zapomenutého hesla.

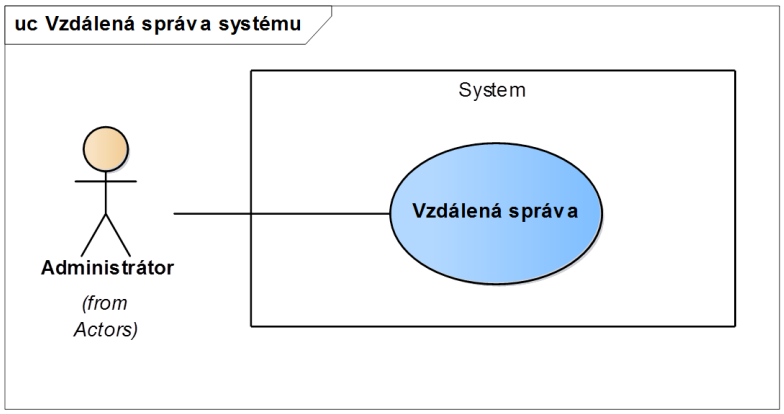
1. Administrátor zvolí v menu aplikace administrátorskou sekci.
2. Systém zkontroluje práva uživatele a nabídne mu dostupné akce.
3. Administrátor zvolí nastavení hesla existujícímu uživateli.
4. Systém zobrazí formulář.
5. Uživatel vyplní nové heslo a odešle formulář.
6. Systém ověří data a uloží výsledek. Uživatel se může přihlásit do systému pomocí nového hesla.

**Přidání práva** Tento případ užití začíná ve chvíli, kdy je potřeba přidat práva uživateli systému. Jedná se o přidání práv novému uživateli nebo rozšíření práv stávajícímu uživateli.

1. Administrátor zvolí v menu aplikace administrátorskou sekci.
2. Systém zkontroluje práva uživatele a nabídne mu dostupné akce.
3. Administrátor zvolí přidání práv uživateli.
4. Systém zobrazí formulář.
5. Administrátor přidělí práva vybrané osobě.
6. Systém ověří data a uloží výsledek.

### 2.7.3 Vzdálená správa

Tento scénář pokrývá funkční požadavek F6. Případ užití začíná, pokud je potřeba systém vzdáleně spravovat mimo webové rozhraní systému.




Obrázek 2.5: Use case – Vzdálená správa

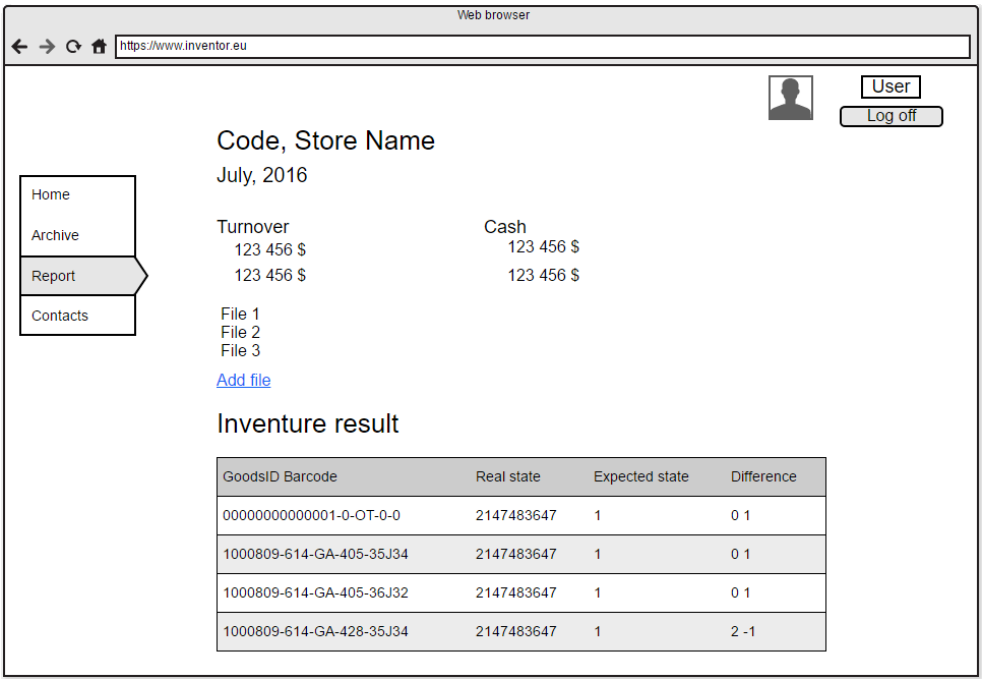
**Vzdálená správa**

1. Uživatel se pomocí SSH tunelu připojí k serveru.
2. Systém zkontroluje práva uživatele a povolí mu vzdálené ovládání.
3. Uživatel využije tunelu a připojí se k VNC serveru.
4. Systém zkontroluje korektnost přístupu.
5. Uživatel provede potřebné úpravy pomocí grafického rozhraní.

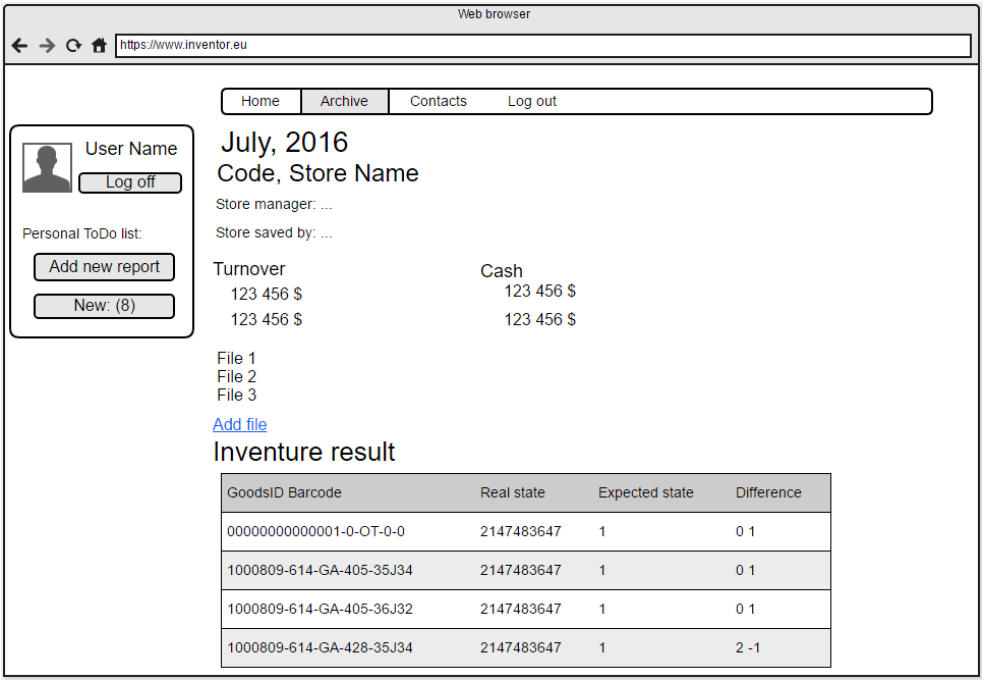
## 2.8 Uživatelské rozhraní

Cílem dobrého uživatelského rozhraní je především spokojenost uživatele. Součástí rozhovorů během analýzy byl také návrh rozhraní systému. Budoucím uživatelům byly předvedeny dvě možnosti vzhledu stránek. Hlavní rozdíl byl v rozložení ovládacích prvků a množství zobrazených informací.

První návrh představuje typické rozmístění prvků na stránce. Navigační panel je umístěn vlevo a karta uživatele v pravém horním rohu. Druhý návrh obsahuje méně obvyklé rozmístění, kdy navigační panel je umístěn na středu a informace o uživateli vlevo. Navíc obsahuje list osobních úkolů a detaily o prodejně.




Obrázek 2.6: Ukázka navrhovaného uživatelského rozhraní 1




Obrázek 2.7: Ukázka navrhovaného uživatelského rozhraní 2

Kapitola 3

# Technologie

Analýza přinesla velmi podrobnou představu, co by měl systém splňovat. Pro realizaci práce je potřeba si vybrat správné technologie pro vytvoření a funkci systému. Následně je nutné implementovat systém a připravit server do požadovaného stavu.

## 3.1 Návrh architektury

Cílem výběru správné architektury při tvorbě systému je především jednodušší tvorba a oddělení logiky od výstupu. Řeší tedy problém, kdy se v jednom souboru (třídě) nacházejí logické operace a zároveň renderování výstupu. Soubor tedy obsahuje databázové dotazy, logiku a HTML tagy. Kód se samozřejmě špatně udržuje, natož rozšiřuje a ztrácíme se v něm.

### 3.1.1 MVC

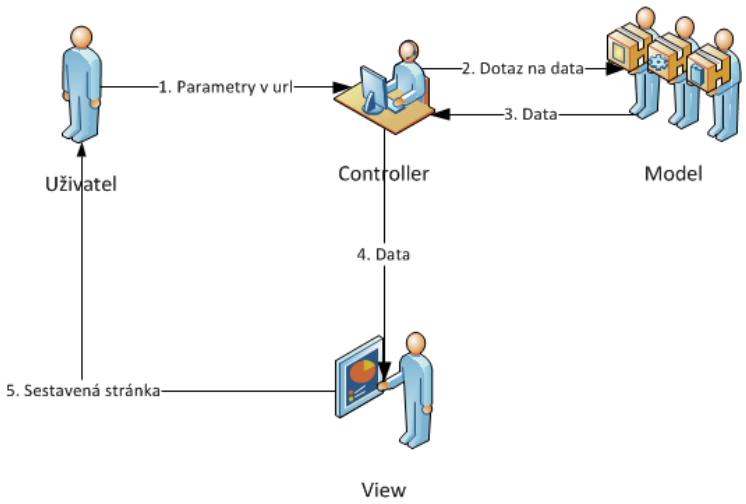
MVC je velmi oblíbený architektonický vzor, který se na webu velmi rozšířil a je typický pro jakýkoliv větší web. Celá aplikace je rozdělena na komponenty tří typů, hovoříme o Modelech, View (pohledech) a Controllerech (kontrolerech), od toho MVC.

**Model** Model obsahuje veškerou logiku jako jsou výpočty, databázové dotazy, validace vstupů a podobně. Jeho funkce spočívá v přijetí parametrů a vydání dat. Model neví, jaký je zdroj přijatých parametrů a ani jak budou výstupní data zformátována a vypsána.

**View** View se stará o zobrazení výstupu uživateli. Jedná se o PHP šablonu, obsahující HTML stránku a tagy skriptovacího jazyka (PHP), který umožňuje do šablony vkládat proměnné, případně provádět iterace (cykly) a podmínky. Pohledů (View) máme mnoho. Do View jsou posílána různá data, vždy podle

toho, jakými daty chceme dynamická políčka naplnit. Tato data jsou poté dosazena do HTML elementů šablony. Obsahuje tedy minimální množství logiky, která je pro výpis nutná. View podobně jako Model vůbec neví, jaký je zdroj přijatých dat, stará se jen o jejich zobrazení uživateli.

**Controller** Controller plní funkci prostředníka, se kterým komunikuje uživatel, Model i View. Drží tedy celý systém pohromadě a komponenty propojuje. On jediný ví, o jaká data žádat a kam je posílat, pokud je to potřeba.




Obrázek 3.1: Diagram propojení MVC

### Proces při zobrazení stránky

1. Životní cyklus zahajuje uživatel, který zadá do prohlížeče adresu webu. URL adresa obsahuje i parametry žádané stránky, kterou si přeje zobrazit. Budeme chtít zobrazit detail uživatele s id 15. Taková stránka by mohla být umístěna na URL: http://www.domena.cz/uzivatel/detail/15 Požadavek je nejprve zachycen tzv. routerem. Ten podle parametrů pozná, který controller voláme. Většinou se uvádí jako 1. parametr. Zde je tedy zavolán controller „uživatel“, kterému jsou předány parametry „detail“ a „15“.

2. Daný controller podle parametrů pozná, jaká operace je od něj vyžadována, tedy že má zobrazit detail uživatele. Zavolá Model, který uživatele vyhledá v databázi.

3. Volaný Model vrátí údaje uživatele nalezené v databázi. Dále zavolá další metodu Modelu, která např. vypočítá věk uživatele. Tyto údaje si controller ukládá do proměnných.

4. V posledním kroku odešle controller výsledná data pro vykreslení View. Název pohledu poznáme podle akce, kterou provádíme. Controller tedy poslechl uživatele, obstaral podle parametrů dotazu data od Modelu a předal je View.

5. View přijme data od controlleru a vloží je do připravené šablony. Vytvořená stránka je zobrazena uživateli. Uživatel taková chování uvnitř aplikace vůbec nepozná.[3]

## 3.2 Použité technologie

### 3.2.1 PHP

*„PHP je nejrozšířenějším skriptovacím jazykem pro web, v současnosti (květen 2016) s podílem 82,2 %.“*[4] Oblíbeným se stal především díky jednoduchosti použití a bohaté zásobě funkcí. V kombinaci s operačním systémem Linux, databázovým systémem (obvykle MySQL nebo PostgreSQL) a webovým serverem Apache je často využíván k tvorbě webových aplikací. Pro tuto kombinaci se využívá zkratka LAMP – tedy spojení Linux, Apache, MySQL a PHP, Perl nebo Python[5] (analogicky pro systém Windows jde o WAMP, pro Mac jde o MAMP).

V této práci autor zvolil jazyk PHP také kvůli perfektní dokumentaci.

### 3.2.2 JavaScript

JavaScript je skriptovací jazyk, který umožňuje provádět operace v aplikaci na straně klienta. Umožňuje například verifikaci dat ještě před odesláním na server.

### 3.2.3 HTML a CSS

Při psaní webu se HTML a CSS nedá vyhnout. Standardy používání se sice velmi rychle mění, ale principy jsou stálé. Novější verze zajistí lepší chod v nejnovějších webových prohlížečích.

**HTML** HyperText Markup Language, je značkovací jazyk využívaný pro tvorbu webových stránek s hypertextovými odkazy a umožňuje publikaci dokumentů na internetu.[6]

**CSS** Cascading Style Sheets je jazyk pro popis vizualizace HTML dokumentů.

**LESS** Leaner CSS je knihovna, která do jazyka CSS přidává proměnné, výrazy a makra či definice. O přetvoření zápisu s makry na čisté CSS se stará preprocesor[6] LESS.[7]

```
@nice-blue: #5B83AD;
@light-blue: @nice-blue + #111;
#header { color: @light-blue; }
```

Obrázek 3.2: Ukázka CSS kódu s využitím LESS knihovny

### 3.2.4 Bootstrap

Bootstrap je volně stažitelný framework, který obsahuje sadu souborů typu CSS a JavaScript pro snadnější tvorbu webových aplikací. Právě tento framework využívá LESS pro efektivnější práci se vzhledem. Obsahuje velké množství připravených prvků, jako ikonky, formuláře či navigační panely.

## 3.3 Použité serverové technologie

Každá webová aplikace potřebuje i server, na kterém bude provozována. Pro vývoj systému a testování byl využit vlastní osobní počítač s operačním systémem Microsoft Windows 10 Pro N. Server byl po zprovoznění testován pomocí veřejné IP adresy.

Všechny použité programy a technologie (vyjma freeSSHd) jsou dostupné pro všechny operační systémy a jejich zprovoznění je analogicky totožné. V práci jsou popsány postupy pro operační systém Microsoft Windows.

#### 3.3.0.1 AMPPS

Z volně dostupných WAMP serverů vybral autor balík AMPPS. Největší výhodou AMPPS je uživatelsky přívětivé prostředí a také je tento balík dostupný pro všechny platformy. Instalací AMPPS se nainstalují nejdůležitější komponenty pro webové servery a není tedy nutné instalovat každý komponent samostatně.

Balík AMPPS obsahuje:

- Apache Web Server – nejrozšířenější web server na světě[8]
- MySQL – databázový server
- PHP – skriptovací jazyk, ve kterém je napsána celá aplikace

[6] Program, který zpracovává vstupní data, aby výstup mohl dále zpracovávat jiný program

- Perl, Python – skriptovací jazyky

- Grafické uživatelské rozhraní schopné nastavit nejběžnější úkony

#### 3.3.0.2 MariaDB

MariaDB je stejně jako MySQL relační databáze a je i její přímý nástupce. Její rozhraní je totožné s MySQL. Obsahuje i některé funkce, které v MySQL chybí (úložné enginy, export do JSON[7]) a je výkonější než MySQL.[10]

Důvodem vzniku samostatého projektu MariaDB je odkoupení MySQL technologie společností Oracle.[9] Samotný zakladatel projektu MySQL nepřešel pod společnost Oracle a dnes vede projekt MariaDB. To umožňilo udržet licenci svobodného softwaru GNU GPL. Databázi MariaDB využívají i obrovské světové společnosti jako Facebook, Twitter či Google.[11]

Ze všech těchto důvodů se autor rozhodl vyměnit MySQL připravenou v balíku AMPPS za databázi MariaDB.

#### 3.3.0.3 PHPMyAdmin

PHPMyAdmin je užitečný nástroj pro správu databáze pomocí webového rozhraní. Umožňuje provést všechny možné úlohy nad databázovým serverem jako úpravy databází, tabulek, přístupů uživatelů, provádět export dat či vytvořit databázi podle návrhu. Při práci s daty je možné použít pole pro ruční zadání SQL příkazů. Případně je k dispozici i konzole pro plně ruční komunikaci s databázovým serverem.

#### 3.3.0.4 HTTPS

Hypertext Transfer Protocol Secure je rozšíření komunikačního síťového protokolu HTTP o šifrování komunikace pomocí SSL nebo TLS[8].

Protokol HTTPS využívá asymetrické šifrování. Obě strany si před zahájením komunikace vygenerují pár klíčů (privátní a veřejný). Při zahájení komunikace si vymění veřejné klíče, kde se využívá princip přenosu důvěry, kdy nám protistrana předá veřejný klíč, který je digitálně podepsaný (certifikační autoritou, které důvěřujeme a jejíž veřejný klíč máme v důvěryhodném úložišti). Digitální certifikáty jsou základním kamenem zabezpečení poskytovaného protokoly SSL/TLS.[12]

[7] JavaScript Object Notation – způsob zápisu dat nezávislý na počítačové platformě

[8] Protokol, resp. vrstva vložená mezi vrstvu transportní a aplikační, která poskytuje zabezpečení komunikace šifrováním. TLS je následníkem SSL.

## 3.4 Nástroje pro vzdálenou správu

Nezávisle na implementaci systému bylo nutné připravit prostředky pro vzdálenou správu serveru. Aby byla udržena vysoká bezpečnost provozovaného systému, autor zvolil kombinaci SSH a VNC.

#### 3.4.0.5 SSH

Secure Shell zajišťuje bezpečnou (šifrovanou) komunikaci mezi dvěma počítači. SSH je použito, pokud je potřeba ovládat počítač na dálku nebo jen přenášet soubory (pomocí protokolu SCP).[13]

Pro využití SSH na platformě Windows byl použit program freeSSHd.

#### 3.4.0.6 VNC

Virtual Network Computing je program umožňující vzdálené ovládání počítače přes síť pomocí grafického rozhraní, které využívá běžně uživatel sedící u počítače k jakékoli práci. Protokol není nijak šifrován. Řešením je VNC server s přídavným šifrováním nebo komunikace přes zabezpečený kanál.[14]

V tomto případě je tedy komunikace VNC vedena přes tunel vytvořený pomocí SSH (více v části 4.4 Bezpečnost).

| Technologie | Verze |
|---|---|
| PHP | 5.6.17 |
| JavaScript | ECMAScript 6 |
| HTML | 5 |
| CSS | 3 |
| Bootstrap | 3.3.6 |
| AMPPS | 3.4 |
| Apache | 2.4.18 |
| MariaDB | 10.1.13 |
| PHPMyAdmin | 4.5.1 |
| freeSSHd | 1.3.1 |
| VNC | 5.3.1 |

Tabulka 3.1: Verze všech použitých technologií a nástrojů

Kapitola 4

# Realizace

Na zakládě předchozích analýz byla provedena implementace samotného informačního systému. Tato kapitola popisuje klíčové části systému a upřesňuje postupy při sestavování serveru.

## 4.1 Struktura aplikace

Aplikace je umístěna v: „InstalačníAdresář/www/doména“ (zkráceno na % Main folder %). Zde je popis nejdůležitějších částí (podrobnější struktura je v příloze B).

*% Main folder %*

- `application` ..... Zdrojové kódy aplikace
  - `controllers` ..... Soubory zajišťující řízení požadavků
  - `core` ..... Důležité samostatné komponenty
  - `css` ..... Soubory vzhledu aplikace
  - `fonts` ..... Fonty a ikony
  - `images` ..... Obrázky v aplikaci
  - `js` ..... JavaScriptové soubory
  - `libs` ..... Použité knihovny (Bootstrap)
  - `models` ..... Logika aplikace
  - `upload` ..... Místo pro nahrané soubory
  - `views` ..... Zobrazení výstupů
- `.htaccess` ..... Konfigurační soubor serveru
- `index.php` ..... Počáteční soubor aplikace

## 4.2 Implementace

Splnění požadavků bylo deklarováno v případech užití aplikace. V této části se autor zaměřuje na nejzajímavější části v implementaci těchto požadavků.

Další detaily lze nalézt v částech 4.4 Bezpečnost a 4.5 Nasazení.

### 4.2.1 Kontrola přístupu

Pomyslnou startovní čarou je vždy soubor „route.php“, který před nasměrováním na požadovanou stránkou zkontroluje stav uživatele (výjimkou je stránka s kontakty). Poté je mu umožněno pracovat dále, nebo je odkázán na přihlašovací stránku.

```
if (($controller_name != "login") && ($controller_name != "contacts")) {
        require_once "application/core/login_check.php";
        if (session_status() == PHP_SESSION_NONE) {
                sec_session_start();
        }
        if (login_check() == false) {
                header("Location: /login");
                return;
        }
}
```

Obrázek 4.1: Ukázka kódu omezujícího přístup nepřihlášenému uživateli

Přihlašování je řešeno formulářem na úvodní stránce a kontrolou údajů oproti záznamu v databázi (více v sekci 4.4 Bezpečnost). Odhlášení odstraní session[9] na serveru a zruší všechny nastavené proměnné sezení.

### 4.2.2 Zadání reportu

Systém umožňuje uživateli uvedenému jako vedoucí prodejny (Store manager) přidat měsíční report. Před zobrazením formuláře v souboru „new_report_view.php“ jsou zkontrolována práva uživatele. Poté je možné vyplnit hodnoty obratů a hotovostí.

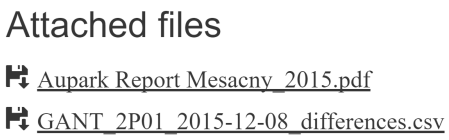
```
if ($_SESSION['user_id'] != $data['chief']){
        echo "User ".$_SESSION['user_id']." is not able to add report";
        echo "to store ".$data['name'].' ('.$data['id'].")";
        echo '<a href="/archive">Go back</a>';
        exit();
}
```

Obrázek 4.2: Ukázka kódu omezujícího zadávání výsledků neoprávněnému uživateli

[9]Relace, neboli sezení představuje permanentní síťové spojení mezi klientem a serverem

### 4.2.3 Ukládání souborů

Na stránce zobrazující report prodejny se nachází tlačítko pro následné uložení souboru a jeho přiložení k reportu. Po nahrání lze stáhnout přiložený soubor kliknutím na název s ikonou diskety. V případě, že jde o soubor s rozdíly ve zboží, připravený firmou Dantem, zobrazí se tento přehled v reportu.




Obrázek 4.3: Soubory přiložené k měsíčnímu reportu

Differences found by Dantem are saved

| GoodsID | Barcode | Real state | Expected state | Difference |
|---|---|---|---|---|
| 10538569-315-GW-G65-37 | 2147483647 | 1 | 0 | 1 |
| 11641910-615-GA-G46-44 | 2147483647 | 1 | 0 | 1 |
| 11641910-615-GA-G46-45 | 2147483647 | 0 | 1 | -1 |
| 11643700-615-GA-G00-46 | 2147483647 | 1 | 0 | 1 |

Obrázek 4.4: Přehled výsledku Dantem inventury

### 4.2.4 Zobrazení archivu výsledků

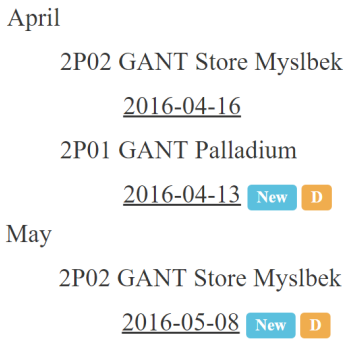
Každý uživatel nalezne v části archiv přehledný seznam hlášení prodejen, na která má právo se podívat. To je zajištěno kontrolou práv oproti zařazení výsledků (země, jednotlivé prodejny). Výsledný výpis také značí, zda už výsledek uživatel označil jako shlédnutý a zda je již u reportu přiložen výsledek Dantem inventury.

### 4.2.5 Administrátorské rozhraní

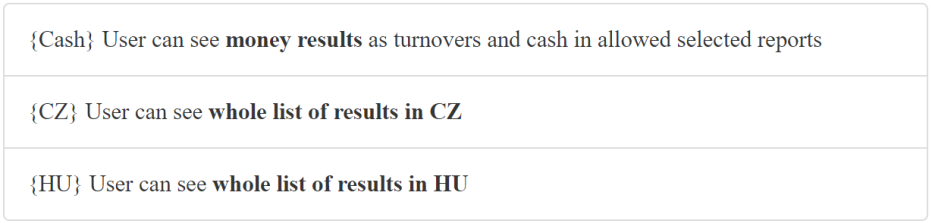
Uživatel s administrátorskými právy má možnost vytvářet nové uživatele a následně jim přidávat práva. Může také znovu nastavit heslo. Administrátoři mají i možnost si prohlédnout práva přidělená uživateli s vysvětlením, co umožňují.

### 4.2.6 Vzdálená správa

Vzdálená správa je zajištěna spojením pomocí SSH. K další spolupráci lze využít VNC. Společně představují účinný nástroj k bezpečnému vzdálenému ovlá-




Obrázek 4.5: Ukázka archivu




Obrázek 4.6: Přehled práv uživatele

dání serveru. Po vytvoření tunelu lze využívat i nezabezpečené spojení VNC, jelikož komunikace probíhá přes šifrovanou komunikaci tunelu. Samotný server je nastaven, aby přijal pouze lokální spojení. Jakýkoli jiný pokus o spojení je odmítnut bez možnosti zadat přihlašovací údaje.

## 4.3 Databázový model

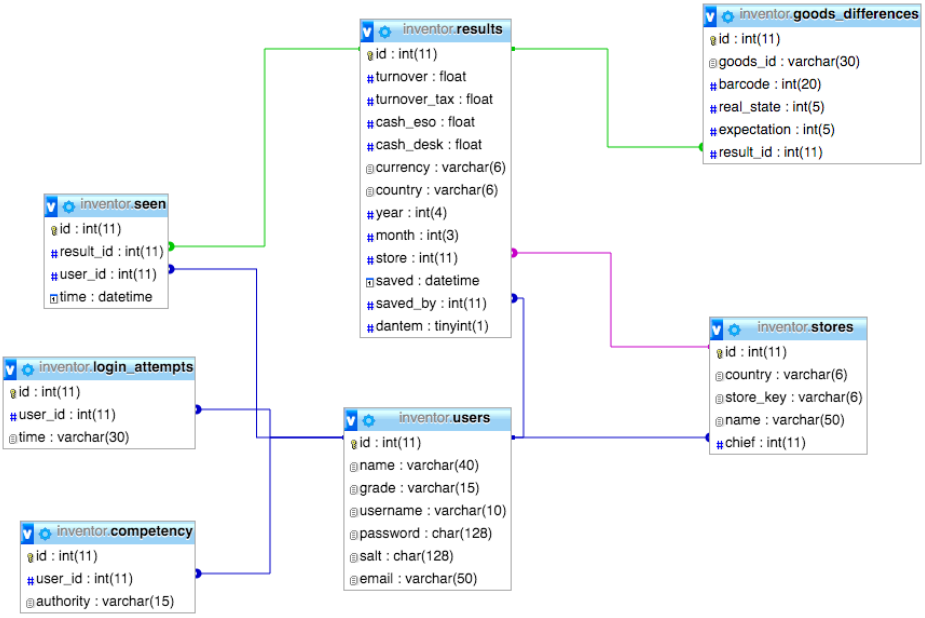
Výsledný model databáze byl vytvořen, aby splňoval všechny požadavky na aplikaci. Popis jednotlivých tabulek:

### Users

Ukládá nezbytné informace o uživateli. Jsou to jméno, pracovní zařazení, přihlašovací jméno, heslo (otisk + sůl) a kontaktní e-mail.

### Results

Ukládá výsledné měsíční hlášení. K tomu je potřeba datové zařazení (rok, měsíc), firemní zařazení (země, obchodní jednotka a tvůrce reportu) a samotný výsledek (obrat, zdaněný obrat, peněžní inventura, porovnání s pokladním systémem, měna). Dále je uložen čas vložení a informace, zda byla provedena Dantem inventura.




Obrázek 4.7: Schéma databáze

### Goods_differences

V případě nahrání výsledku Dantem inventury jsou rozdíly ve zboží uloženy zde. Je zaznamenáno ID zboží, jeho kód, odchylky (reálný stav, očekávaný stav) a výsledek, ke kterému záznam náleží.

### Stores

Tabulka obsahuje údaje vztahující se k obchodní jednotce. Je identifikována zemí, firemním kódem a jménem obchodní jednotky. V tabulce je také uveden vedoucí prodejny.

### Seen

Po rozhovorech s reálnými uživateli vyplynulo, že by se jim hodila funkce označování, co je nové a co už je passé. Proto existuje vazební tabulka mezi uživateli a výsledky. Udává, které reporty už uživatel viděl. Je k tomu potřeba pouze identifikátor výsledku a uživatele. Je ukládán i čas provedení akce.

### Competency

Tabulka ukládá práva každého uživatele. Záznam obsahuje identifikátor uživatele a právo, které je mu přiděleno.

### Login_attempts

Pro rozpoznání pokusu o prolomení hesla uživatele aplikace jsou zaznamenávány chybná zadání do této tabulky. Obsahuje identifikátor uživatele a čas špatného pokusu.

### Users

Ukládá nezbytné informace o uživateli. Jsou to jméno, pracovní zařazení, přihlašovací jméno, heslo (otisk + sůl) a kontaktní e-mail.

## 4.4 Bezpečnost

Výsledná práce je připravena k okamžitému nasazení. K tomu přispívá i vysoká míra zabezpečení systému. V této sekci jsou popsány prostředky a postupy vedoucí ke splnění tohoto předpokladu.

### 4.4.1 Aplikace

Samotný informační systém je ošetřen proti nesprávným vstupům či snahám o poškození systému.

#### 4.4.1.1 Login do aplikace

Pohybovat se po aplikaci je umožněno pouze po přihlášení. Jakýkoli pokus o neoprávněný vstup je přesměrován na přihlašovací obrazovku.


Obrázek 4.8: Přihlašovací formulář aplikace

Z přihlašovacího hesla zadaného uživatelem je před odesláním vytvořen hash, teprve potom jsou data odeslána a na serveru ověřována. Vložené údaje jsou zbaveny speciálních znaků a ověřeny oproti databázi uživatelů. Pokusy o hádání hesla hrubou silou[10] jsou zaznamenávány do databáze. V případě špatného zadání hesla vícekrát za omezený časový interval se přihlášení zablokuje.[15].

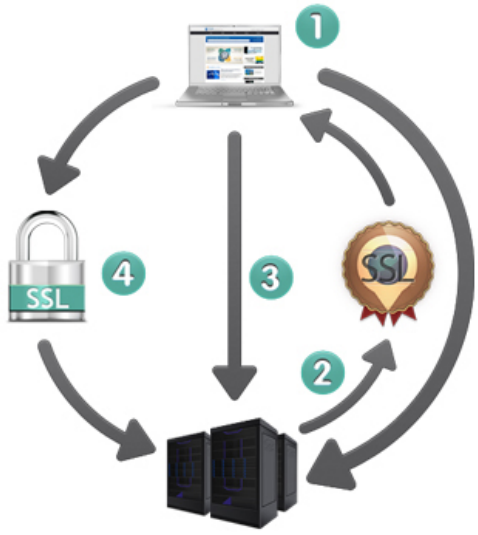
[10]Pokus o rozluštění hesla testováním všech možných kombinací

#### 4.4.1.2 Ukládání hesel

Je důležité poskytnout nejvyšší možnou bezpečnost také uživatelům. Nejcitlivějším údajem uživatele v této práci je především osobní heslo. Pro ochranu hesel je použito ukládání jako hash (jednosměrná šifra[11]). Používanou funkcí je zde SHA512, která vytváří otisk hesla fixní délky. Pro ztížení práce útočníka je hash hesla vytvořen společně s náhodným řetězcem označovaným jako salt (sůl). Vytváření otisku lze popsat jako volání funkcí „sha512(sha512(password) .salt)“. Je poté velmi těžké výpočetní silou zkoušet najít heslo s odpovídajícím hash v případě, že by se útočník k uloženým heslům dostal.

#### 4.4.1.3 HTTPS

Spojení uživatele se serverem pomocí šifrovaného protokolu HTTPS znemožňuje útočníkovi jakýkoli odposlech, zásah do dat či jejich zneužití. HTTPS šifruje data pomocí SSL nebo TLS protokolů. Jde o asymetrické šifrování[12] s použitím veřejného a privátního klíče.[16] Ukázka postupu autentizace:




Obrázek 4.9: Fungování HTTPS

#### 4.4.1.4 SQL Injection a Cross-site scripting

Jde o pokus napadení pomocí dat odesílaných aplikaci přes špatně ošetřený vstup. Všechna data přijatá z formulářů jsou zbavena speciálních znaků. Dále aplikace má v databázi nastavena pouze nezbytná práva (například příkaz DROP chybí). Vykonání nějaké nechtěné akce je tak znemožněno.

[11] Funkce, kterou lze snadno vyčíslit, ale je velmi obtížné z výsledku odvodit její vstup
[12] Metoda šifrování, ve které se pro šifrování a dešifrování používají odlišné klíče

#### 4.4.1.5 Náhrávání souborů

Soubory je velmi obtížné kontrolovat, zda jsou opravdu tím, čím se tváří. V této aplikaci jsou souborová omezení aplikována na velikost souboru a na příponu. Ve složkách, do kterých aplikace soubory ukládá není přiděleno právo spouštění. Tato kombinace je základem, ovšem v budoucnu by bylo dobré se zaměřit například na souborové hlavičky.

#### 4.4.1.6 Session Hijacking

Jde o formu útoku, kdy se útočník snaží získat identifikátor přihlášeného uživatele a pak se za něj vydávat. Toto je opět forma útoku, která se dá těžko odstranit, ale lze ji útočníkovi výrazně ztížit. Mezi hlavní body ochrany použité i v této práci patří získání údajů o počítači uživatele, hashování této informace společně s heslem a nastavení vlastního názvu sezení.

### 4.4.2 Server

Kromě systému je potřeba nezapomenout i na zabezpečení samotného serveru, jelikož z principu je důležité, aby byl veřejně dostupný.

#### 4.4.2.1 Administrátorská rozhraní

AMPPS server využívá webová rozhraní (AMPPS Home a AMPPS Admin) pro příjemnější a jednodušší správu nastavení. Tato rozhraní jsou zaheslována a nepřístupná z webu. Jsou dostupné pouze z lokální adresy. Útočník rozhraní není schopen použít.

#### 4.4.2.2 PHPMyAdmin

Databázový server a PHPMyAdmin rozhraní využívají nejnovější verze pro lepší bezpečnostní celistvost. Pro přístup do databáze existuje jen administrátorský účet a aplikační účet. Aplikační má pouze nezbytná práva pro správné fungování systému. I v případě úspěšného napadení tohoto účtu by měl útočník pouze omezené možnosti. Webové rozhraní PHPMyAdmin je také dostupné pouze z lokálního počítače.

#### 4.4.2.3 Přístup k souborům

Procházení Apache souborů je zakázáno direktivou v souboru „.htaccess“. Není možné zjistit strukturu aplikace.

#### 4.4.2.4 Vzdálená správa

Využití SSH je bezpečnou metodou pro komunikaci. Díky šifrování je možné zadávat příkazy, posílat důvěrná data pomocí SFTP či vytvořit zabezpečené spojení – neboli tunel.

V případě, že bude chtít administrátor využít grafické rozhraní pro vzdálené ovládání PC, může sáhnout po VNC. Přímé spojení se serverem není možné. Server přijme opět pouze lokální připojení. Je tedy potřeba si vytvořit tunel pomocí SSH a následně se skrz tunel připojit k VNC serveru. Toto omezení má svůj smysl, protože připojením přes tunel využijeme šifrování tunelu a tím zabezpečíme provoz jinak nezabezpečené komunikace VNC.

## 4.5 Nasazení

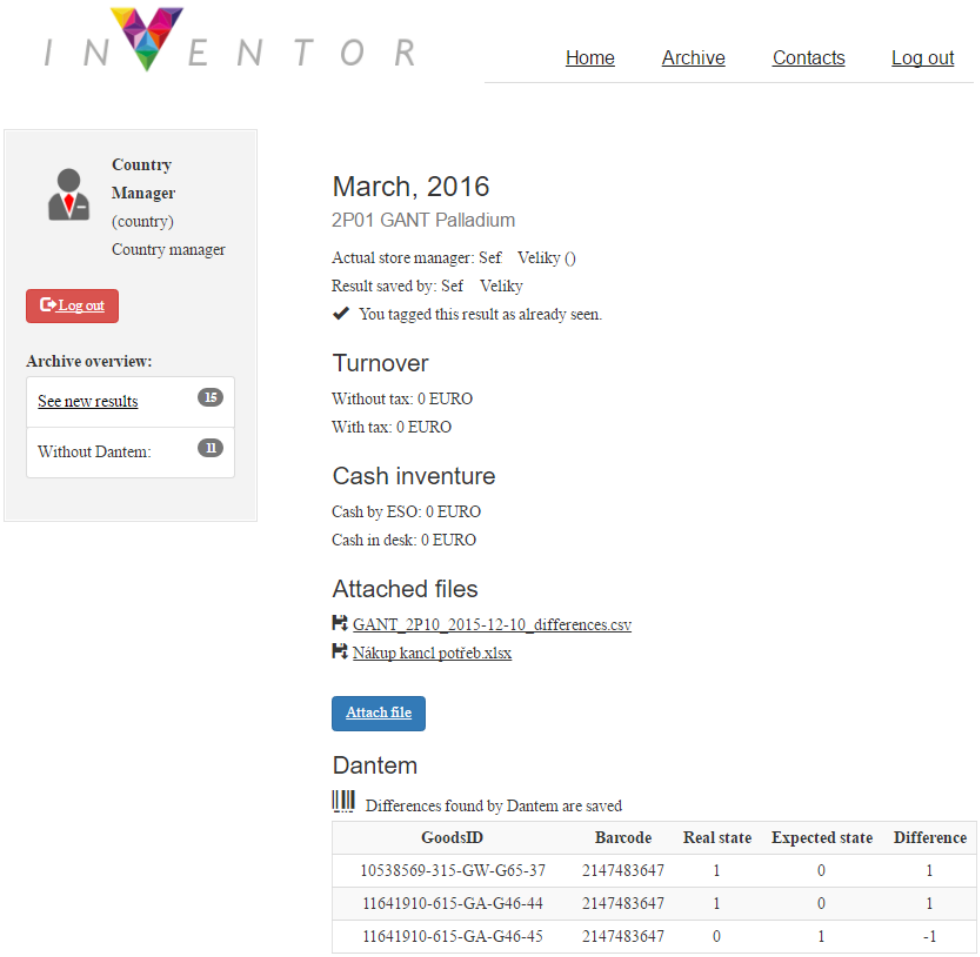
Předpokládá se distribuce aplikace pomocí CD, které je v příloze nebo předáním archivu ve formátu ZIP elektronickou cestou. *Počítač, na kterém má být aplikace provozována, musí mít funkční operační systém, trvalé připojení k internetu a musí být na veřejně dostupné doméně.* U všech nástrojů je doporučeno stáhnout nejnovější verzi.

Detailní popis všech kroků pro správné nasazení systému je v příloze C. Stručný výčet klíčových kroků:

1. Stažení a instalace serverového balíku AMPPS a databáze MariaDB.
2. Vytvoření domény a SSL záznamu na serveru Apache.
3. Upgrade databáze MySQL na MariaDB.
4. Nastavení databázového serveru – přístupy, kódování, databáze aplikace.
5. Vytvoření tabulek v databázi a přesun dat aplikace do uložiště serveru.
6. Úprava konfigurace Apache serveru.
7. Nahrání ověřeného certifikátu.
8. Příprava pro vzdálenou správu – SSL, VNC.

## 4.6 Uživatelské rozhraní

Na základě hlasování byl zvolen druhý návrh v poměru šesti ku dvěma. Tázaní uživatelé především uvedli, že druhá varianta lépe využívá prostor (první varianta vypadala „prázdně“) a při zachované přehlednosti i více informací. Ukázka implementovaného rozhraní je na obrázku 4.10.




Obrázek 4.10: Výsledné uživatelské rozhraní

## 4.7 Firewall a router

Pro správnou a bezpečnou funkci je také důležité zkontrolovat povolení komunikace na potřebných portech. Server je dostupný na portech 80 (HTTP), 222 (SSH), 443 (HTTPS) a 5900 (VNC). Naopak jakýkoli jiný je navíc a měl by být nedostupný.

Je důležité pro komunikaci těchto portů nastavit na routeru správná pravidla pro NAT – pro komunikaci se serverem umístěným v místí síti. Přesný postup tohoto nastavení je závislý na síťovém zařízení a měl by jej provést správce sítě.

## 4.8 Klient

**Uživatel** Běžný uživatel využívá webový prohlížeč pro práci s aplikací zadáním domény do adresního řádku. Nepotřebuje žádné další nástroje.

**Administrátor** Administrátor tohoto systému využívá webový prohlížeč pro běžné operace s aplikací přes webové rozhraní.

Pokud potřebuje využít pokročilejší možnosti na serveru, využívá bezpečné spojení pomocí SSH klienta. Pro Windows například PuTTY. Použije veřejnou adresu serveru, port nastavený pro SSH a nastaví tunel pro port 5900 na cíl 127.0.0.1:5900. Pro navázání spojení je potřeba použít privátní klíč. Po připojení je schopný provádět operace přes příkazový rádek. Má však možnost ovládat server pomocí grafického rozhraní připojením přes VNC na adresu „127.0.0.1“, což je možné díky tunelu. Pro pohodlný přesun souborů je možné využít například WinSCP. Přihlašování probíhá stejným způsobem jako na SSH server, pouze je potřeba vybrat použití SFTP.

## 4.9 Nový proces inventury

Po nasazení systému do provozu se změní proces inventury a sjednotí se tento postup ve všech zemích.

Prodejna údaje o tržbě uloží do nově vytvořeného systému. Podle potřeb obchodních center bude schopen administrativní pracovník či účetní obchodním centrům informaci poskytnout. Výsledek inventury hotovosti bude také zapsán do systému. Za ukládané výsledky nově ručí pouze ukládající, což je Store manager.

Inventuru artiklu v nezměněné podobě bude provádět společnost DANTEM s. r. o. a přijaté soubory (všechny tři) budou nahrány do systému. Obsah souboru označený jako „differences.csv“ bude u výsledného reportu zobrazen. Ostatní soubory, které si prodejny samostatně vytvářejí (docházka, závady,. . . ) je možné nahrát k danému reportu.

Country manager či účetní už neprocházejí e-mailové schránky, ale výsledek shlédnou v archivu se kterým pracují. V případě potřeby si prohlédnou přiložené soubory. Provedou nápravné kroky sami, bez potřeby si ihned vyměnit opět e-mail s informací o problému.

Všechny informace a soubory budou tedy uloženy v novém informačním systému.

Kapitola 5

# Testování

## 5.1 Testování použitelnosti

Cílem tohoto testování bylo zjistit, zda budoucí uživatelé informačního systému budou schopni vykonávat své pracovní úlohy. Z výsledků testů také vyplývá, zda uživatel rozumí akcím, které provádí, nebo zda naopak váhá, jak systém správně použít.

### 5.1.1 Účastníci

**Tomáš** Jako Country manager v tomto testu vystupoval muž Tomáš, kterému je 50 let, má vysokoškolské vzdělání a baví ho moderní technologie a sport.

**Alena** Jako Store manager vystupovala paní Alena, které je 39 let, má středoškolské vzdělání a ve volném čase se věnuje zahrádce, své kočce a literatuře.

### 5.1.2 Testovací scénáře

#### Přidání reportu – Store manager

Paní Alena měla jako Store manager typický úkol, kterým je přidání měsíčního reportu prodejny. Obdržela přihlašovací údaje. Dále měla k dispozici tabulku s hodnotami tržeb a výsledkem inventury hotovosti. Zprvu měla problém s orientací, kde by měla s úkolem začít. S vyplněním už neměla problém. Taktéž nahrání souborů k reportu bylo rychle splněno. Nalezení právě přidaného výsledku v archivu nečinilo problém.

Výsledek je tedy velice příznivý. Paní Alena měla pouze prvotní problémy s orientací. Všechny úkoly už ale zvládla bez zaváhání. Z testu také vyplynulo, že i když se uživatel neorientoval, nebyl ochotný číst menší popisy s vysvětlením kroků, pouze ty největší a nejvýraznější.

Po drobných úpravách systému, kdy uživatel ve vlastním menu má přímý odkaz na akci, kterou by měl provést, se provedení celého scénáře zkrátilo na zlomek času. Z formuláře byly také odstraněny delší popisky a například informace o měně je signalizována ikonou, což zrychluje orientaci.

#### Kontrola výsledků – Country manager

Po nasimulovaném nahrání výsledků z prodejen měl Tomáš jako Country manager za úkol zkontrolovat výsledky inventur prodejen a najít nesrovnalosti. Kolegové je totiž musí napravit – účetní finančně, skladník poslat nové zboží atd. Obdržel tedy přihlašovací údaje. Rychle nalezl archiv a podle časového řazení byl schopen si reporty projít a hledané informace najít. Výsledky se kterými už nechtěl pracovat si označoval jako shlédnuté.

Výsledek byl potěšující. Uživatel se dokázal rychle zorientovat a provést úkol.

## 5.2 Testování běhu aplikace v prohlížečích

Interpretace jazyka HTML je závislá na prohlížeči, který používá klient.[17] Proto je nutné funkčnost aplikace otestovat alespoň ve třech nejvyužívanějších.[18] Aplikace byla testována v prohlížečích: Chrome verze 50.0.2661.94 m, Internet Explorer verze 11.212.10586.0 a Mozilla Firefox verze 46.0.1. Během testování nebyl nalezen žádný funkční rozdíl, pouze drobné rozdíly ve vzhledu – především písmo a zobrazení nadpisů.

## 5.3 Testování bezpečnosti

Tato sekce má zjistit, zda výsledný informační systém skutečně splňuje bezpečnostní očekávání. Byly provedeny série testů, které se snaží napodobit kroky útočníka.

### 5.3.1 Webové rozhraní

V aplikaci není možné se pohybovat bez správných přihlašovacích údajů. Nezdařilo se ani přihlášení pomocí metody SQL Injection zadáváním „a’ or ’b’=’b“ ve snaze o úpravu ověřovacích dotazů na formu „"SELECT * FROM users WHERE jmeno = ’a’ or ’b’=’b’;"“, která by byla vždy správná. Útok hrubou silou na heslo uživatele aplikace vedl pouze k zablokování účtu – přístup nebyl udělen.

Ostatní formuláře také nepovolily akci, která by mohla být nebezpečná (například zadáním hodnot ve formě „a’;DROP TABLE users; –“).

### 5.3.2 Komunikace

Pomocí programu Wireshark[19] byla odchycena komunikace serveru a klienta během přihlašování. Komunikace přes protokol HTTPS byla nečitelná. Při vypnutí HTTPS (pro testování) bylo možné zachytit přihlašovací jméno, ale pouze hash hesla. Takováto data jsou při běžném provozu pomocí HTTPS dále zašifrována. To dokazuje správnou funkci ochrany přihlašovacích údajů a komunikace klientů se serverem.

```
HTML Form URL Encoded: application/x-www-form-urlencoded
> Form item: "username" = "admin"
> Form item: "password" = ""
> Form item: "p" = "c7ad44cbad762a5da0a452f9e854fdc1e0e7a52a38015f23
                   f3eab1d80b931dd472634dfac71cd34ebc35d16ab7fb8a90
                   c81f975113d6c7538dc69dd8de9077ec"
```

Obrázek 5.1: Komunikace protokolu HTTP

## 5.4 Server

Procházení složek aplikace je zakázáno. Webová rozhraní PHPMyAdmin i AMPPS Home jsou dostupná pouze lokálně. Za pomocí scanu portů[20] byly objeveny dostupné porty serveru:

- 80: služba HTTP
- 222: služba SSH
- 443: služba HTTPS
- 5900: služba VNC

Server správně odpovídá na webové požadavky HTTP a HTTPS. VNC server na portu 5900 odmítá spojení jiná než na adresu „localhost“. Na portu 222 server odpovídá na pokus o SSH připojení, které využívá veřejný klíč.

# Závěr

Cílem práce bylo vytvořit uživatelsky komfortní informační systém, který bude sloužit ke zpracování a prezentaci inventárních dat. V úvodu této práce byl analyzován proces inventarizace ve společnosti Vermont Holding a. s. Současné řešení sdílení dat formou e-mailové komunikace je nepostačující. Bylo zaznamenáno, jaká data jsou při procesu inventarizace sdílena a které osoby ve firemní hierarchii s daty pracují. Bylo tedy potřebné implementovat nový informační systém na míru.

V této práci byl navržen, implementován a následně otestován nový informační systém, dostupný přes webové rozhraní, který dokáže proces inventarizace automatizovat a nabídne uživatelsky komfortní rozhraní. Implementace splňuje všechny požadavky na ní kladené. Práce obsahuje také přehled použitých technologií včetně důležitých prvků pro zprovoznění a zabezpečení webového serveru.

V budoucnu je možné na práci navázat rozšiřováním funkcionality systému, či lepších služeb webového serveru. Nabízí se vytvoření firemního jednotného přihlašování (SSO), propojení s pokladním systémem, vytváření statistik výsledků inventur, rozšíření možností administrátorského rozhraní či upozornění uživatelů na e-mail. Server by bylo možné rozšířit například o možnost zálohování.

Realizace této bakalářské práce pro mne byla velkým přínosem. Prošel jsem všemi fázemi nového projektu. Získal jsem nové zkušenosti s návrhem pro skutečné uživatele a analýzou problému v praxi. Také jsem se dozvěděl velmi mnoho informací o technologiích využívaných na webu a naučil jsem se dané technologie nasadit do provozu.

# Literatura


[1] VAŠÍŘOVÁ, Martina. Co obnáší pozice Country Manager [online]. 2012, 9.7.2012 [cit. 2015-12-11]. Dostupné z: http://finance.idnes.cz/co-obnasi-pozice-country-manager-d4c-/podnikani.aspx?c=2002M157Z01D

[2] ITBIZ. Popis pozice - Store manager. In: Itbiz.cz [online]. web [cit. 2016-04-23]. Dostupné z: http://www.itbiz.cz/slovnik/human-resources-hr/store-manager

[3] ČÁPKA, David. MVC architektura [online]. 2013 [cit. 2016-04-25]. Dostupné z: http://www.itnetwork.cz/navrhove-vzory/mvc-architektura-navrhovy-vzor/

[4] Historical trends in the usage of server-side programming languages for websites. In: W3Techs [online]. [cit. 2016-04-25]. Dostupné z: http://w3techs.com/technologies/history_overview /programming_language

[5] PHP: PHP: Hypertext Preprocessor. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-03]. Dostupné z: https://cs.wikipedia.org/wiki/PHP#cite_note-2

[6] HTML. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-03]. Dostupné z: https://cs.wikipedia.org/wiki/HyperText_Markup_Language

[7] CSS: Using LESS. In: Bootstrap [online]. [cit. 2016-05-04]. Dostupné z: http://getbootstrap.com/css/

[8] The Best Linux Web Server Software (& Apache Alternatives) [online]. In: . [cit. 2016-05-01]. Dostupné z: http://www.makeuseof.com/tag/linux-web-server-software-apache-alternatives/

[9] PEARCE, Rohan. Dead database walking: MySQL's creator on why the future belongs to MariaDB: MySQL's creator, Michael "Monty"Widenius, is scathing on database's future with Oracle. In: COMPUTERWORLD [online]. 2013 [cit. 2016-05-01]. Dostupné z: http://www.computerworld.com.au/article/457551/

[10] GAJDOŠOVÁ, Markéta. Srovnání – kdy je lepší MySQL a kdy MariaDB? In: COMPUTERWORLD [online]. 2013 [cit. 2016-05-01]. Dostupné z: http://computerworld.cz/software/srovnani-kdy-je-lepsi-mysql-a-kdy-mariadb-50258

[11] CLARK, Jack. Google swaps out MySQL, moves to MariaDB. In: The Register [online]. 2013 [cit. 2016-05-01]. Dostupné z: http://www.theregister.co.uk/2013/09/12/google_mariadb_mysql_migration/

[12] HTTPS. In: Wikipedia: the free encyclopedia [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-05-05]. Dostupné z: https://cs.wikipedia.org/wiki/HTTPS

[13] SSH – bezpečné používání vzdáleného počítače a kopírování dat. In: Dsl.cz [online]. [cit. 2016-05-03]. Dostupné z: http://www.dsl.cz/jak-na-to/jak-na-ssh

[14] VYSKOČIL, Michal. VNC - používáme vzdálený desktop. In: ABC Linuxu [online]. 2007 [cit. 2016-05-03]. Dostupné z: http://www.abclinuxu.cz/clanky/site/vnc-pouzivame-vzdaleny-desktop

[15] How to Create a Secure Login Script in PHP and MySQL. In: WikiHow [online]. [cit. 2016-05-06]. Dostupné z: http://www.wikihow.com/Create-a-Secure-Login-Script-in-PHP-and-MySQL

[16] Jak funguje certifikát SSL OVH? In: OVH [online]. [cit. 2016-05-07]. Dostupné z: https://www.ovh.cz/ssl/funkcionalita-ssl.xml

[17] JANOVSKÝ, Dušan. Různé prohlížeče. In: Jak psát web [online]. [cit. 2016-05-08]. Dostupné z: http://www.jakpsatweb.cz/prohlizece.html

[18] NOVOTNÝ, Michal. Prohlížeče – jedničkou Google Chrome, mobilní verze získávají. In: Markomu.cz [online]. 2015 [cit. 2016-05-07]. Dostupné z: http://markomu.cz/webove-prohlizece/

[19] Wireshark aplikace. In: Wireshark [online]. 2016 [cit. 2016-05-08]. Dostupné z: https://www.wireshark.org/

[20] Network Port Scanner Tool. In: IPFingerPrints.com [online]. [cit. 2016-05-08]. Dostupné z: http://www.ipfingerprints.com/portscan.php

Příloha **A**

# Seznam použitých zkratek

| | |
|---|---|
| **AMPPS** | Apache, Mysql, PHP, Perl, Python, Softaculous auto-installer |
| **CSV** | Comma separated values |
| **CZ** | Česká republika |
| **DANTEM** | Společnost Dantem s. r. o. |
| **EAN** | European Article Number – čárový kód zboží |
| **FTP** | File Transfer Protocol |
| **GNU GPL** | GNU's Not Unix General Public License |
| **HU** | Maďarská republika |
| **HTTP** | Hypertext Transfer Protocol |
| **HTTPS** | Hypertext Transfer Protocol Secure |
| **ID** | Identification number |
| **IP** | Internet Protocol address |
| **IS** | Informační Systém |
| **JSON** | JavaScript Object Notation |
| **MySQL** | Relační databázový systém s otevřeným zdrojovým kódem |
| **NAS** | Network Attached Storage |
| **NAT** | Network Address Translation |
| **PC** | Personal Computer |
| **PDF** | Portable Document Format |

| | |
|---|---|
| **PHP** | Hypertext Preprocessor nebo Personal Home Page |
| **PuTTY** | Klientský program pro SSH, Telnet a Rlogin protokoly |
| **SFTP** | Secure File Transfer Protocol |
| **SHA512** | Secure Hash Algorithm s otiskem délky 512 bitů |
| **SK** | Slovenská republika |
| **SSH** | Secure Shell |
| **SSL** | Secure Sockets Layer |
| **SSO** | Single Sign-On |
| **SQL** | Structured Query Language |
| **TLS** | Transport Layer Security |
| **URL** | Uniform Resource Locator |
| **VNC** | Virtual Network Computing |
| **VPN** | Virtual Private Network |
| **WinSCP** | SFTP klient a FTP klient |
| **ZIP** | Souborový formát s bezeztrátovou kompresí |

Příloha **B**

# Detailní struktura aplikace

Tato příloha zobrazuje detailnější pohled na strukturu aplikace a její funkční části.

„InstalačníAdresář/www/ZvolenáDoména“

- `application` – Zdrojové kódy aplikace
  - `controllers` – Soubory zajišťující řízení požadavků
    - `controller_404.php` – Řízení vlastní stránky 404
    - `controller_admin.php` – Řízení administrátorské sekce
    - `controller_archive.php` – Řízení archivu a práce s výsledky
    - `controller_contacts.php` – Řízení výpisu kontaktů
    - `controller_files.php` – Řízení práce s ukládáním souborů
    - `controller_login.php` – Řízení přihlašování a odlašování
    - `controller_main.php` – Zobrazení hlavní stránky
  - `core` – Důležité samostatné komponenty
    - `config.php` – Nastavení přístupu k databázi
    - `connection.php` – Rodičovská třída Controller
    - `login_check.php` – Kontrola přihlášení
    - `model.php` – Rodičovská třída Model
    - `route.php` – Směrování požadavků podle URL controllerům
    - `todo_list.php` – Generování individuálního ToDo listu
    - `view.php` – Rodičovská třída View
  - `css` – Soubory vzhledu aplikace
  - `fonts` – Fonty a ikony
  - `images` – Obrázky v aplikaci
  - `js` – JavaScriptové soubory
    - `forms.js` – Validace přihlašovacího formuláře a práce s heslem
    - `sha512.js` – Implementace hashovací funkce SHA512
  - `libs` – Použité knihovny
    - `bootstrap` – Knihovna Bootstrap

„InstalačníAdresář/www/ZvolenáDoména“

- `application` ..... Zdrojové kódy aplikace
  - `models` ..... Logika aplikace
    - `model_admin.js` ..... Práce s daty v administrátorské sekci
    - `model_archive.js` ..... Práce s daty v archivu
    - `model_files.js` ..... Práce s daty při uploadu
    - `model_login.js` ..... Přihlašování, porovnávání otisků hesel
  - `upload` ..... Místo pro nahrané soubory
  - `views` ..... Zobrazení všech výstupů
    - `404_view` ..... Stránka chyby 404
    - `admin_view` ..... Stránka administrátorské sekce
    - `archive_view` ..... Stránka archivu
    - `contacts_view` ..... Stránka kontaktů
    - `login_view` ..... Přihlašovací stránka
    - `main_view` ..... Hlavní stránka
    - `new_competency_view` ..... Stránka přidávání práv
    - `new_password_view` ..... Stránka úpravy hesla
    - `new_report_view` ..... Stránka pro přidání reportu
    - `registration_view` ..... Stránka pro vytvoření uživatele
    - `result_view` ..... Stránka pro zobrazení reportu
    - `show_competency_view` ..... Stránka přehledu práv uživatele
    - `template_view` ..... Stránka šablony
    - `upload_view` ..... Stránka pro nahrání souboru
- `.htaccess` ..... Konfigurační soubor serveru
- `index.php` ..... Počáteční soubor aplikace
- `favicon.ico` ..... Ikona aplikace při zobrazení v prohlížeči

Příloha C

# Podrobný popis nasazení

1. Stáhnutí AMPPS na stránce „http://www.ampps.com/“. Poté instalace probíhá pomocí instačního wizardu (klasický způsob instalace programů na systému Windows).

2. Aplikaci AMPPS spustit jako správce.

3. V okně AMPPS v záložce PHP změnit verzi PHP na 5.6

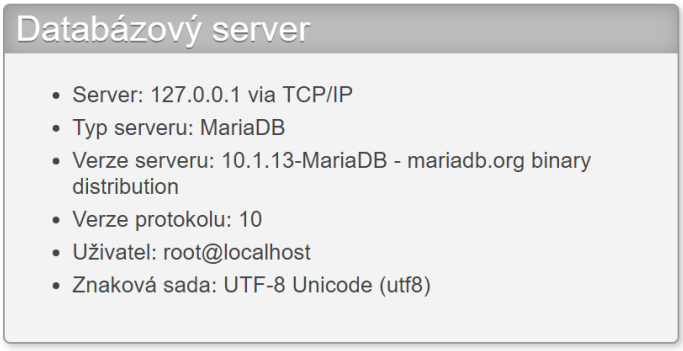
4. Pomocí okna AMPPS nebo zadáním do prohlížeče „localhost/ampps“ otevřít webové rozhraní AMPPS (nazýváno AMPPS Home).

5. Pro napodobení skutečných podmínek se může použít virtuální doména. V AMPPS Home vybrat „Add new domain“. Následně vyplnit název domény, cestu k hlavní složce a zaškrtnout „SSL entry“ a „Add entry to Host File“. Díky vybrání SSL, se připraví záznam domény poslouchající na portu 443 (HTTPS) a vytvoří vlastní certifikát („Selfsigned“ – podepsaný sám sebou, bez potvrzení autoritou. Je potřeba jej vyměnit za certifikát zakoupený u certifikační autority).

6. V sekci „Secure AMPPS“ zaškrtnout zabezpečení a nastavit heslo pro vstup na webové rozhraní. Od této chvíle je potřeba pro vstup používat jméno „soft“ a „nastavené heslo“.

7. V sekci „PHP Configuration“ nastavit maximální velikost nahrávaných souborů.

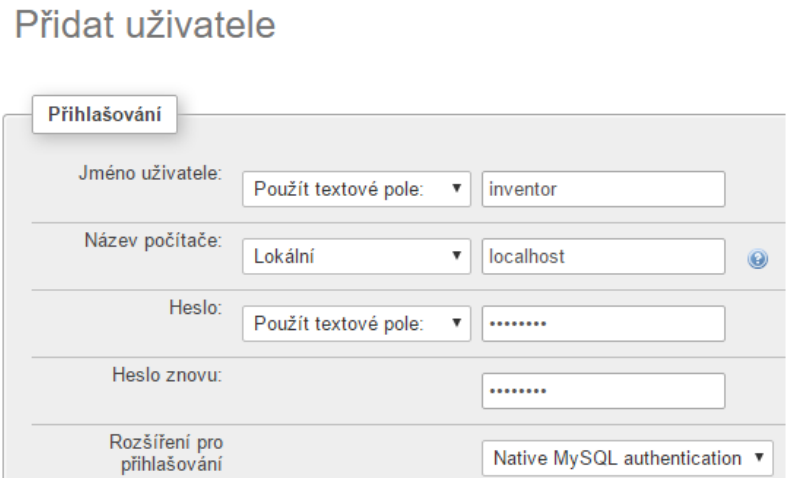
8. Pro upgrade MySQL na MariaDB je potřeba stáhnout nejnovější verzi ve formě ZIP souboru na „https://downloads.mariadb.org/mariadb/“ a obsahem archivu přepsat stávající MySQL adresář „InstalačníAdresář/mysql“. Díky stejným rozhraním a funkcím lze změnu poznat například při otevření PHPMyAdmin v přehledu serveru.




Obrázek C.1: Databázový server MariaDB

9. Dalším krokem je nastavení databáze. Je potřeba v prohlížeči otevřít localhost/phpmyadmin a v horizontálním menu v sekci „databáze“ vytvořit databázi, která se bude používat pro aplikaci. Je vhodné volit stejný název jako doména a tedy i název aplikace. Výběr kódování je určený typem ukládaných dat. V tomto případě „utf8_czech_ci“.

10. Opět v horizontálním menu, ale tentokrát v sekci „Uživatelské účty“ vytvořit uživatele, který bude reprezentovat aplikaci a bude schopen se připojit pouze z daného počítače (žádný vzdálený přístup). Tento uživatel nebude mít žádná globální práva. Pouze v záložce „Databáze“




Obrázek C.2: Vytvoření uživatele databáze

dostane právě vytvořenou databázi a nad ní pouze práva „SELECT, INSERT. UPDATE, DELETE“. Opět je to výhodné z bezpečnostního pohledu – aplikace je plně funkční, ale má pouze nezbytná práva pro

chod. Ostatní účty je vhodné projít, smazat nadbytečné a nastavit hesla zbylým. Vetšinou jde o root účty, které by měly mít silná hesla.

| | Jméno uživatele | Název počítače | Heslo | Globální oprávnění | Skupina uživatele | Přidělování | Operace |
|---|---|---|---|---|---|---|---|
| ☐ | inventor | localhost | Ano | USAGE | | Ne | Upravit oprávnění Export |
| ☐ | root | 127.0.0.1 | Ano | ALL PRIVILEGES | | Ano | Upravit oprávnění Export |

Obrázek C.3: Výsledný přehled uživatelů

11. Dále je třeba naplnit novou databázi tabulkami a základním administrároským účtem pro přístup. Ve vertikálním menu je potřeba vybrat databázi a následně v horizontálním menu vybrat import. Po zvolení importu ze souboru „inventor.sql“ a potvrzení importu je databáze připravena k použití.

12. Aplikační složku je nyní možné překopírovat do:
„InstalačníAdresář/www/ZvolenáDoména“.

13. Ve složce „www/ZvolenáDoména/Application/core“ se nachází soubor „config.php“, ve kterém je nutné změnit přihlašovací údaje do databáze a jméno databáze pro aplikaci.

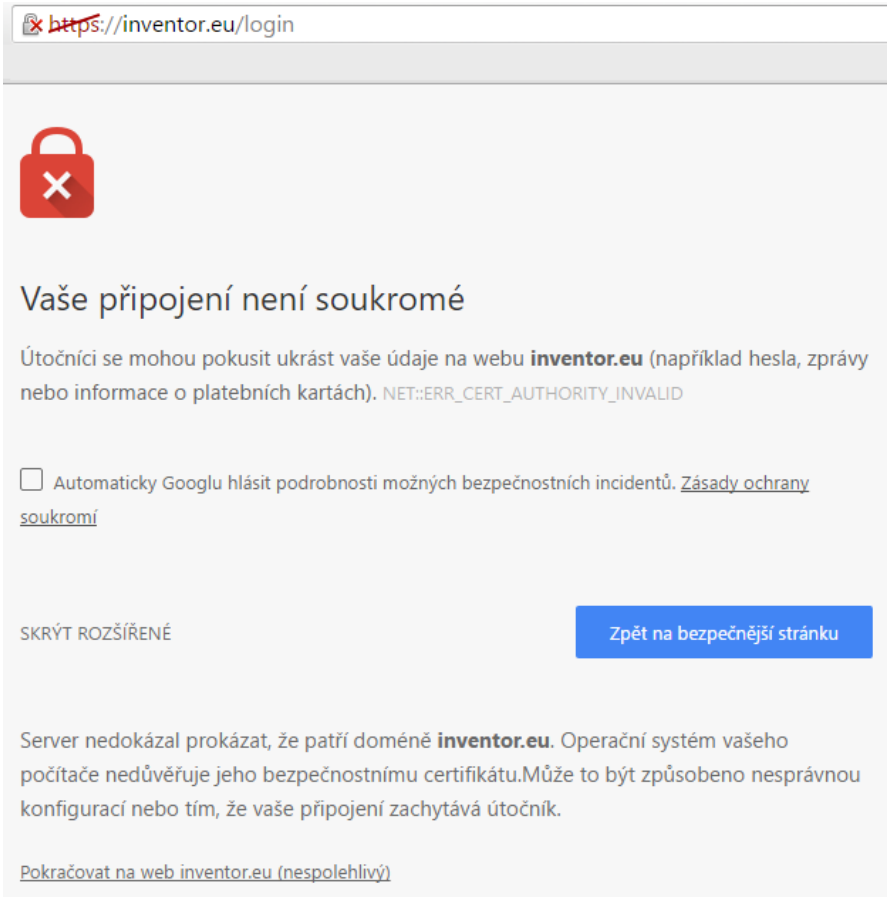
14. V tomto stádiu je aplikace funkční a dostupná při zadání do prohlížeče „doména“ nebo „localhost“. Ovšem je funkční i na „https://doména“. Tato chování je potřeba upravit, stejně jako jiné bezpečnostní prvky. *Server v tomto stádiu není připraven pro ostré nasazení do provozu!* Upozornění na špatný certifikát je očekávané – nebyl zatím nasazen certifikát ověřený certifikační autoritou.

15. Ve složce „InstalačníAdresář/phpMyAmin“ v souboru „config.inc.php“ nastavit parametr „$cfg['Servers'][i]['auth\_type']$“ na „cookie“ – parametr nastaví přístup do rozhraní PHPMyAdmin pouze nastaveným účtům. Výchozí účet má username „root“ a heslo „mysql“ (pokud stále existuje, úprava uživatelů není hotova).

16. Ve stejném souboru dále změnit „['blowfish_secret'] = 'xampp';“ na jiný řetězec a případně další atributy dle zvážení.

17. Správné nastavení lze zkontrolovat v AMPPS Home v sekci „Security center“.

18. Pro dostupnost administrátorských rozhraní pouze z lokálního PC je třeba upravit soubor:
„InstalačníAdresář/AMPPS/apache/conf/httpd.cong“.




Obrázek C.4: Upozornění na neplatný certifikát


Obrázek C.5: Správné zabezpečení pomocí hesel

a) Část „<Directory "InstalačníAdresář/Ampps/phpmyadmin»“ upravit „Deny“ a „Allow“ na „Deny from all“ a „Allow from 127.0.0.1“ Díky tomu už bude rozhraní dostupné pouze při zadání 127.0.0.1/phpmyadmin (není dostupné localhost/phpmyadmin ani jiná doména).

b) Stejný postup platí i pro AMPPS Home a AMPPS Admin. Nalézt

části:
„<Directory "InstalačníAdresář/Ampps/ampps/softaculous /enduser»“ a „<Directory "InstalačníAdresář/Ampps/ampps/softaculous»“
a změnit parametry „Deny“ a „Allow“ na „Deny from all“ a „Allow from 127.0.0.1“

19. Zakázání procházení složek serveru se provádí v souboru: „InstalačníAdresář/www/ZvolenáDoména/.htaccess“, kde se přidáním řádku „Options -Indexes“ tato možnost vypne.

20. Pro správný chod aplikace pomocí HTTPS protokolu je potřeba upravit chování domén. Je už potřeba mít vlastní certifikát pro doménu, kde bude systém umístěn. V případě nesplnění této podmínky se HTTPS spojení nevytvoří a systém bude nedostupný!

21. V části „InstalačníAdresář/Ampps/apache/conf“ se nachází složky s ceritfikáty a klíči k SSL. Sem je potřeba uložit předané soubory získané při vytváření ověřeného certifikátu autoritou (například letsencrypt.org).

22. Dále v souboru:
„InstalačníAdresář/ampps/apache/conf/extra/httpd-vhosts.conf“ v části „<VirtualHost 127.0.0.1:80>“ přidat řádek:
„Redirect / https://ZvolenáDoména“ a adresy „127.0.0.1“ změnit na „*“. Změna způsobí, že přístup přes HTTP protokol bude přesměrován na HTTPS variantu (uživatel změnu nepostřehne) a že server bude reagovat na jakékoli adrese, ne pouze při přístupu přes lokální PC.

23. V menu aplikace AMPPS je třeba zkontrolovat, zda je FTP server vypnut. Není aktuálně potřebný.

24. Pro pohodlnou správu celého serveru vzdáleně je výhodné použít další nástroje. Pro vzdálené ovládání SSH server. Na platformě Windows je to program freeSSHd. Stažení na „http://www.freesshd.com/“. Po instalaci spustit aplikaci jako správce a projít záložky nastavení. Telnet „listen only localhost“, SSH „listen 0.0.0.0“, port změnit například na 222, Authentication „password disabled, public key required“, Encryption „AES 256“, Tuneling „Allow local port forwarding, but only to localhost“ (bude potřebné pro vytváření tunelu pro VNC), SFTP složka se soubory serveru – výhodné nasměrovat na „InstalačníAdresář“, Users „Add“ uživatel pro vzdálené připojení se všemi oprávněními „Public key, Shell, SFTP, Tunnel“.

25. Pro vytvoření klíče se využívá v systému Windows PuTTY Key Generator. Po spuštění je potřeba zadat typ klíče, požadavanou délku (SSH-2

RSA, 2048) a nakonec frázi pro rozšifrování klíče. Po vygenerování uložit vypsaný veřejný klíč do souboru. Soubor pojmenovat stejně jako uživatel, kterému je klíč určen a uložit do „InstalačníAdresářopenSSHd“. Dále uložit soubor s privátním klíčem do svého PC. Při připojování pomocí PuTTY je potřeba v sekci „connection/SSH/auth“ vybrat navíc cestu k souboru s klíčem a přihlašovat se uloženým jménem a frází pro rozšifrování klíče.

26. Instalace VNC serveru začíná stažením z „ https://www.realvnc.com/download/vnc/latest/“, kde stačí free verze díky SSH tunelu. Po instalaci otevřít možnosti VNC. Změnit především v sekci „options/users“ uživatelské přístupy na VNC server. Dále v sekci „connection“ zrušit „Serve VNC Viewer for JAVA“, zamítnout default pravidlo a přidat „Add rule, connection from: 127.0.0.1, accept“. Díky těmto změnám je možné se k VNC serveru připojit jen z místní adresy 127.0.0.1, což je možné pouze přes SSH tunel. Jiná připojení jsou odmítnuta.

27. AMPPS má v oblibě zapínat pro MySQL port 3306. Ve Windows můžeme tento port zakázat ve Firewallu přidáním pravidla „Port, TCP, 3306, blokovat, vždy“.

Příloha D

# Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `src`
  - `impl` ..... zdrojové kódy implementace
    - `inventor` ..... zdrojové kódy informačního systému
    - `inventor.sql` ..... SQL příkazy pro vytvoření databáze a tabulek
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text` ..... text práce
  - `BP_Pavelek_Martin.pdf` ..... text práce ve formátu PDF
  - `ZadaniBP_Pavelek_Martin.pdf` ..... zadání práce ve formátu PDF