

Webový portál pro servis a reklamace

Web Portal for Service Support and Complaints

Martin Chumchal

Bakalářská práce
2014



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin CHUMCHAL**
Osobní číslo: **A10008**
Studijní program: **B3902 Inženýrská informatika**
Studijní obor: **Bezpečnostní technologie, systémy a management**
Forma studia: **prezenční**

Téma práce: **Webový portál pro servis a reklamace**

Zásady pro vypracování:

1. Vytvořte návrh systému pro servis a reklamaci zboží.
2. Navrhněte způsob komunikace se zákazníkem pomocí webových služeb.
3. Specifikujte platformu pro běh systému.
4. Implementujte navržené řešení.
5. Zhodnoťte chování systému v testovacím provozu.
6. Popište hlavní benefity vámi navrženého řešení. Věnujte pozornost zabezpečení systému.

Rozsah bakalářské práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. PROCHÁZKA, David. PHP 6: začínáme programovat. Praha: Grada Publish, 2012. ISBN 978-80-247-3899-4.
2. PONKRÁC, Miloslav. PHP a MySQL: bez předchozích znalostí. Brno: Computer Press, 2007. ISBN 978-80-251-1758-3.
3. KOFLER, Michael a Bernd Öggl. PHP 5 a MySQL 5: Průvodce webového programátora. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.
4. MACH, Jakub. PHP: pro úplné začátečníky. Brno: CP Books, 2005. ISBN 80-7226-834-1.
5. VRÁNA, Jakub. 1 001 tipů a triků pro PHP. Brno: Computer Press, 2010. ISBN 978-80-251-2940-1.
6. PEACOCK, Michael. Programujeme vlastní sociální síť v PHP5. Brno: Computer Press, 2012. ISBN 978-80-251-3626-3.

Vedoucí bakalářské práce:

Ing. David Malaník, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

7. března 2014

Termín odevzdání bakalářské práce:

10. června 2014

Ve Zlíně dne 7. března 2014

prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Mgr. Milan Adámek, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem bakalářské práce je vytvořit návrh systému pro servis a reklamaci zboží, navrhnout způsob komunikace se zákazníkem pomocí webových služeb a specifikovat platformu pro běh systému (PHP, MySQL). Na základě přesného návrhu bude dané řešení implementováno do reálného obchodního trhu na pozici koncového prodejce IT zboží podnikatelského subjektu Počítače Zlín. Dále bude chování systému pečlivě zhodnoceno a testováno v provozu. Posledním krokem je vytvořit základ rozhraní v jazyce PHP pro přístup zákazníků k vybraným reklamacím a servisním protokolům. V závěru budou zhodnoceny výhody a nevýhody navrženého systému.

Klíčová slova:

systém, servis, reklamace, zákazník, programování, PHP, MySQL, webový portál, databáze, dodavatel

ABSTRACT

Aim of this work is to design a system for service and claim the goods, to propose a method of communication with customers using web services and specify the platform to run the system (MySQL + PHP). On the basis of the exact solution of the proposal will be implemented in a real commercial market for the position of IT goods sales end of the business entity Computers Zlín. Furthermore, the behavior of the system carefully evaluated and tested in operation. The final step is to create the basis for an interface in PHP for customer access to the selecting complaints and service protocols and in the end will evaluate the advantages and disadvantages of the proposed system.

Keywords:

system, service, warranty, customer, Programming, PHP, MySQL, web portal, database, supplier

Rád bych touto cestou poděkoval vedoucímu bakalářské práce Ing. David Malaník, Ph.D. za odborné vedení, cenné rady a připomínky v průběhu řešení této práce.

Velký dík rovněž patří mojí rodině, blízkým přátelům a kolegům za věcné připomínky a motivující názory nepostrádající pohled z jiného úhlu.

Who is lazy to work will bear bad consequences. (Adjusted proverb)

Prohlašuji, že

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

Prohlašuji,

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

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	11
1 TEORETICKÉ ZÁKLADY SYSTÉMU	12
1.1 TVORBA DYNAMICKÝCH WEBOVÝCH STRÁNEK	13
1.2 REKLAMACE A SERVIS ZBOŽÍ V ČR	16
1.2.1 Přeprava reklamovaného zboží	17
1.2.2 Nakládání s reklamovaným zbožím	17
1.3 REKLAMAČNÍ SYSTÉM	18
1.4 ZPŮSOB KOMUNIKACE SE ZÁKAZNÍKEM POMOCÍ WEBOVÝCH SLUŽEB	18
2 PRINCIPY NÁVRHU RELAČNÍCH DATABÁZÍ	20
2.1 DATOVÝ MODEL	20
2.2 NORMALIZACE DATOVÉHO MODELU	22
2.3 DATOVÁ INTEGRITA	24
3 ANALÝZA NÁVRHU DS PRO REKLAMACI A SERVIS ZBOŽÍ	25
3.1 PROCES NÁVRHU DATABÁZE	25
3.2 IMPLEMENTACE DATABÁZE (PLATFORMA, VÝBĚR TYPU DS).....	25
3.3 NÁVRH WEBOVÉHO UŽIVATELSKÉHO ROZHRANÍ (INTERNET)	26
II PRAKTICKÁ ČÁST	27
4 NÁSTROJE PRO OFFLINE PRÁCI	28
4.1 WAMP	28
4.2 PHPMYADMIN	28
4.3 NETBEANS	29
4.4 TOTAL COMMANDER.....	29
5 SERVISNÍ A REKLAMAČNÍ SYSTÉM	30
5.1 NÁVRH DATABÁZE	30
5.2 POPIS SERVISNÍHO A REKLAMAČNÍHO SYSTÉMU	30
5.2.1 Parametry systému	30
5.2.2 Profily uživatelů	31
5.2.3 Práva uživatelů	31
5.2.4 Schéma servisní a reklamační systému	32
5.3 TABULKY DATABÁZE	33
5.4 ZABEZPEČENÍ A DATOVÁ SCHÉMA	34
5.4.1 Ošetření formulářových vstupů	35
5.4.2 Potvrzování uživatelských akcí přes email	36
5.4.3 Správa relací a session, ochrana soukromí.....	37
5.4.4 Unikátní ID relace	38
5.4.5 Zpracování dat z internetových formulářů	39
6 WEBOBÉ ROZHRANÍ REKLAMAČNÍHO SYSTÉMU	40
6.1 DEFINICE POŽADAVKŮ	40
6.1.1 Navigace portálu pro Reklamace a Servis - Počítače Zlín	40
6.1.2 Přidávání nových komentářů k reklamacím a servisu.....	41

6.1.3	Správa přihlašování	41
6.1.4	Prohledávání a filtrace v portálu	42
6.2	STRUKTURA ZDROJOVÉHO KÓDU	43
6.2.1	Přístup k databázi, vytvoření formuláře	43
6.3	SPRÁVA PŘIHLAŠOVÁNÍ	44
6.4	VÝBĚR SKRIPTOVACÍHO JAZYKA	45
6.5	TVORBA SQL DOTAZŮ	45
6.6	TVORBA PHP SKRIPTŮ A ÚVODNÍHO FORMULÁŘE	45
7	VYLEPŠENÍ SYSTÉMU A BUDOUCÍ POKRAČOVÁNÍ PRÁCE.....	47
7.1	BENEFITY NAVRŽENÉHO SYSTÉMU V POROVNÁNÍ S PAPIROVOU REKLAMACÍ	47
7.2	ZABEZPEČENÍ SYSTÉMU	47
7.3	BUDOUCÍ POKRAČOVÁNÍ PRÁCE	48
	ZÁVĚR	49
	ZÁVĚR V ANGLIČTINĚ.....	51
	SEZNAM POUŽITÉ LITERATURY.....	52
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....	53
	SEZNAM OBRÁZKŮ	55
	SEZNAM TABULEK.....	56

ÚVOD

Informační technologie (IT) se staly přirozenou součástí našeho osobního i pracovního života. Právě masový vývoj způsobil jejich rychlý rozmach a začaly být využívány v širokém pojetí. Nejvíce k tomu přispěl rozvoj výrobních technologií, přístupných programových rozhraní, které snižují nároky na obsluhu specializovaných a moderních zařízení.

Širokou oblast IT tvoří databáze (nejznámější a nejpoužívanější zejména MySQL, PostgreSQL, SQLite, MS SQL, Oracle, Access, PDO a ODBC), které představují mocný nástroj k zefektivnění zpracování, systémovému uchování dat a v neposlední řadě znovu zpřístupnění požadovaných informací. IT éra dokázala odlehčit přeplněné skříně továren díky digitalizaci informací uložených na server. Toto zabezpečilo zvýšení přehlednosti, zpracovatelnosti a dohledatelnosti s přibývajícími daty. Pokud jsou relační databáze správně navrženy, splňují základní normalizaci, jakož i datovou integritu. Základní struktury popisů ukládaných dat ušetří jejich aktualizacím a zálohám mnoho hodin práce. Tyto benefity zapříčinili, že se zpracování databáze pomocí IT vyskytuje prakticky všude. Mohou napomoci malým rozrůstajícím podnikatelským subjektům stejně tak jako velkým korporacím. Liší se pak jenom ve tvarech struktur, rozměrech, zaměření, čemu odpovídá i vybraná platforma pro běh systému.

Vývoj IT a nároky uživatelů obsluhující databáze zapříčinil rozvoj programování. Moderní objektově orientované programovací jazyky umožňují mnohem rychlejší zvládnutí programovacích technik a rychlejší vývoj aplikací. Robustní aplikace je možné vytvářet za zlomek času pomocí univerzálně použitelných komponent stejně tak i vizuální tvorbu aplikací. Programátorům pomáhá použití již naprogramovaných komponent (modulů) pro vzhled formulářů (chování ovládacích prvků formulářů), navržené technologie pro přístup k datům (BDE, DAO, ADO etc.). V neposlední řadě jsou využity pro komunikaci s periferiemi datová úložiště, IN / OUT jednotky, díky nimž lze věnovat více času samotné aplikaci a jejímu zpracování

Narůstající množství dat zpracovaných v databázích zapříčinil, že vstupuje do popředí i sféra zabezpečení databází. A to jak na softwarové tak i na hardwarové úrovni z důvodu nárůstu rizika ztráty a zneužití dat. [1]

Cílem mé práce je navrhnout systém pro servis a reklamaci zboží s využitím všech možností relačních databází. Dále bude pro zákazníka vytvořeno webové rozhraní pro

pohodlnou komunikaci se serverem. Daný systém bude sloužit podnikatelskému subjektu Počítače Zlín pro evidenci, administraci, zálohu a efektivní práci s dobropisy.

Bakalářská práce je rozdělena na dvě části – teoretickou a praktickou. V teoretické části práce budou osvětleny základní pojmy týkající se databázových systémů (základní principy návrhu relačních databází, myšlenkový datový model, jeho normalizaci včetně popisu struktury ukládaných dat a datové integrity), MySQL, PHP (serverový skriptovací jazyk) a neposlední řadě současné podoby legislativy ČR pro servis a reklamace. V závěru teorie bude nastíněn návrh přístupu k databázím prostřednictvím webového prohlížeče. V praktické části bude popsána reálná implementace servisního a reklamačního systému pomocí programů pro offline/lokální práci (Wamp, PHPmyadmin). Daný systém bude testován na lokálním disku. V případě spolehlivosti bude použit Unreal Commander pro komunikaci s ftp webhostingovým serverem. V praktické části jsou do detailů zobrazeny tabulky databázi, jednotlivé struktury webových stránek, zabezpečení a datové schéma. Dále bude provedena kompletní realizace navržené databáze do prostředí SQL serveru MySQL verze 5. V neposlední řadě představen základ pro nové webové rozhraní k databázi pro přístup zákazníků k vybraným údajům z reklamační databáze prostřednictvím internetového prohlížeče pomocí aktivních stránek ve skriptovacím jazyku PHP.

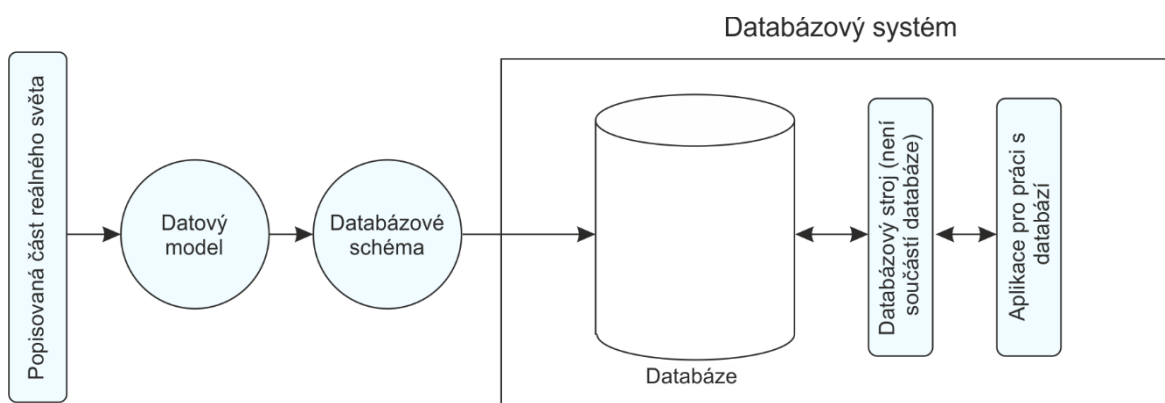
V závěru budou zhodnoceny výhody a nevýhody, které využití MySQL a PHP přináší. Rovněž budoucí vize na pokračování práce.

I. TEORETICKÁ ČÁST

1 TEORETICKÉ ZÁKLADY SYSTÉMU

Před tím než bude započata praktická část práce při tvorbě a správě databází, je potřeba vysvětlit několik základních pojmů, aby bylo jasné, jak databáze fungují. Dále je potřeba být obeznámen se základními principy, na kterých je postavena většina webových systémů. **Databáze** je možno chápat jako seznam nějakých dat v našem případě se budě jednat o data z okruhu reklamačního a servisního oddělení. Pro správu dat v databázích se používají programy, které se označují **DBMS** (DataBase Management System), neboli *system řízení báze dat*. DBMS provádí veškeré zpracování, ukládání a získávání dat. V bakalářské práci bude používán **MySQL**, který je s výjimkou některých účelů používat zdarma. Většina moderních DBMS je postavena na tzv. **relačním modelu dat**, kde jsou data uspořádána do databázových tabulek, kde každá tabulka uspořádává údaje určitého druhu. Mezi jednotlivými databázovými tabulkami existují tzv. *relace (vztahy)*. [2-6]

Pomocí relační databáze lze snadno modelovat určitou část reality kolem nás. Proto je důležitá detailní analýza této části, čímž si vytvoříme **datový model**. V samotném datovém modelu jsou definovány jednotlivé *entity, jejich atributy, obory hodnot a omezení*. Například můžeme entitou rozumět pojmu *Dodavatel*. Jeho atributy budou jméno a kontaktní adresa omezené například vždy zadaným jménem dodavatele. Dále je součástí datového modelu popis vztahů mezi entitami a omezení pro tyto vztahy. Například jedna servisní zakázka nesmí mít více dodavatelů a podobně. [1]



Obr. 1: Tvorba relačních databází

Databázové schéma je možno definovat jako fyzické rozvržení systému, seznam tabulek, pohledů a relací. Vždy vychází z datového modelu, které se implementuje na zvolený systém pro správu databází (databázový systém). Dále **databázový stroj** zajišťuje práci s databází a nakonec **aplikace**, pomocí, které uživatel komunikuje s databází, viz *Obr. 1* **Chyba! Nenalezen zdroj odkazů..** [7]

1.1 Tvorba dynamických webových stránek

Pokud bychom chtěli vytvořit jednoduché osobní stránky s několika obrázky, textovou částí a pár novinkami, lze si vystačit se statickými webovými stránkami vytvořené pomocí HTML příp. XHTML. V případě hlubšího zájmu o webové technologie, pokročilý design a náročnější vzhled Vašich stránek můžete sáhnout po CSS kaskádových stylech (Cascading Style Sheets). V případě zájmu o komunikaci, interaktivitu a dynamiku webu je vhodné vybrat **PHP** (PHP Hypertext Preprocessor) **serverový skriptovací jazyk** z čeho je patrné, že každý příkaz je zpracován na straně serveru. Co to v praxi znamená? Uživatel požádá o internetovou stránku, proběhne překlad adres pomocí DNS a jiné operace na síti související se směrováním paketů. Pokud se v kódu nachází programovací vsuvka PHP, je vyhodnocena. Poté je začleněna do HTML stránky a uživatel přijme už neměnný kód jazyka HTML. [8]

Webový server generuje obsah podle různých stávajících podmínek:

- aktuální stav databáze,
- aktuální čas a data,
- IP adresa uživatele přicházejícího na webové stránky,
- Prohlížeč, který používá uživatel přicházejícího na webové stránky a mnohé jiné.

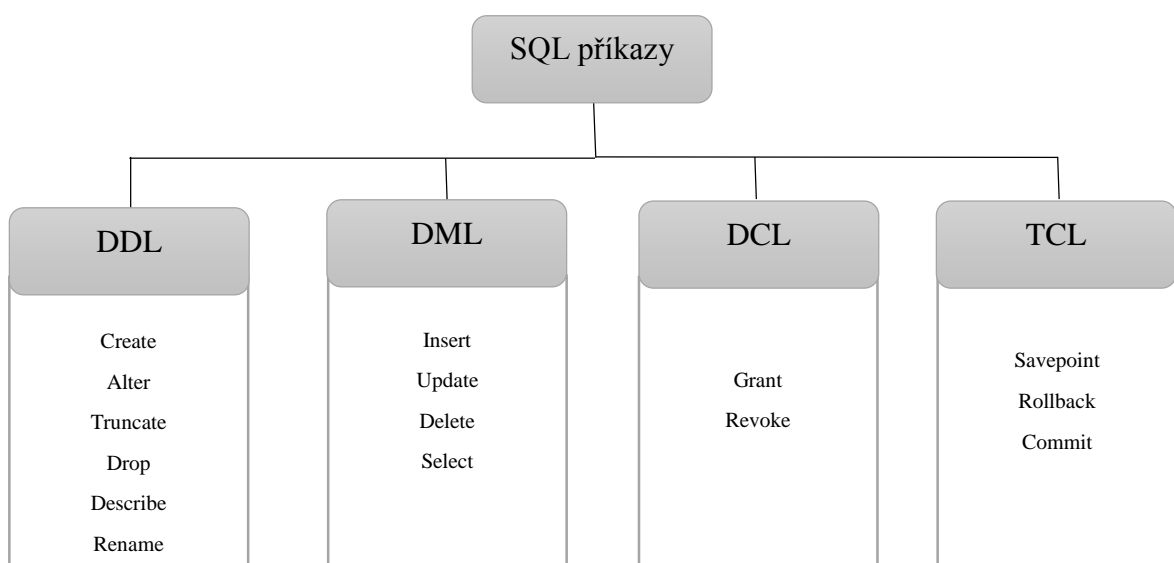
PHP se hodí vždy, pokud je na stránkách potřeba reagovat na různé požadavky zákazníka (návštěvníka) stránky a nabídnout mu co ho právě zajímá. Každý z nás má jiné priority. Pokud vezmeme například jako produkt mobilní telefon či tablet. Jeden zákazník je bude filtrovat podle ceny, jiný podle jména a značky případně podle aktuální obliby na trhu prodeje. Tyto požadavky obslouží právě skript jazyka PHP. Zákazník se obslouží sám a je na 100% spokojen. Dalším z důvodů proč používat PHP je, že využívá otevřené zdrojové kódy (tzv. open source). Jinak řečeno pokud je programátor a tvůrce webu schopný administrátor, může kód modifikovat zcela zdarma. Navíc podporuje procedurální i objektové programování. V současné době je k dispozici PHP verze 5.5. Tato technologie programování je velice oblíbená. Profesionální programátoři s dlouholetou praxí podle [9] jej popisují jako jednoduchý a intuitivní jazyk se širokou škálou podpory databázových serverů a platform operačních systémů. Jazyk PHP je primárně optimalizovaný pro web, nezatěžuje provoz a volně pracuje s typem proměnné. PHP je koneckonců celosvětově

oblíbený a má obrovskou podporu nadšenců, kteří se rádi ve fóru dělí o své zkušenosti. Případně je možnost stažení referenční příručky PHP [10] .

SQL (*strukturovaný dotazovací jazyk*) je označován za standardizovaný jazyk pro práci s daty v databázích. Většina databázových serverů ho používá a SQL se stává synonymem databáze. Správu databáze zajišťuje databázový relační systém – např. **MySQL**. Dále slouží k vytváření tabulek, k ukládání dat do tabulek. Aktualizování těchto dat, mazání tabulek či dat z nich, k různým vyhledávacím dotazům apod. Obecně jsou jako pojem databáze označovány souhrny příbuzných dat. Data v databázích jsou udržována v tabulkách. [7]

Jazyk SQL je možné rozdělit do základních podmnožin DDL, DML, DCL případně TCL:

- DDL (Data Definition Language)
 - tvorba tabulek, pohledů, vytváření indexů modifikace, odstranění objektů
- DML (Data Manipulation Language)
 - změna dat v objektech, vkládání, mazání, úprava a výběr dat
- DCL (Data Control Language)
 - řízení běhu a údržby databáze, nastavování práv jednotlivých skupin uživatelů - mohou spadat příkazy ze skupiny DDL i DML
- TCL (Transaction Control Language)
 - řízení transakce – MySQL nepodporuje



Obr. 2: Rozdělení SQL dotazů do podmnožin

V současnosti databáze zpracovávají, získávají a uchovávají data. Neztrácí svojí hodnotu, ba naopak zažívají rozvoj. Informace zpracované do použitelné podoby (odtud databáze) jsou velmi vysoce ceněny.

Pro obecné testování stránek na lokálním počítači je ještě potřeba webový server, který je možné vytvořit virtuálně právě pomocí **Apache**, na kterém je následně spuštěna služba MySQL. Apache je programový server, který zajišťuje obsluhu stránek u webových prohlížečů. Hlavní výhodou je open-source a dostupnost pro všechny platformy (Windows, Linux, OS X a další). Podle statistik z roku 2010 slouží Apache na 70% všech internetových serverech. Apache umožňuje využívání externích modulů. Podporuje různé formy autentizace, umí vytvářet logy návštěvnosti, které se aktuálně využívají v pokročilé webové analytice. [10]

MySQL je databázový systém, o kterém bylo stručně zmíněno výše. Pokud tento systém nainstalujeme na server, dá se říct, že se bude jednat o databázový server. Na všech free hosting serverech se dnes jedná samozřejmě o standard. Pokud se testuje databáze lokálně, vznikají minimální náklady, okamžitá odezva a snadná implementace systému, protože MySQL je open source. Z výše popsaného, je již patrné spojení PHP a MySQL pro lokální ladění internetových stránek. Existuje několik volně šiřitelných programových balíčků pro tvorbu webu (Easy WAMP, XAMP). Budou popsány podrobněji dále v praktické části. [11]

PHP My Admin je nástroj napsaný v PHP, který zjednodušuje správu MySQL přes webové rozhraní. Podporuje širokou škálu operací v MySQL (administrace databází, tabulek, sloupců, relací, indexů, uživatelů, práv, aj.). Tyto aplikace je možné provádět pomocí GUI, případně je stále k dispozici přímé zadávání SQL dotazů. Rozhraní obsahuje skvěle propracovanou dokumentaci, která je pravidelně aktualizována týmem podpory phpMyAdmin, který denně řeší spoustu problémů. PHP My Admin je přes 15 let starým projektem avšak je stabilní, flexibilní a stále se rozvíjí.

Apache, MySQL a PHP se často označuje jako triáda k vytváření dynamických internetových stránek. Lze si zdarma vytvořit velmi kvalitní vývojové prostředí. [12]

1.2 Reklamace a servis zboží v ČR

Jako kdysi řekl Stew Leonard: „Zákazník, který si stěžuje, je můj nejlepší kamarád. ” platí toto pořekadlo i v podniku Počítače Zlín – Martin Chumchal. Protože právě nespokojený zákazník dává odezvu co vylepšovat a kde odstraňovat chyby. Je pravda, že nekvalitní produkt, resp. reklamované zboží představuje tu nejméně vítanou zpětnou vazbu. Je ale potřebné, dívat se na reklamace s nadsázkou a optimizmem, pokud se však nejedná o spekulanta. Stížnosti a reklamace budou ve firmách nadále standardní součástí života. Proto se logicky očekává, že budou mít firmy vybudovány postupy a mechanismy pro efektivní práci se stížnostmi a reklamacemi. Právě jeden mechanismus bude prostřednictvím portálu pro reklamace a servis v této práci vytvořen a zaveden do reálného podnikatelského subjektu na trhu. [13]

Je nesmírně důležité, najmě v oblasti elektroniky a výpočetní techniky držet krok s moderními technologickými postupy a mechanismy, které zefektivňují práci. Bohužel je situace stále častěji nastavena tak, že zákazník je nalepen na obrazovce. Veškeré úkony chce řešit elektronicky vzdáleně. Zákazník vyžaduje, aby firma všechno zprostředkovala za něho, na což má ze zákona právo. Pro zaběhlou firmu je to rychlá formalita. Samozřejmě je potřeba myslet na výpadky, být připravený na standardizovaný způsob sepsání reklamačního protokolu, vyřízení a předání reklamací osobně.

Podle současného znění legislativy ČR má prodávající povinnost prodávat výrobky ve správné hmotnosti, míře nebo množství. Umožnit spotřebiteli překontrolovat si správnost těchto údajů, o což se starají správné popisky. Prodejce dále odpovídá kupujícímu za to, že prodávaná věc je při převzatí kupujícím ve shodě s kupní smlouvou, zejména že je bez vad. V České republice je prodávající povinen spotřebitele rádně informovat o rozsahu, podmínkách a způsobu uplatnění odpovědnosti za vady výrobků a služeb včetně údajů o tom, kde lze reklamaci uplatnit a o uskutečněných záručních opravách.

Právo z odpovědnosti za vady může odběratel uplatnit u prodávajícího, u kterého zboží zakoupil. Je-li však v záručním listě uveden jiný podnikatelský subjekt určený k

opravě, který je v místě prodávajícího nebo v místě pro kupujícího blíže, může uplatnit odběratel právo na opravu u podnikatele určeného k provedení záruční opravy.

V rámci reklamačního řízení se rozhoduje o tom, zda je reklamáce oprávněná, či nikoliv. Dále, zda jdou reklamované vady odstranitelné nebo neodstranitelné.

Povinnosti koncového prodejce, pokud se jedná o vady odstranitelné, jsou následující:

- povinnost odstranit vady,
- povinnost vyměnit zboží nebo vyměnit součásti, týká-li se vada jen součásti věci.

Reklamací vad zboží nebo služeb je možné uplatnit pouze v záruční době, což činí v případě elektro 24 měsíců u spotřebního zboží, případně 6 měsíců u zhotovení věci na zakázku.

1.2.1 Přeprava reklamovaného zboží

V podnikatelském subjektu Počítače Zlín – Martin Chumchal probíhá svoz případného reklamovaného zboží ihned, jak je to jen možné, tedy při další cestě do místa vzniku reklamáce. To nastane většinou velmi brzy, protože firma většinou provádí pravidelné dodávky odběratelům, se kterými spolupracuje. Jakmile je reklamované zboží naloženo, vrací se zpět do řetězce dodavatele a odtud do výrobního podniku či autorizovaného servisního střediska, kde je podrobena kontrole a podle výsledku kontroly je s ním dále naloženo.

1.2.2 Nakládání s reklamovaným zbožím

S reklamovaným zbožím se zachází podle příčiny dané reklamáce. Pokud je poškozen jen obalový materiál, je možnost dalšího užití reklamovaného produktu. Například pokud se jedná o poškrábaný obal mobilu při dopravě, je na náklady firmy tento obal vyměněn z jiného telefonu. Pokud je poškozen produkt bez zapříčinění odběratele v rozsahu, že jeho oprava se již nevyplatí, je mu produkt vyměněn za nový. V opačném případě je zákazníkovi vyplacena plná náhrada škody.

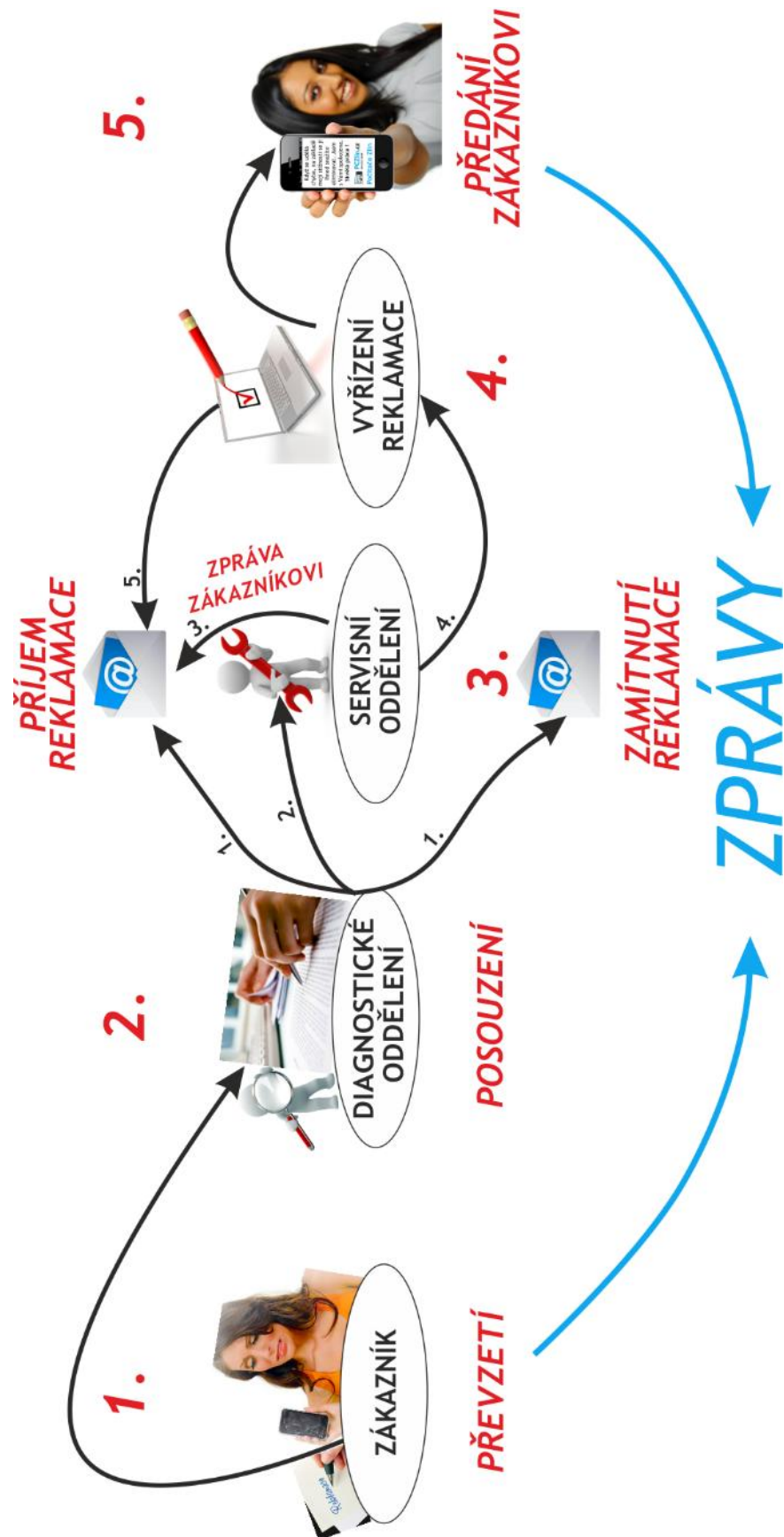
1.3 Reklamační systém

Cílem této bakalářské práce je navrhnout reklamační systém, který bude realizován jako webový portál. Avšak předem je potřeba důkladně analyzovat jednotlivé kroky reklamačního procesu od převzetí reklamace, posouzení zavinění až po vyřízení a předání zpět zboží zákazníkovi. Jelikož reklamace, záruční a pozáruční servis, spolu úzce souvisí, bylo implementováno i řešení pro servis do toho systému. Níže na Obr. 3 je detailně rozebrán každý krok, podle kterého byl sestaven portál. Jedná se o pěti krokový proces, který začíná přinesením reklamované věci zákazníkem na kamennou prodejnu. Na základě identifikace zboží, vyslechnutí opodstatnění důvodu reklamace obsluhující prodáváč udělá zápis do portálu přes “*Vstup pro Admina*”, kde se vyplní potřebné údaje jako číslo faktury, klíčové informace o závadě a základní informace o zákazníkovi u kterých je potřebná přítomnost. Zbylé údaje budou doplňovány v průběhu vyřizování.

Dále reklamační proces pokračuje na diagnostické oddělení, kde se jednotlivému reklamovanému zboží pečlivě věnuje servisní technik, který analyzuje vzniklou vadu. Jestli je vůbec reklamace oprávněná, nebo se jedná o spekulaci a vada vznikla nesprávnou manipulací. V tomto případě je reklamace zamítnuta, zákazníkovi se zasílá emailová zpráva s protokolem o odborném posouzení vady a je mu nabídnut pozáruční servis. V případě zájmu je výrobek předán servisnímu oddělení, které dostane přístup do administrace dané zakázky a může modifikovat údaje ohledně servisu. Díky navrženému systému průběžně informuje firmu a zákazníka o průběhu opravy až do úplného vyřízení. Servisní středisko předává opravený produkt prodejně počítače Zlín, která ho následně předvede a předá zákazníkovi při podpisu protokolu o vyřízení reklamace.

1.4 Způsob komunikace se zákazníkem pomocí webových služeb

Zákazník, který má vytvořený reklamační záznam je průběžně informován o každém kroku a změně, která se ho týká. K aktuálnímu stavu má možnost aktivně přidávat komentáře na reklamační oddělení ohledně průběhu vzniklých vad na produktu. V případě neuznání reklamace je mu zaslán email s dotazem na pozáruční servis. Komunikace se zákazníkem probíhá elektronickou poštou s veškerými, údaji zadanými do kořenové databáze uživatele, s nimiž bude nakládáno ve smyslu zákona o ochraně osobních údajů č. 101/2000 Sb. ve znění pozdějších předpisů.



Obr. 3: Průběh vyřizování reklamace a servisu firmou Počítače Zlín

2 PRINCIPY NÁVRHU RELAČNÍCH DATABÁZÍ

2.1 Datový model

Datový model je abstraktní část celého návrhu databáze, která popisuje možné části reality za pomoci entit, atributů, domén (oborů hodnot) a vztahů. Popisuje datové struktury a vazby mezi nimi. Zjednodušeně je to mapa databáze.

Entita je každá součást datového modelu, o které potřebujeme uchovávat nějaké informace. Seznam entit se stanovuje při zahájení prací na návrhu datového modelu (zaměstnanci, dodavatelé, reklamace, odběratelé apod.). Tento libovolný objekt reálného světa musí být rozlišitelný od ostatních entit a existovat nezávisle na nich.

Atributy jsou charakteristickými vlastnostmi zaznamenaných entit (objektu). Procházka v [1] říká, že: „*Atribut přiřadí každé entitě z množiny entit hodnotu z nějaké neprázdné množiny hodnot, nazvané doména atributu (obor hodnot atributu). Atribut je tedy zobrazení množiny entit do domény atributu.*“ Příkladem entity může být uživatel portálu, který bude obsahovat atributy Jméno, Heslo, E-mail atd. Podle zaměření navrhujeme, jak bude databáze široká, čemuž odpovídá i detailnost atributů. Vždy jde o nalezení optimálního stavu detailů atributu. Pokud bude entita zpracovávána do detailu, je to užitečné, avšak výkonově náročné. Správa databáze je složitější, čím větší množství nabízených informací je k dispozici a naopak.

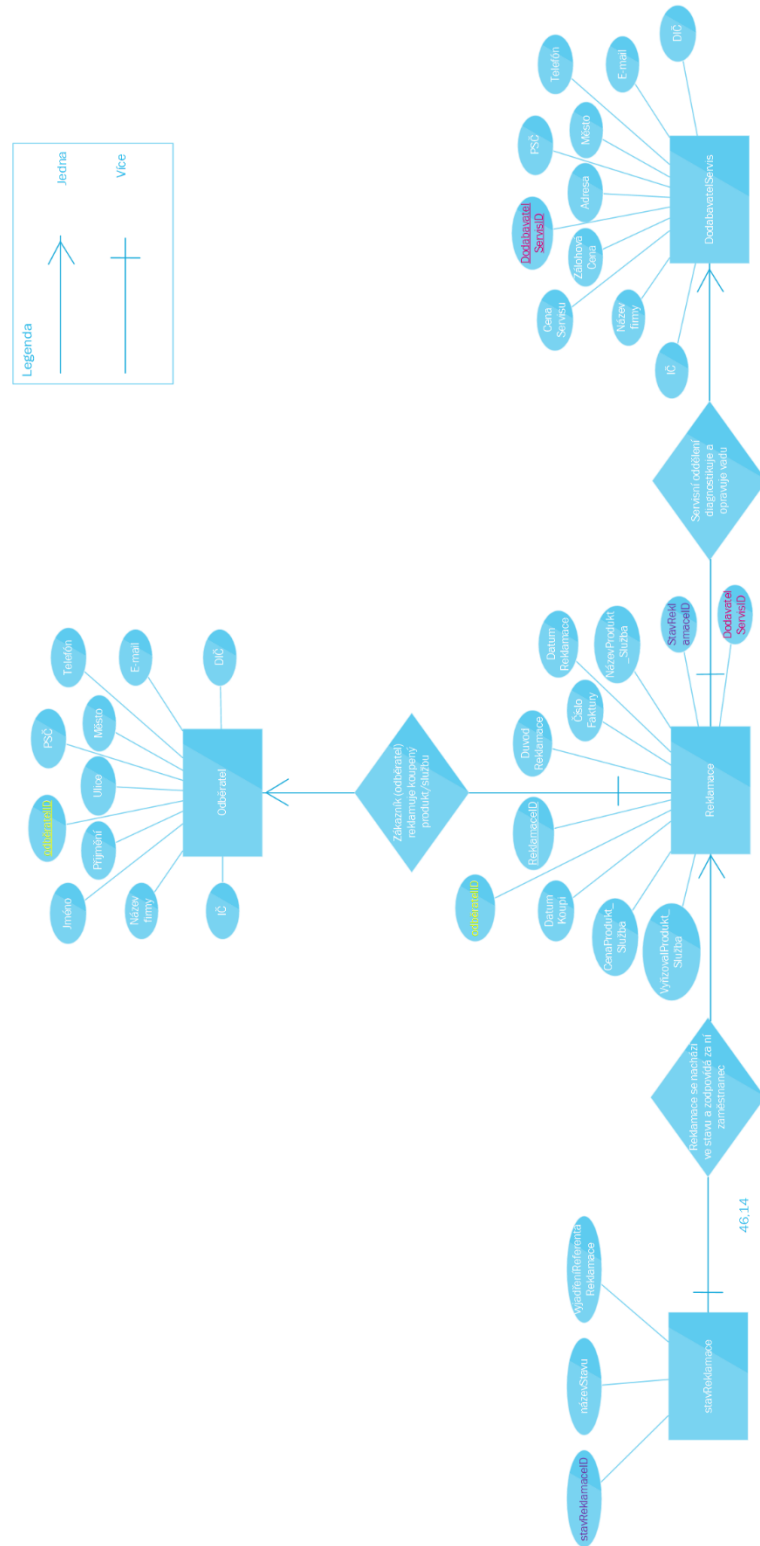
Obor hodnot (domény) popisuje typ dat reprezentující daný atribut. Je to tedy množina hodnot, které smí atribut obsahovat. *Datový typ* je fyzická reprezentace atributu (řetězec), obor hodnot je logický pojem (reklamační lhůta).

Vztahy (relace) jsou definované spojení tabulek na základě klíče. Měly by být popisem skutečných vztahů mezi jednotlivými reálnými entitami. Teoreticky lze předpokládat, že se bude jednat o vztahy mezi dvěma entitami (binární vztahy). V praxi se často setkáváme s ternárními vztahy.

Obecně vztahy mezi objekty mohou být:

- 1:1 jedna k jedné
- 1:N jedna k více
- M:N více k více

Datový model se často znázorňuje jako E-R diagram, který má podobný význam jako vývojový diagram pro návrh algoritmu. Je určen pro návrh struktury databáze a popis hlavních vlastností uložených dat v databázi. [1]



Obr. 4: E/R diagram podle Chena – diagram entit a relací části portálu pro reklamaci

2.2 Normalizace datového modelu

Návrh databáze je často podceňovanou oblastí, proto jí věnuji část své práce. Právě důkladná analýza a detailní průzkum E/R diagramu (Entity Relationship Diagrams) ¹, nám dovolí, prozkoumat veškeré skryté entity a její atributy, které odhalíme, až v průběhu programování, nebo co je horší u testování až uživatelem. Cílem ideálního návrhu je odstranit veškeré redundance (ukládání stejných údajů na více míst v databázi), pokrýt všechny možné problémy zadané oblasti zájmu a dodržovat principy normalizace.

Kupříkladu relace *dodavatelServis* a *zamestnanecServis* viz Údaje *telefon* jsou v databázi redundantní (Tab. 1). Při pokusu aktualizovat např. údaj *zamestnanecServis.telefon* je třeba upravit záznamy jak v tabulce *zamestnanecServis*, tak *dodavatelServis*. Ne že by se jednalo o složitou akci, ale programátor musí myslet na všechny záznamy, jinak by databáze ztratila svoji integritu (konzistentnost).

První, druhá, třetí normální formě:

Tab. 1: Relace *dodavatelServis* ne v INF

dodavatelServisID	zamestnanecID	nazevFirmy	adresa	mesto	PSC	IC	DIC	telefon	email	cenaServisu	zalohovaCena	stavID
1		Počítače Zlín	Okružní 4699	Zlín	760 05	1330977	C290093062	778090044, 778090055,	pczlin@email.cz	399/1h	499	1
2		NWT a.s. (PATRO)	Nábřeží 5574	Zlín, Malenovice	760 05	63469511	C263469511	570570170	obchod@patro.cz	399/1h	359	2
3		Počítače Zlín	Okružní 4699	Zlín, Brno	760 05	1330977	C290093062	778090033	pczlin@email.cz	399/1h	499	3
4		K24 International s.r.o.	Varšavská 595/88	Ostrava	709 00	27851559	C227851559	591141142,6	servis@k24.sk	329/1h	459	2
5		ExaSoft	Areál EXPO (HESPO) 114	Zlín – Malenovice	763 02	29383161	C229383161	591141685	reklamace@exaso	499/1h	299	3
6		PROBIT Elektro	Nádražní 982	Česká Třebová	560 02	67323286	C279081953	461031104	probit@probit.cz	359/1h	329	1

Tab. 2: Relace *zamestnanecServis*

zamestnanecID	titul	jmeno	prijmeni	telefon	email
1	NULL	David	Žilka	778090055	d.zilka@seznam.cz
2	NULL	Martin	Chumchal	778090033	matys@seznam.cz
3	NULL	Libor	Miček	606737235	libor.mickek@nwt.cz
4	NULL	Pavel	Grentek	724843909	pavel.grentek@nwt.cz
5	NULL	Jiří	Svoboda	570570126	jiri.svoboda@patro.cz
6	NULL	SERVIS	K24	591141142	servis@k24.cz
7	NULL	SERVIS	ExaSoft	577103686	info@h-centrum.cz
8	Ing.	Petr	Skopal	776200101	probit@probit.cz

Pro datový model se zavádějí určitá pravidla - *normalizace*, která jsou od jednotlivých úrovní normalizací k následující úrovni přísnější. Aby datový model dodržoval pravidla relační struktury databáze, musí být normalizovaný. Nejsou však povinným nebo univerzálním předpisem.

¹ Peter Pin Shan Chen zavedl poprvé v roce 1976 model entit a vztahů popisující data jako entity, jejich atributy a relace (vztahy). Současně navrhl metodu zobrazení do E/R diagramů, které se rychle rozšířily a staly se všeobecně uznávanými.

Při vytváření *plně normalizovaného datového modelu* odstraňujeme redundance tím, že rozdělujeme původní relace, avšak musí být možné tabulky zpětně spojit bez ztráty informací – bezztrátová dekompozice. [3] Z výše uvedené tabulky stačí odstranit sloupec *telefon* a rozdělit sloupec *město* na *pobočky* pomocí *městoID* a další tabulky. Získáme relaci v první normální formě (všechny atributy relace jsou definovány nad skalárními obory hodnot - doménami). Podobně se dá vyřešit i problém se sloupcem *telefon*.

Relace v první normální formě, která má navíc všechny atributy závislé na celém kandidátním klíči je ve *druhé normální formě*. Kandidátní klíč je množina jednoho nebo více atributů jednoznačně identifikující každý jednotlivý záznam relace. Např. relace *stavServis* (viz. Tab. 4) je ve druhé normální formě, protože pokud bychom uvažovali za kandidátní klíč množinu atributů *stavID*, potom atribut *nazevStavuServisu* je závislý na kandidátním klíči stejně jako atribut *casZmenyStavu*. Kandidátní klíč *stavID* sám o sobě jednoznačně identifikuje každý záznam.

Tab. 3: Relace *dodavatelServis* v 2. NF

dodavatelServisID	zamestnaneclD	nazevFirmy	adresa	mesto	PSC	IC	DIC	telefon	email	cenaServisu	zalozovaCena	stavID
1	1	Počítače Zlín	Okružní 4699	Zlín	760 05	1330977	C2900930624	778090033	pczlin@email.cz	399/1h	499	1
2	3	NWT a.s. (PATRO)	Nábřeží 5574	Zlín	760 05	63469511	C263469511	570570170	obchod@patro.cz	399/1h	359	2
3	2	Počítače Zlín	Okružní 4699	Zlín	760 05	1330977	C2900930624	778090033	pczlin@email.cz	399/1h	499	3
4	6	K24 International s.r.o.	Varšavská 595/88	Ostrava	709 00	27851559	C227851559	591141142	servis@k24.sk	329/1h	459	2
5	7	ExaSoft	Areál EXPO (HESPO) 114	Malenovice	763 02	29383161	C229383161	591141685	reklamace@exaso	499/1h	299	3
6	8	PROBIT Elektro	Nádražní 982	Česká Třebová	560 02	67323286	CZ790819532	461031104	probit@probit.cz	359/1h	329	1

Tab. 4: Relace *stavServis*

stavID	nazevStavuServisu	casZmenyStavu
1	prijato	2014-01-01 09:00:00
2	servisovano	2014-01-08 14:00:00
3	vyrizeno	2014-03-22 16:00:00
4	servisovano	2014-01-08 14:00:00

Normalizace vyšších forem je vyžadována pouze u speciálních případů (3. NF stanovuje, že všechny neklíčové atributy musí být vzájemně nezávislé. 4. NF, že relace popisuje pouze příčinnou souvislost mezi klíčem a atributy a 5. NF, že relaci již není možno bezztrátově rozložit). [7]

2.3 Datová integrita

Datová integrita zabezpečuje, že uložená skutečná fyzická data budou správná. Správnost je zabezpečena s využitím pravidel, pomocí nichž databáze zajišťuje alespoň částečnou správnost a věrohodnost dat. Může jít o omezení na úrovni hodnot atributů pomocí domény (oboru hodnot) nebo vztahů (referenční integrita). Pokud se zabezpečuje integrita jednotlivých entit v systému, mluvíme o entitovém omezení (primární klíč) – jednoznačná identifikace každého záznamu v dané entitě. Povolení hodnot NULL pro atributy.

Referenční integrita zabezpečuje a udržuje relace mezi dekomponovanými relacemi. Ztráta těchto vazeb by zapříčinila nekonzistentnost a nepoužitelnost databáze. Nelze kupříkladu odstranit záznam z relace *stavServis* z výše uvedeného příkladu, aniž by nedošlo k vymazání souvisejícího záznamu z relace *dodavatelServis* v příkladu *Tab. 3*

3 ANALÝZA NÁVRHU DS PRO REKLAMACI A SERVIS ZBOŽÍ

Pro analýzu a návrh *db* se využívá evoluční vývojový model, u kterého se nejprve provede předběžná analýza celého systému. Poté návrh architektury celého systému a následuje spirálový vývoj jednotlivých komponent systému, který přináší flexibilitu při vývoji.

3.1 Proces návrhu databáze

Zpracování návrhu databáze bylo prováděno v následujících krocích:

1) Parametry systému

Každý projekt by měl začít definicí cílů, které se snažíme naplnit, což nám stanovuje i záběr systému.

2) Pracovní procesy

Dále je potřeba dopředu stanovit seznam pracovních procesů, pro která mají data ukládaná v databázi sloužit. Tedy nejen ukládání a načítání dat, ale napomáhat jedné nebo více dalších činností.

3) Datový model

Myšlenkový datový model nám pomůže s definicí množin tabulkových struktur a s vyjádřením využití dat v celém systému.

4) Databázové schéma

Převedení datového modelu do fyzických popisů tabulek a architektury dat je závislé na záběru systému a finančních schopnostech zadavatele projektu.

5) Uživatelské rozhraní v prostředí internetu

Přístup k uživatelům zajistí právě uživatelské rozhraní, které musí být navrženo intuitivně a s největší možnou vypovídající hodnotou. Je třeba spolupracovat s budoucími uživateli systému již v průběhu návrhu uživatelského rozhraní.

3.2 Implementace databáze (platforma, výběr typu DS)

V průběhu návrhu databáze se zjišťuje její záběr, podle kterého se určí typ použité fyzické databáze. V dnešní době je široký výběr a proto je třeba zvážit několik aspektů:

- Požadavky na rychlost, možnosti a samotnou platformu databáze

- Velikost databáze
- Cena

Níže je uvedený příklad používaných databází:

- Lokální:
MS ACCESS, PARADOX, DBASE
- SQL servery:
MySQL , MS SQL Server, Oracle, Sybase, InterBas ...

3.3 Návrh webového uživatelského rozhraní (internet)

Z důvodu širokého pokrytí spolehlivého připojení k internetu je možné databázi zpřístupnit prostřednictvím dynamických webových stránek. Vývoj skriptovacích jazyků pro tvorbu webových prezentací umožňuje vytvářet aplikace s uživatelským přístupem přes internetový prohlížeč.

Výhoda webových aplikací:

- Dostupnost
Možnost použití na místech s pokrytím internetu s prohlížečem.
- Aktualizace a správa aplikace
Aktualizace a správa probíhá na straně serveru, klient si nic neinstaluje.
- Nezávislost na platformě
Klientská stanice může být počítač, notebook, tablet, mobilní telefon.

Portál pro servis a reklamace je typickým příkladem databázové aplikace pro webové prostředí s aktivním obsahem. Nejpoužívanější skriptovací jazyky pro tvorbu dynamického obsahu webu jsou již zmiňovaný PHP, ASP, resp. ASP. NET. Nástrojů pro vývoj ve skriptovacích jazycích je na trhu dostatek – Eclipse, Microsoft Visual Studio nebo NetBeans. V našem ladění aplikace je použit komplexní nástroj NetBeans.

II. PRAKTICKÁ ČÁST

4 NÁSTROJE PRO OFFLINE PRÁCI

V praktické části bych v úvodu představil nástroje, které byly použity při programování webového portálu. Portál byl vyvíjen a testován nejdříve na lokálním serveru. Nedocházelo tak ke komplikacím při online práci, čímž se ušetřil čas strávený připojováním k FTP serveru (každá drobná změna a režie internetu obecně). Všechny použité programy v jejich posledních verzích jsou k nalezení na CD, které je součástí BP.

4.1 WAMP

Než webový portál opustil svoji vývojovou fázi, prošel množstvím postupných návrhů, testování a vylepšování. Původní návrh není veřejnosti přístupný (viditelný pro běžné návštěvníky serveru). V této práci byl použit software umožňující simulovat existenci webového serveru. WAMP je jeden z nejpoužívanějších off-line webových lokálních serverů. Jeho název vznikl podle prvních písmen klíčových nástrojů v balíčku, které byly popsány:

“W” jako **Windows**,

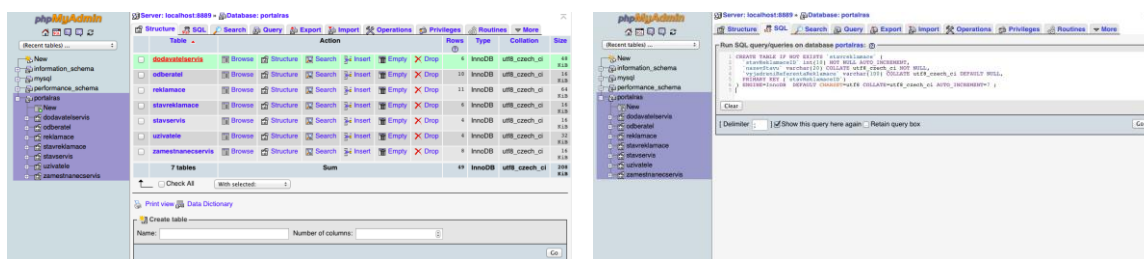
“A” jako **Apache**,

“M” jako **MySQL**,

“P” jako **PHP**.

4.2 PhpMyAdmin

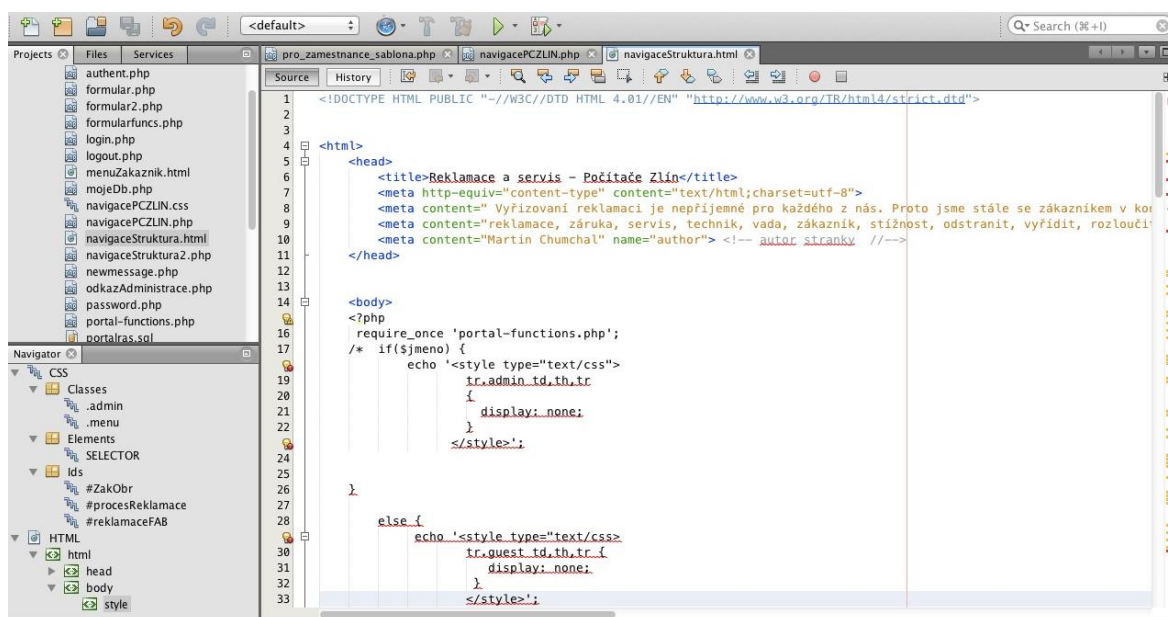
Dalším skvělým nástrojem, který mi byl nápomocný při testování SQL dotazů je PhpMyAdmin. Aplikace je specifický nástroj pro správu MySQL databáze. Software je volně šiřitelný pod licencí GNU/GPL. Rozhraní aplikace je plně lokalizováno do češtiny. V případě, že využíváte webhostingové služby, standardně je implementováno u většiny poskytovatelů. Webové rozhraní nástroje phpMyAdmin je zobrazeno na následujícím obrázku:



Obr. 5: Ukázka nástroje phpMyAdmin na správu a administraci databáze přes Web.

4.3 NetBeans

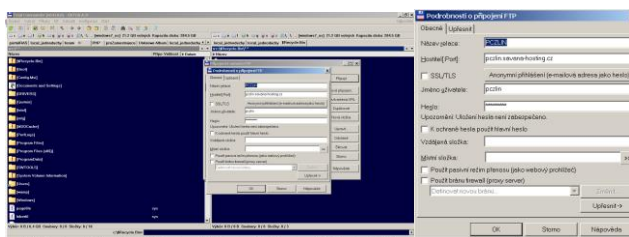
NetBeans v aktuální verzi 8.0 nabízí vývojové prostředí pro PHP aplikace. Součástí je i podpora pro XHTML, CSS, JavaScript, XML a SQL. Úplný seznam podporovaných technologií lze nalézt na *oficiálních stránkách projektu*. NetBeans podporuje práci v panelech, široké možnosti nastavení, podporu ladícího nástroje (debugger), napovídání funkcí a metod, tříd, barevné označování syntaxe, vlastní systém pro ukládání jednotlivých verzí pro případný návrat k předchozím úpravám a další funkce, které jsou k nalezení na *oficiálních stránkách projektu*.



Obr. 6: Vývojové prostředí NetBeans 8.0 s podporou PHP a jiných webových technologií.

4.4 Total Commander

Total Commander je populární souborový manažer s mnoha užitečnými funkcemi. Mimo jiné umí se zároveň připojovat na FTP jako klient, který podporuje i SSL/TLS komunikaci. Tento nástroj pro administraci byl použit pro nahrávání souborů na FTP webhostingu.



Obr. 7: Ukázka práce s FTP managerem v Total Commanderu

5 SERVISNÍ A REKLAMAČNÍ SYSTÉM

5.1 Návrh databáze

Databáze *reklamacniportal* obsahuje sedm tabulek ve formátu InnoDB (MyISAM podporuje fulltextové vyhledávání). Všechny sloupce s řetězci uvnitř používají znakovou sadu *utf8_czech_ci* a porovnání *utf8_czech_ci*. Schéma databáze je naznačeno na otázku níže:

5.2 Popis servisního a reklamačního systému

Ke každé reklamační zakázce musí být zaměstnanec schopný dohledat:

1. Fakturu nebo paragon (identifikační číslo prodeje).

Dále musí vyplnit a vytisknout po dvou stejnopisech:

2. Protokol o převzetí k reklamaci (servisu).

V případě záruční opravy zákazník obdrží:

3. Potvrzení o vyřízení reklamace (servisu).

V případě pozáruční (mimo záruční) opravy nebo nadstandardní reklamace:

4. Daňový doklad o provedených opravách.

V souvislosti s požadavky zákazníků na online přístup k vlastním protokolům o převzetí a vyřízení reklamace (servisu), budou vybrané údaje zpřístupněny prostřednictvím PHP skriptů přes webové rozhraní.

5.2.1 Parametry systému

Při zahájení práce a v jejím průběhu byly stanoveny následující požadavky, které má výsledný systém plnit:

- Evidence reklamací a servisů
 -
- Evidence zákazníků (odběratelů)
- Evidence servisních středisek (dodavatelů)
 - Dodavatel je uživatel systému s přiděleným právem vytvářet reklamační a servisní zakázky

- Dodavatel je podle reklamací služeb příslušně oprávněn k úkonům, ze systému je viditelná odpovědnost za vadu v případě reklamace (v případě služeb)
- Evidence reklamačních a servisních protokolů
 - Stanovení viníka škody (špatné nacenění, opomenutí objednávky subdodávek – příslušenství atd.)
 - Popis závady s vyjádřením servisního technika
- Tiskové sestavy
 - Jednotlivé protokoly o převzetí a vyřízení reklamace (servisu)
 - Přehledy reklamací a servisů

5.2.2 Profily uživatelů

Zaměstnanci s přístupem pro vkládání dat do databáze jsou filtrováni podle *adminID*, kde mají vyhrazeny první tři ID a podle toho jsou rozlišována i práva:

Tab. 5: Profily uživatelů s oprávněním přístupu

<i>Martin</i>	<i>ID=1</i>	<i>Super Admin (Správce)</i>
<i>Lukas</i>	<i>ID=2</i>	<i>Servisní technik (Posuzovatel)</i>
<i>David</i>	<i>ID=3</i>	<i>Logistik (Přijímatel reklamace)</i>
<i>Ostatní</i>	<i>ID=4+</i>	<i>Zákazník (Odběratel)</i>

5.2.3 Práva uživatelů

Jsou rozlišována i práva:

- *Super Admin (Správce)*
- *Servisní technik*
- *Logistik (prodejce přijímatel reklamace)*

5.3 Tabulky databáze

Tabulka uživatel:

V této tabulce se nachází data všech uživatelů portálu, kteří se do něj přihlásili. Sloupce s názvy *jmeno*, *heslo* a *email* asi nepotřebují žádné další vysvětlování. Pro sloupec *jmeno* je nastaven index *UNIQUE*, který zabraňuje tomu, aby dva uživatelé měli stejné přihlašovací jméno. Sloupec *ts* je typu *TIMESTAMP* a ukládá se v něm čas posledního přihlášení. Sloupec *active* udává, zda lze účet použit (jestli je aktivní). Sloupec *authent* konečně obsahuje náhodný řetězec, který se mimo jiné používá při ověřování pomocí e-mailu. V této tabulce mají administrátory Martin, Lukas, David nastaveno `<uzivatelID<3` .

Tabulka odběratel:

Tato tabulka obsahuje základní údaje o koncovém zákazníkovi nebo firmě (*jmeno*, *prijmeni*, *ulice*, *město*, *telefon*, *email*). Ve výchozím stavu jsou *PSC*, *nazevFirmy*, *IC* a *DIC* defaultně nastaveny na *NULL*, asi nepotřebují žádné další vysvětlování. Je možné využít automatické doplňování pomocí PHP, když se zadá *IC* a *DIC* nebo název firmy.

Tabulka stavServisu:

Tato tabulka obsahuje všechny stavy, ve kterých se může nacházet servisní zakázka (*přijato*, *servisováno*, *vyřízeno*). Ke každému stavu servisu se ukládá čas změny stavu.

Z technického hlediska není problém rozšířit celý projekt tak, aby jej mohl spravovat jenom jeden administrátor. Spíš jde o problém, jestli má vyšší práva servisní technik nebo zprostředkovatel servisu. Proto budou nastavena stejná práva. V tabulce *stavServisu* se kromě primárního klíče *stavID* nevyskytují žádné další indexy, protože tabulka obsahuje velmi málo záznamů.

Tabulka Zamestnanec(servisni_technik):

Tato tabulka obsahuje základní informace o zaměstnancích servisu. Jedná se o statickou tabulku (neměnná data), které jsou jednorázově nahrány do databáze a nejsou v průběhu aplikace měněny. Obsahuje sloupce se základními informacemi o každém zaměstnanci (*titul*, *jmeno*, *prijmeni*, *telefon* a *email*), k čemu asi není třeba žádné další vysvětlování.

Tabulka dodavatelServisu:

Tato tabulka obsahuje základní informace o servisu, ve které jsou propojeny stavy servisu s dodavatelem servisu. Obsahuje sloupce se základními informacemi o každém servisním středisku (*nazevFirmy*, *adresa*, *město*, *telefon* a *email*). V případě pozáručního servisu výši ceny (*cenaServisu*) a poplatků za diagnostiku závady (*zalohovaCena*). Eviduje index servisní zakázky (*dodavatelServisID*), stavu (*StavID*) a zaměstnance (*zamestnanecID*), který servis vyřizoval. *StavID* je cizí klíč, kde je uložen jeden ze třech možných stavů.

Tabulka stavReklamace:

Tato tabulka obsahuje všechny stavy, ve kterých se může nacházet reklamační proces (**PŘIJATA, ZAMÍTNUTA, ČEKÁ NA ROZHODNUTÍ – do 3 dnů, VYŘIZUJE SE DO TŘICETI DNŮ, VYŘIZUJE SE NAD TŘICET DNŮ a VYŘÍZENA**). Ke každému stavu servisu se ukládá čas změny stavu a vyjádření zaměstnance zodpovědného za reklamacii, kde je možno psát poznámky v průběhu reklamačního procesu (*vyjadreniReferentaReklamace*).

Tabulka reklamace:

Tato tabulka obsahuje reklamační a servisní záznamy. Na sloupec *reklamaceID* je nastaven primární index. Ve sloupci *stavReklamace_ID* nalezneme aktuální stav reklamacii s popisem referenta. Sloupec *dodavatelServisID* udává, které servisní středisko vyřizovalo servisní zakázku, případně posuzovalo nárok na reklamacii použitého zboží. Sloupec *odberatelID* udává zákazníka, který čeká na vyřízení.

5.4 Zabezpečení a datová schéma

Zabezpečení je prováděno správou uživatelů a omezení jejich práv se určuje podle profilu uvedených výše. Z hlediska databáze je soubor *password.php* oddělen a přesunut do zabezpečené složky na serveru. Všechny přihlašovací údaje se ukládají do *\$_SESSION* nebo pokud je povolena v prohlížeči možnost cookies, tak do *\$_COOKIE*, kde mají životnost 14 dní. Pro okamžité smazání je v hlavičce možnost odhlášení. Proměnné *\$_SESSION* se tím odstraní a u *\$_COOKIE* se nastaví na hodnotu *FALSE*. Zavolání stránky vás poté přesměruje na úvodní navigaci.

5.4.1 Ošetření formulářových vstupů

Ve zdrojovém kódu bylo myšleno i na zajištění konzistentnosti dat pomocí funkce **validaceDat()** pro kontrolu validity, která vrací hodnotu **TRUE** nebo **FALSE**. To podle toho, jestli se nevyskytla náhodná chyba při vkládání dat. Tato funkce ověřuje pomocí php funkce **preg_match()**, jestli zadané uživatelské jméno nebude prázdné a nebylo již použito. Z hlediska bezpečnosti je nutné zadat hesla opakovaně a v neposlední řadě se kontroluje správná struktura e-mailové adresy. Kontrola však proběhne až poté, co uživatel potvrdí prostřednictvím zadaného e-mailu zaslaný odkaz. Pokud se nevyskytla žádná chyba, pak se data uloží do tabulky uživatelé podle zadaných řetězců (*jméno* a zašifrované *heslo* pomocí SQL funkce **PASSWORD()**). Navíc zadávané pole pro heslo má nastaven typ password, kdy uživatel nevidí psaný text, ale jenom hvězdičky.

U formulářových vstupů je kromě registrace použita druhá funkce **validaceDatFV()**. Ta vrací hodnotu 0 nebo 1 podle toho, odpovídá-li vstupní hodnota zvolenému typu. Je v ní použita i PHP funkce **ereg()**, která porovnává řetězec s regulárním výrazem. V typu text jsou navíc povoleny znaky pro korektní gramatiku. U čísel je volitelné znaménko + a -.

```
function validaceDatFV($typ="text",$value, $require=0) {
switch($typ){
case "text":
if($require==0){
return erereg("[a-zA-Z0-9 ěščřžýáíéóúůďťň\?,\.,\|:\|*|- \+]*$", $value);
}
else {
return erereg("[a-zA-Z0-9 ěščřžýáíéóúůďťňĚŠČŘŽÝÁÍÉÓÚŮĎŤŇ\?,\.,\|:\|*|- \+]+$", $value);
} break;
case "email":
return erereg("^[a-zA-Z0-9 ěščřžýáíéóúůďťňĚŠČŘŽÝÁÍÉÓÚŮĎŤŇ\?,\.,\|:\|*|- \+]+$", $value);
break;
case "cislo":
if($povinne==0){
return erereg("[\+|-]?[1234567890]*$", $value);
}
else {
return erereg("[\+|-]?[1234567890]+$", $value);
}
);
```

```

break;
}
}

```

Obr. 9: Ošetření formulářových vstupů

5.4.2 Potvrzování uživatelských akcí přes email

Po dokončení registrace uživatele nebo přidání zakázky je odeslán potvrzovací email. Tento email obsahuje kompletní shrnutí informací o reklamaci, případně servisu. Hlavním důvodem, proč zasílat zákazníkovi kontrolní email je ten, že ověří, zda uživatel zadal opravdu svoji emailovou adresu, ke které má přístup pouze on. Jedná se v podstatě o jedinou povinnou položku, která ověřuje identitu uživatele. V každém potvrzovacím emailu je odkaz, který obsahuje jedinečný identifikátor vytvořený díky `MT_RANDOM()`. Jedná se o implementovanou funkci jazyka PHP pro tvorbu pseudonáhodného čísla. V tabulce *uzivatele*, sloupec *active*, má defaultně nastavenou hodnotu `NULL`, která se změní na `jedna` v případě ověření odkazu v e-mailu v této podobě:

```
http://pczlin.cz/authent.php?uzivatelID=$uzivatelID&authent=$authent
```

Obr. 10: Potvrzování uživatelských akcí přes email

Po ověření se změní proměnná *active* z hodnoty `NULL` na `1`:

Tab. 6: Tabulka uživatele, která zabezpečuje správu portálu a rozdělení práv

uzivatelID	jmeno	heslo	ts	email	active	authent
1	Martin	*B6B672B3F814EC98AEA8F3CD292D5EB02739439	2014-06-04 19:36:50	matsys@seznam.cz	1	122345712
2	Lukas	*99DC65EB06940BE8D34EDE7F0EEF22860B9DF34	2014-06-04 19:37:39	lukassino@gmail.com	1	842345721
3	David	*37DD18E63CE0D8A70A581AD3E56A9EA3BC3C6A5	2014-06-04 19:36:31	david.pilr@gmail.com	1	452125753
4	Hans	*A8FC7E6BAC7B874A1D9B3B1CC825271FD3F6BE8	2014-06-04 19:35:38	hanszimmer@email.cz	1	815953212
5	Ludmila	*6CA98768494B3A329380DF50924700D2A6D323B	2014-06-04 19:36:04	liduska8504@gmail.com	1	381253756

Uživatelé, kteří do 12 hodin svůj účet neaktivují, budou z tabulky uživatelů odstraněni. Tím se zabrání hromadění neplatných přihlašovacích údajů a zbytečné tak zamezíme neplatným uživatelským jménům.

Navíc text v popisu prochází před výpisem funkcí `htmlspecialchars()`, která převede potenciálně nebezpečné symboly na odpovídající entity. Je tím zaručeno, že se nebezpečný skript neprovede a odkazy jsou také převedeny do textové formy. Je třeba poznamenat, že uživatel tím není nijak omezen při výběru znaků.

5.4.3 Správa relací a session, ochrana soukromí

Bezpečnost dat na webových stránkách je úzce spojena i se způsoby, jakým si webový server ukládá informace na počítači klienta a jak je volá. Jedna z možností je cookie, avšak tady tkví nevýhoda v tom, že se neví, jestli byly po ukončení relace cookies smazány. Právě fakt, že cookies jsou uloženy na pracovní stanici klienta, kde jejich může kdokoliv změnit, popřípadě odstranit se používá Session. Navíc lze cookies vytvořit nebo upravit jenom před vysláním obsahu stránky podle definice HTTP. Pro každou relaci (Session) se klientovi přiřadí nějaký jednoznačný identifikátor (Session ID), pomocí něhož jej server pozná. Všechny informace tedy zůstanou na serveru, což je výhodné jak z hlediska zabezpečení, tak z hlediska práce s proměnnými. Pokud má uživatel povoleny cookies tak se tam uloží jenom zdrojový kód s názvem `PHPSESSID` a nikoliv všechny proměnné obsažené v relaci. Druhý způsob je možnost relace bez cookies, kdy předá PHP Session ID jako parametr při spuštění skriptu, avšak tato možnost musí být povolena v konfiguraci PHP, přitom se adresa URL rozšíří o Session ID například pro navigaci portálu:

```
http://localhost/SaR\_portal\_local/navigacePCZLIN.php?PHPSESSID=796d3fd5e16450a5024bfbc163911aa7
```

Obr. 11: Ukázka Session ID v URL

Z hlediska bezpečnosti je toto řešení hrozba:

- Pokud se pošle odkaz na Session ID e-mailem,
- Pokud se Session ID vyhledá prostřednictvím historie na veřejně přístupných PC
- Pokud je zařazena stránka do oblíbených Session ID je pořád stejné

Tato forma relace vyžaduje odpovídající způsob přeložení interpreterem PHP a při každém přechodu na novou stránku se musí správně přiřadit Session ID.

```
ini_set('session.use_trans_sid', 1); //povolení parametru session.use_trans_sid
session_start(); //spuštění relace
```

Obr. 12: Spuštění relace a správně přiřazení Session ID

PHP toto zařizuje pomocí modulu `url_rewriter`. V kódu musí být řešeno i přenos Session ID, PHP má proto klíčové slovo `SID`. Následující řádky zdrojového kódu ukazují, jak se dostávají informace ze `SID` do parametr `header`. Nejdřív je provedena inicializace proměnné `sid`. Pokud je `SID` prázdné, zůstává obsah zmíněné proměnné také prázdný. V opačném případě se před obsah `SID` umístí znak otazníku.

```
$sid = SID ? '?' . SID : '';
function baseurl() { // návrat http://<thiswebsite>/<thisdirectory>
    return "http://" . $_SERVER['HTTP_HOST'] .
        dirname($_SERVER['PHP_SELF']);
}
header("Location: " . baseurl() . "/navigacePCZLIN.php$sid");
```

Obr. 13: Realokace informací ze `SID` do parametr `header`.

Bohužel s PHP nezměníme samotnou podstatu protokolu HTTP, který pracuje se stavy. Po vyslání požadavku od klienta webový server tento požadavek vyřídí, další požadavek řeší nezávisle na předchozím. Zvýšení míry zabezpečení je možno dosáhnout sledováním IP-adresy klienta, který odesílá požadavky (`$_SERVER['REMOTE_IP']`) a sledování, že je stejný prohlížeč jako na začátku komunikace (`$_SERVER['USER_AGENT']`). V žádném případě se nejedná o ideální konečné řešení. Relace představují jednoduché a přitom silné řešení, jak trvale udržet data každého jednotlivého uživatele. Pokud nevytváříte z hlediska zabezpečení žádné kritické aplikace, ušetří Vám při sledování dat uživatele spoustu práce.

5.4.4 Unikátní ID relace

Hodnota unikátního klíče určujícího uživatelskou relaci je generována pomocí funkcí `UNIQID()` a `MT_RAND()`. Pro případ, že by ve stejný okamžik přišlo více různých dotazů, je navíc zkombinována s IP adresou klienta. Z výsledného řetězce je následně vytvořen hash pomocí `MD5()`:

```
&SESSID = MD5 (UNIQID (MT_RAND ())) . $_SERVER['REMOTE_ADR'];
```

Obr. 14: Unikátní ID relace

5.4.5 Zpracování dat z internetových formulářů

Nejčastěji objevené trhliny vycházejících z nedbale zpracovaných dat z internetových formulářů s využitím **SQL Injection** nebo **Cross Site Scripting (XSS)**. Bohužel ideální případ pro ochranu bylo nezobrazovat uživatelem zadané data, avšak v případě našeho portálu, kde jsou popisy je to nerealistické. Ztráta dat často vzniká drobnými problémy souvisejícími se znakovými sadami, důvodem je špatné zakódování. Záhy se nedá pak dodatečně zjistit, co vlastně uživatel do políček formuláře původně zadal. Pokud bude potřeba provádět ukládání do databáze, nebo zobrazovat data vyskytují se potíže se speciálními znaky HTML (&, <, >...). Samotný interpret PHP používá automatické ošetřování problematických znaků (*MagicQuotes*).

Data odeslané z formuláře se ukládají v našem případě do globální proměnné `&_POST`, protože jsme zvolili metodu odesílání dat POST. Metoda *POST* přenáší data z formuláře na webový server neviditelně. To je důvod proč jsme vybrali právě ji. Tato metoda je určena pro přenos dat, která se mají zpracovat pouze jednou, v případě překlepu a návratu zpět se musí data znovu zadat pokud se průběžně neukládají. Nasledující kód řeší načtení platných znaků:

```
function array_item($pole, $key) {  
    if(is_array($pole) && array_key_exists($key, $pole))  
        return($pole[$key]);  
    else  
        return FALSE;  
}  
  
$formdata = array_item($_POST, "form");  
if (is_array($formdata)) {  
    // vyhnutí se magic quotes  
    if (get_magic_quotes_gpc())  
        while ($i = each($formdata))  
            $formdata[$i[0]] = stripslashes($i[1]);  
}
```

Obr. 15: Řešení načtení platných znaků v kódu

6 WEBOBÉ ROZHRANÍ REKLAMAČNÍHO SYSTÉMU

6.1 Definice požadavků

Vytvořit zkušební verzi portálu (webových stránek) pro přístup zákazníků k základním informacím o reklamaci a servisu. Zadáání čísla reklamace a emailu provede vyhledání potřebných údajů. Tyto zkušební webové stránky budou sloužit jak pro dodavatele, tak pro odběratele.

Reklamační a servisní portály jsou důležitým prvkem prodejního procesu. Jelikož podnikatelský subjekt vystupuje jako koncový prodejce. V těchto částech řetězce můžou vzniknout vady, které bohužel zjistí až koncový zákazník. Na každý prodaný produkt kromě baterií se vztahuje záruka 2 roky v souladu s českou legislativou. Z důvodu zefektivnění komunikaci zákazníka s prodejcem, urychlení vyřízení reklamace, tvorby statistických přehledů vadnosti a poruchovosti výrobků vznikl reklamační portál. Jelikož reklamace úzce souvisí se servisem (záruční příp. pozáruční) byl portál rozšířen o evidenci servisních zakázek. Pro zákazníka to přináší výhodu přehlednosti a stavu reklamace produktu, navíc vidí detailní stav a rozbor, problému.

V tomto portálu pro reklamaci a servis mohou všichni přihlášení uživatelé číst vlastní reklamace, komentovat stavy, odpovídat na dotazy servisních techniků, popřípadě přidávat vlastní názory na použité produkty.

Portál pro reklamace a servis bude mít tyto vlastnosti:

- správa přihlašování uživatelů s kontrolou pomoci e-mailové adresy (noví uživatelé se stanou aktivními až po potvrzení odeslaném e-mailové schránky nového uživatele).
- automatické přihlašování uživatelů, kteří povolí používání *cookies*
- zobrazení seznamu vláken pomocí odsazování
- rozvržení vyhledávání na stránce a formátování pomocí CSS
- vyhledávání vlastních reklamací

6.1.1 Navigace portálu pro Reklamace a Servis - Počítače Zlín

Úvodní stránka portálu *navigacePCZLIN.php* - v hlavičce naleznete logo firmy Počítače Zlín s odkazem na úvodní stránku, nadpis s jednoduchými a demonstrativními ikonami. V pravé části je blok s možností přihlášení případně registrace nového uživatele. Pokud chce

uživatel přispívat do portálu a komentovat reklamace vlastními příspěvky, musí se přihlásit. V těle ukazuje zákazníkovi i technikovi celý reklamační a servisní proces, kterým se každá reklamace či servis řídí. Dále portál dovoluje vybrat, jestli chce návštěvník vstoupit jako odběratel a prohlížet reklamace nebo jako Administrátor a tvořit nové zakázky, dělat podporu pro zákazníky a řídit reklamační procesy, aktualizovat stavy a v neposlední řadě zpracovávat statistiky vyřízených reklamací. V patičce je Copyright firmy s odkazem na kořen <http://pczlin.cz>.

Každý zákazník může prohlížet reklamační a servisní zakázky k aktuálnímu datu a také přirozeně formulář pro vyhledávání. Reklamační protokoly jsou defaultně řazeny podle nejnovějšího příspěvku, případně posledního komentáře k příspěvku. Číslo v závorce udává počet příspěvků v protokolu. Pokud je počet protokolů vyšší než hodnota `pagemax` (ta je nastavena v souboru `limity.php`, standardní hodnota je 10), objeví se pod seznamem zakázek odkazy na další stránky. Na tomto místě se zároveň i objeví i odkaz na NOVÝ PROTOKOL pro vytvoření nové reklamační či servisní zakázky.

Texty jednotlivých příspěvků se postupně tak, jak přicházejí, zároveň i odsazují, takže je na první pohled jasné, kdo komu odpověděl. Po klepnutí na odkaz ODPOVĚĎ můžete velmi snadno přidat do protokolu příspěvek. Pokud existuje více odpovědí, pak se tyto třídí podle času, kdy přišli. Jako první se tedy zobrazí první přijata odpověď.

6.1.2 Přidávání nových komentářů k reklamacím a servisu

Klepnutí na odkaz NOVÝ KOMENTÁŘ pod seznamem zakázek, popřípadě klepnutím na odkaz ODPOVĚĎ pod každým příspěvkem, se dostanete na stránku `newcomment.php`, kde sami můžete napsat komentář. Pokud ještě nejste přihlášení, pak se namísto formuláře pro zadání dotazu objeví formulář pro přihlášení.

6.1.3 Správa přihlašování

Pokud se nepřihlásíte, pak se ke stránce pro přihlášení dostanete automaticky, pokud budete chtít přidat protokol, nebo komentář. Přihlásit se do fóra můžete prostřednictvím odkazu PŘIHLÁSIT SE v hlavičce. Pokud se uživatel přihlásí, změní se text odkazu na ODHLÁSIT SE.

Již registrovaným uživatelům stačí na stránce pro přihlášení zadat pouze uživatelské jméno a heslo. Pokud má uživatel na svém počítači povoleno přijímání souborů `cookies`, pak zůstane přihlášen až na 30 dnů, samozřejmě pokud se sám klepnutím na odkaz ODHLÁSIT

SE neodhlásí. Je to sice velmi pohodlné, ale při používání zejména na veřejně přístupných počítačích zrovna příliš bezpečný není. Dobu platnosti souborů *cookies* lze změnit v souborech *prihlaseni.php* a *authent.php*.)

Nový uživatel musí zadat uživatelské jméno, dvakrát za sebou stejné heslo a také platnou e-mailovou adresu. Pokud zadaná hesla souhlasí a uživatelské jméno ještě není obsazeno, odešle se prostřednictvím e-mailu odkaz pro ověření, registrace. Tento odkaz může vypadat kupříkladu takto:

```
http://localhost/testik/PHP/authent.php?userID'985&authent=151245315
```

Obr. 16: Odkaz pro ověření, registrace.

Uživatel se bude moci přihlásit až po klepnutí na tento odkaz. Pokud aktivace neproběhne do 24 hodin, uživatelem zadaná data se opět odstraní. Tímto způsobem je zajištěno, že je zadaní e-mailová adresa platná alespoň v době přihlášení a že k ní má uživatel přístup. Daná e-mailová adresa se používá v případě jakékoliv změny ve vytvořené protokolu, nebo komentovanému protokolu je uživatel automaticky informován elektronickou správou.

6.1.4 Prohledávání a filtrace v portálu

Funkci pro vyhledávání nejspíše použijete, když v záhlaví stránky pro zákazníky zadáte ID číslo nebo e-mail a poté stisknete tlačítko *HLEDAT*. V internetovém prohlížeči se zobrazí stránka *search.php*, která bude obsahovat podrobný formulář pro vyhledávání a taky výsledky posledního hledání. Před vyhledanými odkazy se vždy zobrazí číslo, které udává úspěšnost vyhledání. Na stránkách s výsledky jsou výrazy, které se vyhledávaly barevně zvýrazněny.

Možnosti vyhledávání je možné rozšířit pomocí znaku hvězdička. Výchozí vyhledávání prochází výrazy obsahující všechna slova, která jste zadali. V případě, že znáte začátek hledaného řetězce, použijte hvězdičku (*ASUS.php.php* tedy bude hledat všechny produkty značky ASUS bez ohledu na modelovém a sériovém označení). Jiné speciální znaky použít nelze. V případě zadání budou automaticky odstraněny před zahájením vyhledávání.

6.2 Struktura zdrojového kódu

Tab. 7: Struktura zdrojového kódu

<i>navigacePCZLIN.php</i>	Zobrazuje úvodní stránku portálu pro reklamace a servis
<i>pro_zamestnance.php</i>	
<i>pro_zakazniky.php</i>	
<i>login.php</i>	Přihlášení
<i>authent.php</i>	Potvrzení přihlášení (ověření e-mailové adresy)
<i>logout.php</i>	Odhlášení
<i>portal-function.php</i>	Různé pomocné funkce
<i>mjoeDb.php</i>	Třída MojeDb pro přístup k MySQL
<i>password.php</i>	Přihlašovací údaje pro <i>mjoeDb.php</i>
<i>formfunctions.php</i>	Funkce pro zobrazení formuláře
<i>myPortal.css</i>	Soubor CSS pro formátování tabulky s výsledky

6.2.1 Přístup k databázi, vytvoření formuláře

Všechny přístupy k databázi se provádí přes objekt třídy *mjoeDb*, která zabraňuje výskytu nadbytečného zdrojového kódu a je velmi užitečná i při vyhledávání chyb. Všechny soubory PHP práce začínají tím, že nejprve načítají soubor *mjoeDb.php*, pak vytvoří objekt *mjoeDb* a uloží do proměnné *databaze*. V případě chyby, vytvoří funkce konstruktoru *mjoeDb* kompletní stránku ve formátu HTML – tato stránku zobrazí chybové hlášení a ukončí činnost programu.

Stejně tak neefektivní a v případě budoucí perspektivy rozšiřování portálu je výhodné pro vytváření formulářů i pro soubory *login.php* vytvořit soubor s pomocnými funkcemi *portal-function.php* a funkce pro zobrazení formuláře *formfunctions.php*, které ušetří spoustu času.

Nejdůležitější pomocní funkce pro portál *portal-function.php*:

<i>array_item(\$array, \$key)</i>	Zobrazuje úvodní stránku portálu pro reklamace a servis
<i>build_href(\$url, \$query, \$txt)</i>	
<i>baseurl()</i>	

<code>html_start(\$title, \$css)</code>	Přihlášení
<code>html_end()</code>	Potvrzení přihlášení (ověření e-mailové adresy)
<code>get_user_id(\$userID, \$username, \$database)</code>	Odhlášení
<code>Show_menu(\$username, \$reklamaceID, \$nazevProdukt_Sluzba, \$searchform)</code>	Různé pomocné funkce

Tab. 8: Pomocní funkce ve skriptu `portal-function.php`

6.3 Správa přihlašování

Správa přihlašování patří ke standardním operacím velkého množství webových portálů. To je důvod pro zvýšenou pozornost z pohledu portálu. V Portál pro Reklamace a Servis - Počítače Zlín musíme při prvním přihlašování zadávat e-mailovou adresu. Na tuto adresu se záhy pošle odkaz, kterým uživatel přihlášení do systému dokončí. Přihlášení se ukládá nadvakrát, jednou jako *relace (session)* a jednou jako *cookie* :

Proměnné relace s názvy `$_SESSION['jmeno']` a `$_SESSION['adminID']` zajišťují, že se přihlašovací údaje zadané uživatelem neztratí ani po přesunu na jinou stránku aplikace. Tato možnost bude fungovat i tehdy, když uživatel nebude mít na svém počítači povolené přijímání souborů cookies.

Proměnné cookies s názvy `$_SESSION['portal_adminID']` a `$_COOKIE['portal_Authent']` slouží k pohodlnému ovládní aplikace a při dalším návštěvě stránky uživatelem automaticky přihlásí.

Pokud uživatel nepovolí ukládání souborů cookies na svém počítači, automatické přihlašování samozřejmě fungovat nebude. Správa relace je ale naprogramována tak, aby fungovala i bez souborů cookies (`ini_set('session.use_trans_sid',1)`). Bohužel je tam i nevýhoda, že ID bude součástí relace, čili může se vyskytnout mezi oblíbenými odkazy.

Abychom zabránili pokusům o zneužití pomocí fungovaných souborů cookies, neobsahuje soubor cookie uživatelské jméno přímo, ale obsahuje pouze adminID a poté náhodné číslo pro ověření. Pomocí dotazu na databázi z něj můžeme zjistit uživatelské jméno, přičemž zároveň probíhá kontrola, zda jsou v souboru cookie uloženy správné údaje.

6.4 Výběr skriptovacího jazyka

Skriptovací jazyky pro dynamický obsah jsou ASP a PHP. Skriptovací jazyk ASP je spojen s technologií Microsoftu (webový Windows server). Skriptovací jazyk PHP je v tomto ohledu univerzálnější. Je lehké ho nainstalovat jak na webovém serveru na platformě Microsoft, Mac OS X nebo Linux serveru, co bývá i nejčastější řešení v kombinaci s MySQL databází. Vzhledem k výběru databázového serveru MySQL je vhodné volit jako skriptovací jazyk PHP.

6.5 Tvorba SQL dotazů

Pro výběr požadovaných údajů z databáze byly vytvořeny následující dotazy:

- Informace o reklamaci:

```
SELECT
    `nazevProdukt_Sluzba`,DATE(`datumReklamace`), `duvodReklamace`,
    `nazevStavu`, `vyjadreniReferentaReklamace`
FROM `reklamace`
INNER JOIN `stavreklamace` ON
`reklamace`.`stavReklamaceID`=`stavreklamace`.`stavReklamaceID`
INNER JOIN `odberatel` ON `reklamace`.`odberatelID`=`odberatel`.`odberatelID`
```

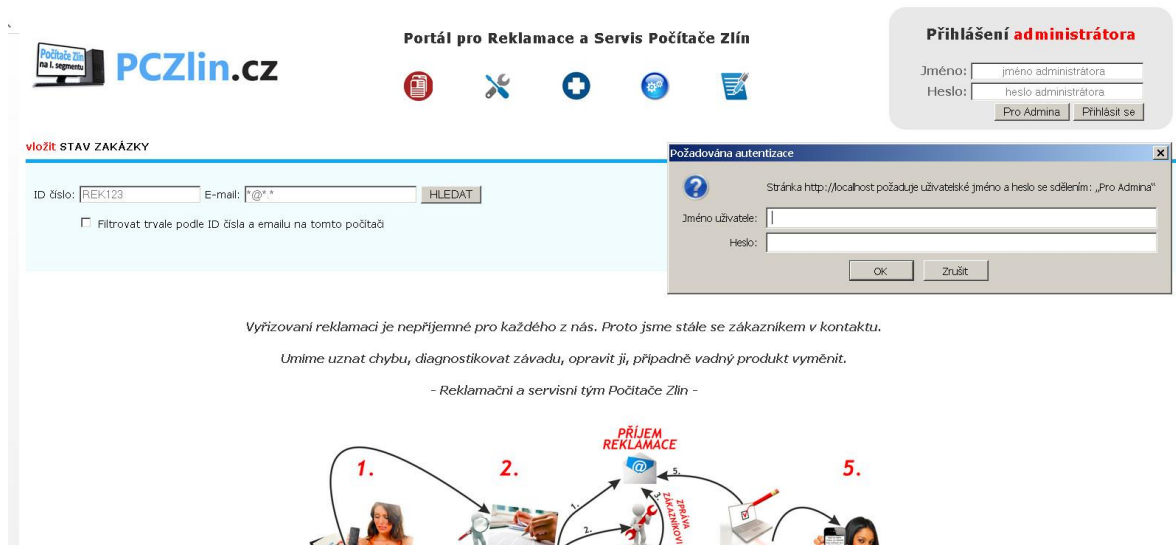
Tab. 9: Ukázka tvorby jednoho SQL dotazů pro výběr údajů z databáze

Vstupními údaji do těchto dotazů bude číslo reklamace a zároveň registrační email zákazníka.

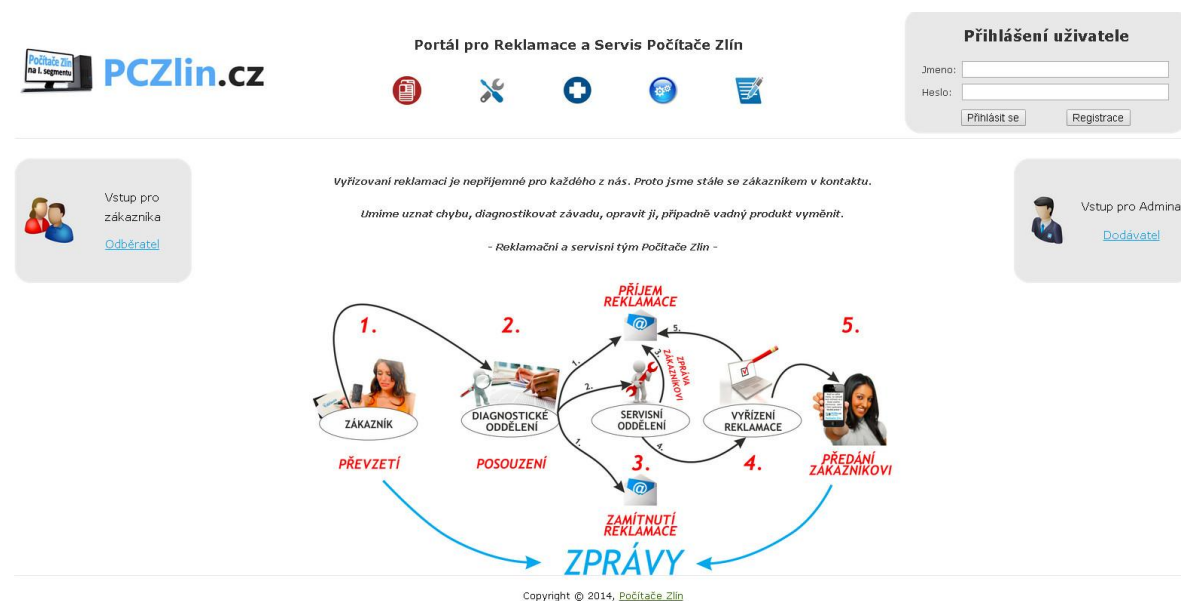
6.6 Tvorba PHP skriptů a úvodního formuláře

Formulář pro zadávání vstupních dat a spuštění dotazu je vytvořen ve statickém html jazyce, využívá syntaxe ovládacího prvku input:

Obr. 17: Ukázka formulářů pro zadávání vstupních dat, po přihlášení



Obr. 18: Ukázka řešení autentizace přes Apache, .htaccess a .htpasswd



Obr. 19: Ukázka navigace portál u pro reklamaci a servis.

Dynamické stránky *reklamace_odberatel.php*, resp. *reklamace_dodavatel.php* obsahují kód pro připojení k databázi, načtení informací SQL dotazem a výpisu získaných informací do prohlížeče.

7 VYLEPŠENÍ SYSTÉMU A BUDOUCÍ POKRAČOVÁNÍ PRÁCE

Tato práce svým rozsahem plně pokryla základní cíle, které byly v úvodu stanoveny pro vytvoření portálu, který bude zpracovat reklamační a servisní zakázky. Jelikož se jedná o webovou prezentaci, možnost vylepšení je v podstatě do nekonečna, vždy je co ladit a vylepšovat. Dále je možné navrhnout aplikaci pro Windows, která by připravila data offline a v případě připojení na internet se uploadují do databáze. Dále co se týče zdrojového kódu, je možnost v budoucnu použít PHP Framework, který často eliminuje bezpečnostní rizika (XSS, CSRF, session hijacking, session fixation), zabezpečuje znouvoužitelnost, většina již podporuje moderní technologie a koncepce (AJAX/AJAJ, Dependency injection, SEO, DRY, , KISS, MVC, Web 2.0) navíc často poskytuje i ladící nástroje, které rychle odhalí všechny chyby aplikace a navíc Framework zabezpečí snadnou rozšiřitelnost.

7.1 Benefity navrženého systému v porovnání s papírovou reklamací

Daný portál má benefity jak pro dodavatele (servisní technik, vyřizovatel reklamace) tak pro odběratele (zákazníky). Na jedné straně zákazníkovi dává možnost mít přehled o svých reklamacích, servisních opravách, pokud zákazník není přímo z okolí má možnost prostřednictvím internetu se ptát na aktuální stav, podělit se o své poznatky o vadě a ušetřit tak čas na opravě a snížit tak celkovou cenu v případě pozáruční opravy. Dále zákazník je neprodleně informován o zásadních změnách, má archivován celou reklamaci případně servis. V neposlední řadě je obrovská ekologická co je dnes velmi populární se šetří papír (lesy), co v konečném důsledku snižuje celkovou cenu. Na druhé straně dodavatelovi dává silný nástroj pro logistiku reklamace, kdy vědění a kontrolér jakosti vidí, co bylo zanedbáno, kde nejčastěji vznikají vady. Toto poskytuje možnost statistického vyhodnocení chybovosti, produktu, případně pokud se jedná o reklamaci služby je jasně popsán problém a kdo nese za to zodpovědnost. Portál spojuje koncového prodejce, servisní středisko a zákazníka, pomocí svých databázových relací a roztříděných dat.

7.2 Zabezpečení systému

Stávající zabezpečení je pomocí správy uživatelů, co již bylo popsáno. Co je však možné vylepšit pro zvýšení bezpečnosti, je možné ještě přidat při registraci nového uživatele Captchu, pro zamezení vytváření "fake" uživatelů pomocí robotů. Dále je možno přidat tlačítko pro zapomenuté heslo, aby si uživatel mohl obnovit svoje heslo. Pro administrátora

by mohlo být zajímavé přidělat speciální menu, pro správu uživatelských účtů, přímo z webového sídla.

7.3 Budoucí pokračování práce

Model portálu pro servis a reklamace byl programován z tzv. “prázdného listu“, které bylo sice z počátku rychlé, později však takto napsanou webovou aplikaci je velmi obtížné jakkoliv rozšiřovat. Proto v budoucí práci se zaměřím na použití vhodného Framework a budu se zabývat jen podstatnou logikou aplikace a ne vším tím nudným psaním okolo. V budoucnu, je možné přidělat modul, který by se staral o doručení reklamované případně servisovaného produktu, byl by napojen na českou poštu a zákazník, by měl přehled i o poloze zásilky. Navíc je možno dodělat pro zákazníka přehledné grafické rozhraní, které mu zobrazí aktuální stav vyřizované závady. Nicméně tento portál pro reklamaci a servis bude přemístěn do podkategorie, hlavní webové stránky s e-shopem a bude tam odkaz na navigaci portálu. Je možné ještě optimalizovat SEO a SEM pro pozáruční servis pro zákazníky. Pak bude možné sjednotit přihlášení uživatele z e-shopu s portálem. V případě výměny komponentů, bude možné zařadit použité repasované komponenty, do prodeje v kategoriích opravené a použité zboží bez záruky. Tyto cíle a mnoho dalších otázek si budu klást v diplomové práci, ve které bych chtěl řešit i responzivní design stránky.

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout systém pro reklamaci a servis zboží, který byl postaven na relační databázi InnoDB v prostředí server MySQL v.5. Pro přístup zákazníků, servisních techniků a dodavatele bylo navrženo a naprogramováno nové dynamické webové rozhraní. K databázi lze přistupovat díky vybraným údajům prostřednictvím internetového prohlížeče pomocí aktivních stránek ve skriptovacím jazyku PHP (jako způsob komunikace s odběratelem). Dále byla detailně popsána platforma pro běh systému.

V teoretické části jsem popsal základní terminologii a principy tvorby dynamických webových stránek a přitom využil efektivní produktivní nástroje, které se nejčastěji používají. Dále jsem se okrajově zmínil o současném legislativním rámci na území České republiky pro reklamace, na základě něhož byl předběžně navrhnout reklamační systém. Celý proces byl rozdělen do několika kroků. Od převzetí reklamovaného zboží až po zpětné předání zákazníkovi. Reklamační systém nyní funguje v podnikatelském subjektu Počítače Zlín – Martin Chumchál. Do té doby se používalo pouze manuálního vypisování tištěných reklamačních protokolů, avšak bez jakékoliv logistiky. Toto byl hlavní důvod navržení aplikace webového systému do provozu. Dále v teoretické části byla rozebrána problematika datového modelu, normalizace a obecně proces návrhu a implementace databáze. V závěru teoretické části jsem popsal komunikaci se zákazníkem prostřednictvím daného systému a poukázal jsem na ochranu osobních údajů související se zpracováním osobních údajů dle zákona 101/2000 Sb. ve znění pozdějších předpisů. Teoretická část je uzavřena softwarovými nástroji pro offline práci (WAMP, PhpMyAdmin, Netbeans, Total Commander), které jsem využíval při programování a testování na serveru.

V praktické části jsem se zabýval samotným návrhem systému na základě parametrů, které by měl splňovat. Bylo potřeba vytvořit podrobné E/R schéma aby byly detailně rozepsány tabulky, zkontrolována normalizace a tím zabezpečená datová integrita se kterou souvisí bezpečnost uživatelů. Otázkou bezpečnosti jsem se hlavně zabýval z pohledu potvrzování uživatelských registrací přes email. Dále správou relací, samotným přihlašování a zpracováním dat z internetových formulářů. Následně jsem stručně popsal strukturu kódu a uvedl příklad tvorby SQL dotazů pro výběr informací v konkrétní reklamaci. Následně bylo provedeno testování fiktivními reklamacemi jako v reálném

provozu, aby byl systém zhodnocen. Tím přinesl možnost statistického vyhodnocení chybovosti. Možnost lépe řídit reklamační procesy a aktualizovat jejich stavy.

V závěru práce byly shrnuty benefity a naznačeno další pokračování své dosavadní práce.

ZÁVĚR V ANGLIČTINĚ

The aim of this thesis was to design a system for complaints and service catalog, which was built on a relational database server environment InnoDB in MySQL v.5. To access customers, service technicians and contractors have designed and programmed a new dynamic web interface. The database can be accessed through the selected data via a web browser using the active sites in the scripting language PHP (as a way of communication with the customer). Was also described in detail platform for running the system.

The theoretical part describes the basic terminology and principles of creating dynamic web pages and used while effective productivity tools that are most commonly used. Furthermore, I mentioned in passing the current legislative framework in the Czech Republic for the claim, which was based on pre-designed complaint system. The whole process was divided into several steps. Since taking over the goods claimed to feedback to the customer. Complaints system now works in the corporate entity Computers Zlín - Martin Chumchal. Until then, use only the printed manual logging of complaints protocols, but without any logistics. This was the main reason to design a web application system into operation. Furthermore, the theoretical part was dealt with the issue of data model, normalization, and generally the design and implementation of the database. At the end of the theoretical part I described communication with the customer through the system and I pointed to the protection of personal data related to the processing of personal data pursuant to Act 101/2000 Sb. as amended. The theoretical part is enclosed software tools for offline work (WAMP, PhpMyAdmin, Netbeans, Total Commander) used for programming.

In the practical part I dealt Samotnia system design based on parameters that should meet. It was necessary to create detailed E / R diagram that detail how the table, checked standardization and thus secure the integrity of the data that relates to the safety of users. The question of safety, I mainly dealt with from the perspective of user registration and validation via email. Furthermore, session management, logging and processing of the actual data from web forms. Then I briefly described the structure of the code, and illustrated the creation of SQL queries to select specific information in the complaint. Subsequently, testing was carried out fictitious complaint as in real traffic that the system is evaluated. This brought a statistical evaluation of errors. The ability to better manage the complaint process and update their status. In conclusion, the thesis summarizes the benefits and suggested the continuation of his previous work.

SEZNAM POUŽITÉ LITERATURY

- [1] PROCHÁZKA, David. *PHP 6: začínáme programovat. 1. vyd.* Praha: Grada, 2012, 183 s. Průvodce (Grada). ISBN 978-80-247-3899-4,
- [2] PONKRÁC, Miloslav. *PHP a MySQL: bez předchozích znalostí : [průvodce pro samouky]*. Vyd. 1. Brno: Computer Press, 2007, 221 s. Průvodce (Grada). ISBN 978-80-251-1758-3,
- [3] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007, 607 s. Průvodce (Grada). ISBN 978-80-251-1813-9,
- [4] MACH, Jakub a Bernd ÖGGL. *PHP pro úplně začátečníky.: průvodce webového programátora. 2., přeprac. a rozš. vyd.* Brno: Computer Press, 2003, 167 s. Průvodce (Grada). ISBN 80-722-6834-1,
- [5] VRÁNA, Jakub a Bernd ÖGGL. *1001 tipů a triků pro PHP: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2010, 456 s. Průvodce (Grada). ISBN 978-80-251-2940-1,
- [6] PEACOCK, Michael a Bernd ÖGGL. *Programujeme vlastní sociální síť v PHP 5: průvodce webového programátora. 1. vyd.* Brno: Computer Press, 2012, 424 s. Průvodce (Grada). ISBN 978-80-251-3626-3,
- [7] RIORDAN, R. M.. *Designing Relational Database Systems*. Microsoft Press, 1999. ISBN 978-0735606340
- [8] *Jak psát web: HTML, JS, CSS ...* [online]. [cit. 2009-04-10]. Dostupný z WWW: <<http://www.jakpsatweb.cz/>>.
- [9] Castagnetto, J., Rawat, H., Schumann, S., Scyllo, C., Veliath, D. *Programujeme PHP profesionálně. 2. vyd.* Brno, Computer Press, 2004. 656 s. ISBN 80-7226-310-2.
- [10] *PHP: Hypertext Preprocessor*. [online]. [cit. 2014-06-01]. Dostupné z: <http://cz.php.net>.
- [11] Williams, H. E., Lane, D. *PHP a MySQL: Vytváříme webové databázové aplikace. 1. vyd.* Praha, Computer Press, 2002. 530 s. ISBN 80-7226-760-4.
- [12] Lacko, L. *PHP 5 a MySQL 5: Hotová řešení. 1. vyd.* Brno, Computer Press, 2007. 316 s. ISBN 978-80-251-1695-1.
- [13] NENADÁL, J., NOSKIEVICOVÁ, D. *Moderní management jakosti. Praha: Management press, 2008. 377 s. ISBN 978-80-7261-186-7.*

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

SQL	Structured Query Language,
DS	Databázový systém,
WAMP	Windows, Apache, MySQL, and PHP,
IT	Informační technologie,
BDE	Borland Database Engine,
DAO	Data access object,
ADO	ActiveX Data Objects,
IN	Input (vstup),
OUT	Output (výstup),
ČR	Česká republika,
SEM	Search Engine Marketing,
SEO	Search Engine Optimization,
ASCII	American Standard Code for Information Interchange
DBMS	DataBase Management System - systém řízení báze dat
CSS	Kaskádové styly – nadstavba HTML rozšiřující jeho designové možnosti
DCL	Data Control Language, skupina příkazů jazyka SQL pro běh a údržbu databáze
DDL	Data Definition Language, skupina příkazů jazyka SQL pro definice objektů
DML	Data Manipulation Language, skupina příkazů jazyka SQL pro práci s daty
DB	Databáze
BP	Bakalářská práce
Esc	Escape (ESC), klávesa pro únik, ukončení činnosti, pravý horní roh klávesnice
FTP	File Transfer Protocol, internetový protokol určený pro přenos souborů
GUI	Graphics User Interface, uživatelsky příjemně graficky zpracované programy
HTML	programovací jazyk pro tvorbu webových stránek – nástupce XHTML

- PHP PHP Hypertext Preprocessor – technologie pro tvorbu webových stránek
- UTC Universal Time Coordinated (Coordinated UT), časová pásma
- NF normální forma

SEZNAM OBRÁZKŮ

<i>Obr. 1: Tvorba relačních databází</i>	12
<i>Obr. 2: Rozdělení SQL dotazů do podmnožin.....</i>	14
<i>Obr. 3: Průběh vyřizování reklamace a servisu firmou Počítače Zlín</i>	19
<i>Obr. 4: E/R diagram podle Chena diagram entit a relací části portálu pro reklamaci</i>	21
<i>Obr. 5: Ukázka nástroje phpMyAdmin na správu administraci databáze přes Web</i>	28
<i>Obr. 6: Vývojové prostředí NetBeans s PHP a jiných webových technologií</i>	29
<i>Obr. 7: Ukázka práce s FTP managerem v Total Commanderu</i>	29
<i>Obr. 8: Schéma servisní a reklamační systému</i>	32
<i>Obr. 9: Ošetření formulářových vstupů</i>	36
<i>Obr. 10: Potvrzování uživatelských akcí přes email.....</i>	36
<i>Obr. 11: Ukázka Session ID v URL</i>	37
<i>Obr. 12: Spuštění relace a správně přiřazení Session ID</i>	38
<i>Obr. 13: Realokace informací ze SID do parametr header.</i>	38
<i>Obr. 14: Unikátní ID relace</i>	39
<i>Obr. 15: Řešení načtení platných znaků v kódu.....</i>	39
<i>Obr. 16: Odkaz pro ověření, registrace.....</i>	42
<i>Obr. 17: Ukázka formulářů pro zadávání vstupních dat, po přihlášení.....</i>	45
<i>Obr. 18: Ukázka řešení autentizace přes Apache, .htaccess a .htpasswd</i>	46
<i>Obr. 19: Ukázka navigace portál u pro reklamaci a servis.....</i>	46

SEZNAM TABULEK

<i>Tab. 1: Relace dodavatelServis ne v INF</i>	22
<i>Tab. 2: Relace zamestnanecServis</i>	22
<i>Tab. 3: Relace dodavatelServis v 2. NF</i>	23
<i>Tab. 4: Relace stavServis</i>	23
<i>Tab. 5: Profily uživatelů s oprávněním přístupu</i>	31
<i>Tab. 6: Tabulka uživatelů, která zabezpečuje správu portálu a rozdělení práv</i>	36
<i>Tab. 7: Struktura zdrojového kódu</i>	43
<i>Tab. 8: Pomocní funkce ve skriptu portal-function.php</i>	44
<i>Tab. 9: Ukázka tvorby jednoho SQL dotazů pro výběr údajů z databáze</i>	45

