

# **Evidence elektronické ošetrovatelské dokumentace pro pečovatelské domy**

Records of Electronic Nursing Documentation for Nursing Homes

Bc. Jan Pich

---

Diplomová práce  
2012



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2011/2012

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan PICH**  
Osobní číslo: **A10331**  
Studijní program: **N 3902 Inženýrská informatika**  
Studijní obor: **Bezpečnostní technologie, systémy a management**

Téma práce: **Evidence elektronické ošetrovatelské dokumentace pro pečovatelské domy**

Zásady pro vypracování:

1. Zmapujte existující systémy, které je možno implementovat v prostředí domovů pro seniory a podobných zařízeních. Vytvořte návrh vlastního řešení.
2. Vytvořte zprávu elektronické ošetrovatelské dokumentace pro domy pečující o seniory.
3. Vytvořte potřebné typy uživatelských účtů s požadovanými oprávněními.
4. Vytvořte pro každého klienta jeho kartu s ošetrovatelskou dokumentací.
5. Vytvořte a implementujte programové rozhraní pro správu této dokumentace na základě přidělených uživatelských práv a celý systém implementujte v praxi.

Rozsah diplomové práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **KOFLER, Michael a Bernd ÖGGL. PHP 5 a MySQL 5: průvodce webového programátora. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 978-802-5118-139.**
2. **LAVIN, Peter. PHP – objektivě orientované: koncepty, techniky a kód. 1. vyd. Praha: Grada, 2009, 211 s. Průvodce (Grada). ISBN 978-802-4721-378.**
3. **MySQL profesionálně: optimalizace pro vysoký výkon. Vyd. 1. Brno: Zoner Press, 2009, 712 s. ISBN 978-807-4130-359.**
4. **SCHLOSSNAGLE, George. Pokročilé programování v PHP 5. Brno: Zoner Press, 2004, 640 s. ISBN 80-868-1514-5.**
5. **RAK, Michal. Ošetrovatelská dokumentace. České Budějovice, 2009. Diplomová. Jihočeská univerzita. Vedoucí práce Mgr. Helena Michálková.**

Vedoucí diplomové práce:

**Ing. Dalibor Slovák**

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

**24. února 2012**

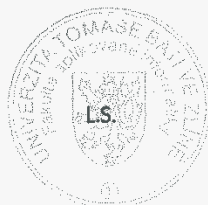
Termín odevzdání diplomové práce:

**15. května 2012**

Ve Zlíně dne 24. února 2012

prof. Ing. Vladimír Vašek, CSc.

*ka*  
děkan



doc. RNDr. Vojtěch Křesálek, CSc.

*JK*  
ředitel ústavu

## ABSTRAKT

Ošetřovatelská dokumentace se stává nedílnou součástí naší zdravotní dokumentace. Je potřeba na ní nahlížet nejen jako na administrativu, ale jako na soubor důležitých informací týkající se našeho celkového stavu, protože to může být právě ona, která dá zdravotníkům správný klíč pro záchranu našeho života. Obsahy těchto dokumentací se drobně liší dle vedení v jednotlivých zařízeních, ale jejich společný účel plní všechny – orientace pro zdravotníky o naší osobě ze zdravotnického hlediska. Úkolem diplomové práce bude zmapovat tuto oblast v době, kdy do ní přichází v plné síle informační technologie a účast fyzických či právnických osob ve vlastnictví těchto zařízeních je čím dál tím větší. Práce by měla odpovědět na otázku – „Jaké mají tyto trendy dopad na bezpečnost osob při stávající legislativě?“. Pro účely diplomové práce jsem se zaměřil na bezpečnost klientů využívající služeb domů pečující o seniory.

Klíčová slova:

Ošetřovatelská dokumentace, aplikace, informační technologie, php.

## **ABSTRACT**

Nursing documentation is an integral part of our medical records. You need only look at it as an administration, but as a set of important information regarding our general condition, because it may just be she who gives medical professionals the right key to save our lives. The contents of these documents are slightly different according to the guidance in individual facilities, but their common purpose of fulfilling all - focus on our health for a person from a health standpoint. The task of the thesis is to map this area when it comes into full force in information technology and the participation of natural or legal persons in these facilities is becoming more and more. The work should answer the question - "What are these trends impact on the safety of persons in the existing legislation?". For the purpose of the thesis I focused on the safety of clients using the services of home care for seniors.

Keywords:

Nursing documentation, applications, information technology, php.

Poděkování, motto

Za odborné konzultace děkuji paní Bc. Janě Řehákové, slečně Michaele Dolečkové a slečně Ivě Bambulové.

Za uvedení do problematiky děkuji své matce, paní Petře Pichové.

Za vedení práce a patří velké poděkování panu Ing. Daliboru Slovákovi.

**Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/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 diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/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 diplomovou/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 – diplomovou/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 diplomové/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 diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/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 diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....  
podpis diplomanta

**OBSAH**

<b>ÚVOD</b> .....	<b>11</b>
<b>I TEORETICKÁ ČÁST</b> .....	<b>12</b>
<b>1 DOKUMENTACE</b> .....	<b>13</b>
1.1 ZDRAVOTNÍ DOKUMENTACE.....	13
1.2 OŠETŘOVATELSKÁ DOKUMENTACE.....	15
1.2.1 Základní složky.....	16
1.2.2 Základní principy vedení.....	17
1.2.3 Právní úprava.....	18
1.3 SOCIÁLNÍ DOKUMENTACE.....	18
1.4 NAHLÍŽENÍ DO DOKUMENTACE.....	18
1.4.1 Klient.....	18
1.4.2 Lékař, vrchní sestra, staniční sestra.....	18
1.4.3 Pracovníci přímé obslužné péče.....	18
<b>2 ELEKTRONICKÁ DOKUMENTACE</b> .....	<b>19</b>
2.1 ZÁKLADNÍ LEGISLATIVA.....	20
2.2 ZÁKLADNÍ OCHRANA DAT.....	20
2.3 INFORMAČNÍ SYSTÉM CYGNUS.....	20
2.3.1 Moduly.....	21
2.3.2 Sociální část.....	21
2.3.3 Dokumentace klienta.....	23
2.3.4 Vykazování na ZP.....	25
2.3.5 Stravovací část.....	25
2.3.6 Sklady.....	26
2.3.7 Zaměstnanci.....	27
2.3.8 Majetek.....	29
2.3.9 Manažerská část.....	29
2.4 FONS ENTERPRISE - NEMOCNIČNÍ INFORMAČNÍ SYSTÉM.....	30
2.5 SHRUTÍ.....	33
2.6 VLASTNÍ NÁVRH.....	35
<b>II PRAKTICKÁ ČÁST</b> .....	<b>40</b>
<b>3 TECHNICKÁ DOKUMENTACE APLIKACE</b> .....	<b>41</b>
3.1 VÝVOJOVÉ PROSTŘEDKY.....	41
3.2 PHP.....	41
3.3 DATABÁZE.....	41
3.3.1 Struktura databáze.....	41
3.4 KOŘENOVÝ ADRESÁŘ.....	45
3.5 INDEX.PHP.....	45
3.6 FUNKC.PHP.....	46
3.7 NAVIGACE.PHP.....	48
3.8 SPRÁVA UŽIVATELŮ.....	49
3.8.1 Nový uživatel.....	49
3.8.2 Smazání uživatele.....	51



3.8.3	Editace uživatele .....	52
3.8.4	Uzivatele.php .....	52
3.8.5	Zpracovaniuuz.php.....	53
3.9	HISTORIE.....	54
3.9.1	Způsob ukládání historie při běhu aplikace .....	54
3.9.2	Historie uživatelů aplikace .....	55
3.9.3	Historie klientů.....	56
3.9.4	Historie léky .....	56
3.10	NÁVŠTĚVNÍ KNIHA .....	56
3.10.1	Vstup návštěvy .....	57
3.10.2	Odchod návštěvy .....	58
3.10.3	Aktivní návštěvy .....	58
3.10.4	Proběhlé návštěvy .....	59
3.11	KLIENSKÁ ČÁST APLIKACE .....	60
3.11.1	Nový klient.....	60
3.11.2	Editace klienta.....	61
3.11.3	Zpracovanikl.php .....	62
3.11.4	Podpůrné soubory pro editaci jednotlivých částí klientské dokumentace .....	63
3.11.5	nemoci.php .....	64
3.11.6	Zobrazení klientské karty .....	66
3.11.7	Zpracovanizob.php.....	67
3.11.8	Karta.php .....	69
<b>4</b>	<b>UŽIVATELSKÝ MANUÁL K APLIKACI.....</b>	<b>71</b>
4.1	PŘÍSTUP K APLIKACI.....	71
4.2	PŘIHLÁŠENÍ .....	71
4.3	ZÁKLADNÍ VZHLED.....	72
4.4	MENU .....	73
4.5	AKTIVNÍ OKNO .....	74
4.6	JEDNOTLIVÉ FUNKCE .....	74
4.6.1	Správa uživatelů .....	74
4.6.2	Upravit uživatele .....	75
4.6.3	Vymazat uživatele .....	76
4.6.4	Nový uživatel .....	76
4.6.5	Historie .....	77
4.6.6	Historie klienti.....	79
4.6.7	Historie léky .....	79
4.6.8	Nový klient.....	80
4.6.9	Editace klienta .....	81
4.6.10	Smazání klienta .....	82
4.6.11	Zobrazení klienta.....	83
4.6.12	Skupiny léků.....	85
4.6.13	Vstup návštěvy .....	86
4.6.14	Aktivní návštěvy .....	86
4.6.15	Odchod návštěvy .....	87
4.6.16	Black list.....	88
	<b>ZÁVĚR .....</b>	<b>89</b>

---

<b>ZÁVĚR V ANGLIČTINĚ.....</b>	<b>90</b>
<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>91</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>93</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>94</b>
<b>SEZNAM TABULEK.....</b>	<b>97</b>
<b>SEZNAM PŘÍLOH.....</b>	<b>98</b>

## ÚVOD

Ošetrovatelská dokumentace se používá nejen v sociální péči ale i ve zdravotnictví. Pro účely této diplomové práce jsem se zaměřil na ošetrovatelskou dokumentaci využívanou domy pečujícími o seniory. Obecně problematika zdravotnictví a sociální péče se stává stále více aktuální, kdy se tato oblast přelévá více ze státních rukou do rukou soukromých. Tento stav bude pokračovat několik dalších let a bude potřeba změnit nejen legislativu upravující tuto problematiku. Spolu s touto problematikou nastává další problém a to rychlejší nástup IT technologií do této oblasti se zastaralými legislativními úpravami a tento trend se také musí odrazit na změně zákonů. Obecně finanční problém má v tomto sektoru za následek jeden důležitý faktor a to úbytek personálu s vyšší zdravotnickou kvalifikací. Stále více zařízení pracuje v provozu, kdy je kriticky málo kvalifikovaného personálu na jednotlivých směnách a provoz tak zajišťuje nižší zdravotnický personál. Co takový problém může způsobit? Proč vzniká dokumentace? Kdo jí vytváří? Jakou formu má a jaká je situace v této problematice? Jak bezpečné je pro klienty využívající sociální služby využití IT systémů v této oblasti při stávající legislativě? Na tyto a další otázky se pokusím odpovědět touto prací, kdy výstupem bude mé vlastní řešení, konkrétně software, který se pokusí zlepšit bezpečnostní úroveň sociálních služeb v domovech pro seniory při používání kombinace klasické a elektronické dokumentace.

## **I. TEORETICKÁ ČÁST**

## 1 DOKUMENTACE

Obecně se dají dokumentace rozdělit na dokumentaci ošetrovatelskou, zdravotnickou a sociální. Nakládáním s dokumentací se rozumí zpracování, vedení, evidence. V první řadě je potřeba zajistit ochranu osobních údajů dle zákona č. 101/2000 Sb. o ochraně osobních údajů v platném znění. Kromě vedení zdravotnické a ošetrovatelské dokumentace jsou osobní údaje vyžadovány např. kvůli poskytování sociálních služeb, jako je vyřizování důchodů nebo sociálních dávek. Pracovníci, kteří mají přístup k osobním údajům, jsou podle zákona č. 108/2006Sb., § 100 vázáni mlčenlivostí, a to i po ukončení pracovního poměru

### 1.1 Zdravotní dokumentace

Zdravotní dokumentace týkající se pečovatelských domů pro seniory se řídí následujícími zákony.

Zákon č. 20/1966 Sb. o zdraví lidu.

Zákon č. 260/2002 Sb.

Zákon č. 111/2007 Sb. byl přijat s platností od 15. května 2007

Novela zákona č. 20/1966 Sb. z 26. června 2001 (dále jen "novela"), provedená zákonem č. 260/2001 Sb., platným od 1. srpna 2001, vložení § 67 - všeobecná povinnost zdravotnických zařízení vést zdravotnickou dokumentaci.

Novela specifikuje formu vedení dokumentace nově (§ 67b odst. 5 až 8):

- dokumentace může mít textovou, grafickou nebo audiovizuální formu,
- dokumentace může být vedena v listinné nebo elektronické formě,
- údaje pořízené v listinné formě (a to nejen text, ale i např. obrázky) nesmí být převedeny do elektronické formy bez archivace listinného originálu,
- zápis v elektronické formě bez zaručeného elektronického podpisu je nutno vytisknout na papír, který se opatří datem a podpisem osoby, která zápis provedla, a zařadí se do dokumentace pacienta; každá část (list, sestava, výtisk) se považuje za samostatnou část dokumentace a musí být nejen všechny podepsány, ale také opatřeny dostatečnou identifikací pacienta,

- zápisy v elektronické formě obsahující zaručený elektronický podpis pořizovatele musí splňovat tyto podmínky:
- být podepsány tímto podpisem v každé jednotlivé části dokumentace,
- být zálohovány minimálně jednou denně na záložní médium,
- jsou z nich nejméně jedenkrát ročně pořízeny archivní kopie, které není možné dodatečně upravovat (např. na CD-ROM),
- jejich archivní kopie musí umožnit čitelnost a přístupnost informací po celou předepsanou dobu archivace.

Tím, že novela nyní taxativně vyjmenovává kategorie osob, které mohou při plnění svých konkrétních úkolů nahlížet do dokumentace v nezbytně nutném rozsahu, vyčleňuje pro tyto (a jen tyto!) konkrétní případy takto přístupné informace z okruhu povinnou mlčenlivostí chráněných informací, k jejichž poskytnutí dosud bylo zapotřebí získat souhlas klienta. I nadále zůstává věcí názoru, že právo na nahlížení lze interpretovat i jako právo na kopie (potřebné části) dokumentace.

Kdo smí nahlížet do dokumentace:

- Zdravotní pracovníci a jiní odborní pracovníci v souvislosti s poskytováním zdravotní péče
- Funkcionáři komor, šetřící stížnosti na lékaře nebo lékárníky
- Revizní lékaři pojišťoven
- Soudní znalci v oboru zdravotnictví (většinou lékaři)
- Lékaři ve státní správě, šetřící stížnosti a podobná podání
- Lékaři ministerstva zdravotnictví, šetřící stížnosti a podobná podání
- Lékaři státního úřadu pro jadernou bezpečnost
- Členové ústřední znalecké komise a územních znaleckých komisí
- Pracovníci orgánu ochrany veřejného zdraví (“hygienici”)
- Lékaři orgánů sociálního zabezpečení, lékaři úřadů práce a lékaři okresních úřadů při posuzování zdravotního stavu pro účely poskytování různých dávek a služeb, při odvodech branců a dalších úkolech
- Zaměstnanci zdravotnických zařízení (nebo smluvních podniků), zajišťující zpracování dat při vedení dokumentace
- Zaměstnanci státu nebo zpracovatele, kteří zajišťují úkoly nzis (“statistika”)

- Pověřené zdravotnické zařízení, které posuzuje zdravotní stav pro účely pojistných a podobných smluv
- Veřejný ochránce práv v souvislosti s šetřením
- Inspektoři ústavu pro odborné zjišťování příčin leteckých nehod
- Zaměstnanci státního ústavu pro kontrolu léčiv, kteří provádějí kontrolu

## 1.2 Ošetřovatelská dokumentace

Ošetřovatelská dokumentace se postupem času stává nedílnou součástí dokumentace zdravotnické. Její základní struktura vychází z Koncepce ošetřovatelství ČR a je nedílnou součástí ošetřovatelského procesu. Shromažďuje informace o nemocném z ošetřovatelského pohledu, záznamy o poskytnuté péči a jejím efektu na pacienta. Všechny tyto údaje o fyzickém, emocionálním i sociálním stavu klienta, jeho reakce na lékařské a ošetřovatelské intervence slouží všem ostatním členům týmu a přispívají k uspokojování potřeb pacienta/klienta. [4]

Ošetřovatelskou dokumentaci zavede vrchní nebo staniční sestra při příchodu nového klienta. Struktura ošetřovatelské dokumentace je nejčastěji následující:

- Ošetřovatelská anamnéza
- Plán ošetřovatelské péče
- Ošetřovatelský plán
- Překladová ošetřovatelská zpráva
- Ošetřovatelský dekurz
- Sledování fyziologických funkcí
- Barthelův test
- Denní hlášení
- Protokol o pádu
- Test mentálních funkcí

Ošetřovatelská dokumentace poskytuje aktuální přehled o vývoji zdravotního stavu, je zdrojem informací o aktuálních potřebách jedince, cílech péče a jednotlivých intervencí, potřebných k jejímu naplnění. Dává ucelený a systematický přehled o poskytované péči a slouží jako doklad, že poskytovaná péče byla provedena. Je formou písemné komunikace

mezi členy ošetrovatelského týmu, usnadňuje přesné sdělování informací o klientovi, pozitivní či negativní reakce pacienta na léčbu a ošetrovatelskou péči, umožňuje hodnotit účinnost a neúčinnost jednotlivých ošetrovatelských zákroků.[4]

### 1.2.1 Základní složky

Základní ošetrovatelská dokumentace obsahuje:

Vstupní ošetrovatelský záznam, jehož součástí je ošetrovatelská anamnéza, subjektivní hodnocení nemocného sestrou a objektivní hodnocení pomocí měřících technik.

Plán ošetrovatelské péče, jehož součástí je ošetrovatelská diagnóza, cíle ošetrovatelské péče, plánované sesterské činnosti, hodnocení efektu péče.

Výstupní ošetrovatelská zpráva, která zajišťuje kontinuitu péče při překladech pacienta na jiné pracoviště.

Doplňující záznamy o různých ošetrovatelských testech, které jsou typické pro druh zařízení a poskytovanou péči, např. hodnocení soběstačnosti, hodnocení bolesti, hodnocení rizika vzniku dekubitů, hodnocení rizika pádu, polohovací záznam, záznam o příjmu a výdeji tekutin, nutriční screening, záznam o stolici, záznam o převazu rány/ošetření kůže, záznam tlaku a váhy, karta diabetika, edukační záznam. [4]

- **Ošetrovatelská anamnéza**

Ošetrovatelská anamnéza v elektronické podobě je vyplňována sestrou, na základě informací od klienta. Jedná se o informace, které se použijí při zpracování ošetrovatelské diagnózy a plánu ošetrovatelské péče.

Informace pro stanovení ošetrovatelské anamnézy se získávají pomocí ošetrovatelských modelů. Např. podle modelu Marjory Gordon.[5]

- **Ošetrovatelský plán**

Pro řešení ošetrovatelských problémů se stanovuje ošetrovatelský plán. Jde o jakousi písemnou formu návrhu na postup uspokojování individuálních potřeb klienta. Obsahuje ošetrovatelské diagnózy v pořadí naléhavosti, jejich řešení, cíle ošetrovatelské péče, plán sesterské intervence, realizace, zápis o jejich provedení a zhodnocení efektu poskytnuté péče.



- Realizace ošetrovatelského plánu

Jedná se o realizaci ošetrovatelského plánu, vzniklého viz předchozí kapitola. Plnění a dodržování toho ošetrovatelského plánu slouží jako podklad pro pojišťovny.

- Denní záznamy

Denní záznam lze předvést jako výkaz realizace ošetrovatelského plánu. Každý záznam musí mít identifikovaného tvůrce.[5]

### 1.2.2 Základní principy vedení

- Vedení ošetrovatelské dokumentace je nedílnou součástí ošetrovatelské praxe
- Profesionální vedení ošetrovatelské dokumentace je známkou kvalifikovaného pracovníka
- Základní složky ošetrovatelské dokumentace jsou: ošetrovatelská anamnéza, záznam vývoje stavu pacienta/klienta, ošetrovatelský plán a překladová/propouštěcí zpráva.
- Záznamy by neměly obsahovat, nespisovné výrazy, bezvýznamné fráze, irelevantní spekulace a urážlivé subjektivní výroky
- Záznamy by měly být konkrétní, soustavné a přesné
- Záznamy by měly být psány přehledně a tak, aby text nemohl být vymazán
- Je třeba zajistit, aby každý zápis do ošetrovatelské dokumentace byl snadno identifikovatelný
- Veškeré zápisy, které uděláte do záznamů pacienta/klienta, mohou být v určitém okamžiku podrobně posuzovány
- Ošetrovatelské záznamy mají být pořizovány vždy při změně pacientova/ klientova zdravotního stavu, za každou pracovní směnu a volná místa proškrtnutá.
- Při auditu ošetrovatelské dokumentace můžete posoudit její úroveň a identifikovat oblasti, kde by mělo dojít ke zlepšení příp. k poučení pracovníků
- Použití dokumentace při výzkumu by mělo být schváleno Vaší místní etickou komisí
- Princip důvěrnosti zaznamenávaných informací o Vašich pacientech/klientech je také u počítačových záznamů stejně důležitý, jako u všech ostatních záznamů
- Vaší povinností je chránit důvěrnost dokumentace pacienta/klienta
- Příspěvek každého člena ošetrovatelského týmu do ošetrovatelské dokumentace by měl být považován za stejně důležitý [4]

Dobré vedení dokumentace pomáhá dodržet stanovenou kvalitu péče pro pacienty/klienty.

### **1.2.3 Právní úprava**

Vedení dokumentace se řídí vyhláškou č. 64/2007 Sb. Změnila se tak vyhláška č. 385/2006 Sb., o zdravotnické dokumentaci, ve znění vyhlášky č. 479/2006 Sb. Zákon č.20/1996 Sb., o péči o zdraví lidu, ve znění změn a doplňků v § 55 odst. 2, písm. d) ukládá zdravotnickým pracovníkům povinnost zachovávat mlčenlivost o skutečnostech, získaných v souvislosti s výkonem svého povolání. Zákon 101/2000 Sb., o ochraně osobních údajů, v § 13.[5]

## **1.3 Sociální dokumentace**

Sociální dokumentaci lze představit jako osobní desky klienta, které se uloží na ošetřovně a obsah se dá shrnout jako: vstupní dotazník, osobní anamnéza, popsané listy průběhu práce, založené individuální plány a další potřebné osobní informace a údaje o klientovi. Na tvorbě se podílí zejména klíčový pracovník klienta a pracovníci v přímé obslužné péči.

## **1.4 Nahlížení do dokumentace**

### **1.4.1 Klient**

Klient má právo nahlížet do své dokumentace vedené o jeho osobě po domluvě za přítomnosti oprávněného personálu. Klient má právo rozhodnout o fyzické osobě, která též může nahlédnout do dokumentace o něm vedené.

### **1.4.2 Lékař, vrchní sestra, staniční sestra**

Nahlízejí v rozsahu nutného pro výkon povolání do zdravotní dokumentace.

### **1.4.3 Pracovníci přímé obslužné péče**

Pracovníci přímé obslužné péče mohou být seznámeni s určitou částí ošetřovatelské a sociální dokumentace v rámci plnění své práce.

## 2 ELEKTRONICKÁ DOKUMENTACE

Elektronická dokumentace se v dnešní IT době stává standardem v téměř každém sociálním a zdravotním zařízení. Tento druh softwaru (SW) se snaží komplexně pokrýt veškerou dokumentaci. Problémem je, že v takovém množství informací je složité dohledat a vytáhnout všechny důležitá data v co nejrychlejším čase. Nicméně v těchto komplexních systémech je nejen všechna potřebná dokumentace, ale nabízí se i například správa stravování, nebo zaměstnanecké docházky či kontrola práce.

Na otázku proč zavést evidenci této dokumentace v elektronické podobě lze odpovědět několika body:

- Snadnější monitoring péče a její vyhodnocení
- Rychlejší a kvalitnější komunikace mezi ošetřujícím personálem
- Zrychlení přehledu o klientech
- Lepší dostupnost jednotlivých dokumentací
- Lepší čitelnost dokumentace než při použití klasické dokumentace
- Možnost efektivněji měnit informace v dokumentaci s jednoznačným podpisem osoby tuto akci provádějící

Nevýhody proč nevyužít elektronickou dokumentaci mohou být:

- Časová náročnost pro zadávání dat
- Větší finanční nároky na vybavení
- Nutnost školení zaměstnanců pro použití výpočetní techniky nebo systému

Příkladem společností, které se zabývají vývojem takových typů aplikací:

- SMS, spol. s.r.o.
- HiComp a.s.
- Medicalc
- ICZ. a.s.
- IRESOFT

## 2.1 Základní legislativa

Elektronická dokumentace musí být vypracovávána stejně jako klasická dokumentace, takže všechny její součásti musí být stejné.

Podle zákona č. 227/2007 Sb., musí všechny části zdravotnické dokumentace obsahovat elektronický podpis. Jde o nahrazení klasického podpisu v tištěné podobě podpisem elektronického dokumentu. Elektronický podpis je chápán zpravidla jako číslo, které vytváří podepisující osoba pomocí svých dat pro vytváření elektronického podpisu a pomocí zprávy, kterou podepisuje. Elektronický podpis v současné době může být i ve formě elektronického snímání prstů, snímání oční duhovky, manuálního zadávání číselného kódu (PIN), prokázání totožnosti elektronickým kódem v čipu magnetického pásku a digitálního podpisu.

## 2.2 Základní ochrana dat

Základním úkolem by měla být ochrana dat před destrukcí, ochrana před neoprávněným přístupem, kompromitací a zneužitím. Na tento typ dat se vztahuje zákon č. 101/2000 Sb., o ochraně osobních údajů. Tyto data musí být přístupná pouze oprávněným osobám, tudíž je potřeba správně nastavit používané informační technologie a hlavně databázový systém.

Dalším typem legislativy vztahující se k této problematice je úmluva o ochraně základních lidských práv a svobod a listina základních práv a svobod, která je naší právní úpravou.

Pro práci s informačním systémem (IS) je potřeba rozlišit jednotlivé uživatele a nastavit jim odpovídající práva, jakou část mohou vidět, upravovat, tisknout, atd. Pravidla určující IT politiku společnosti by mělo vydat vedení společnosti a zaujmout jednoznačný postoj pro tuto problematiku.

## 2.3 Informační systém Cygnus

Informační systém Cygnus je systém pro sociální zařízení z dílen firmy IReSoft, s.r.o., která sídlí v Brně.

Jedná se o systém vhodný pro poskytovatele sociálních služeb. Skládá se z několika modulů, které tvoří jeden ucelený funkční systém. Umožňuje připojení jednotlivých modulů, tudíž je variabilní.

Systémové požadavky dle výrobce:

PC Pentium 4, 2 GHz, 1 GB RAM, SVGA 1024x768, myš, CD-ROM, tiskárna, připojení k internetu, Microsoft Windows XP SP2, Microsoft Word 2003.

### 2.3.1 Moduly



Obr. 1 Cygnus – moduly [13]

### 2.3.2 Sociální část

Tato část slouží pro editaci klientů, jak současných, tak i těch minulých. Vyplňují se zde základní informace jako adresa, kontaktní osoby. Je zde možnost i finančního zpracování klientů, jako jsou důchody, finanční depozita.

Specifikace výrobce:

- Evidence aktuálních a bývalých klientů, včetně jejich osobních údajů, adres, kontaktů
- Evidence žadatelů o péči, včetně bodového hodnocení dle Vámi stanovených kritérií
- Stanovení předpisu úhrady pro dospělé klienty a děti do 18 let ve variantách celoročního, týdenního nebo denního pobytu, výpočet pro přechodné pobyty.
- K dispozici nástroj pro hromadnou valorizaci důchodů, hromadný přepočítání úhrad při změně číselníků atd.
- Možnost generovat dokumenty pomocí šablon ve Wordu. Vytvoření smlouvy o poskytování sociálních služeb i podle vlastní šablony
- Nástroj Přehled přítomnosti umožňuje hromadné zadávání nepřítomnosti (celodenní i částečné) a hlášení počtu strávníků do modulu Stravovací část
- Výpočet celodenních i částečných vratek za nepřítomnost
- Evidence finančních depozit klientů (např. peněžní depozita, vkladní knížky, osobní účty) s možností vedení depozitních pokladen
- Užitečné nástroje na hromadné zadávání dokladů. Tisk pokladních dokladů, inventur, výkazů příjmů a výdajů, pokladních deníků, uzávěrek apod.
- Evidence hmotných depozit klientů. Tisk inventur, vyřazovacích protokolů aj.
- Efektivní nástroj pro zpracování hromadné výplatnice důchodů. Automatické vytvoření pokladních dokladů a plateb do vyúčtování
- Evidence fakultativních služeb a doplatek za léky
- Nástroj pro vyúčtování klientů zahrnující předpisy a platby za úhradu (lze dělit na bydlení a stravu), platby příbuzných, příspěvek na péči, fakultativní služby, doplatky za léky, příspěvek na dítě
- Ordinance léků s možností vytvářet hromadné objednávky, tisknout recepty, medikační karty, podklady pro lékárnou
- Možnost evidovat a objednávat pomůcky, včetně tisku poukazů. Sledování dekursu klienta. Tisk příkazů k transportu, poukazů na vyšetření
- Účast klientů na akcích, včetně hodnocení a tiskových výstupů
- Možnost ukládání libovolných dokumentů, obrázků a záznamů na kartu klienta
- Možnost editovat všechny sestavy ve Wordu, export dat do Excelu

- Import plateb z elektronického bankovníctví, import doplatků za léky, export vratek do programu Pošta 2002, export plateb z výplatnice na kompatibilní médium ČS aj.
- Podklady pro statistické vykazování dat, tisky, přehledy a mnoho dalšího.
- Díky širokým možnostem nastavení snadno přizpůsobíte modul na míru Vašemu zařízení.

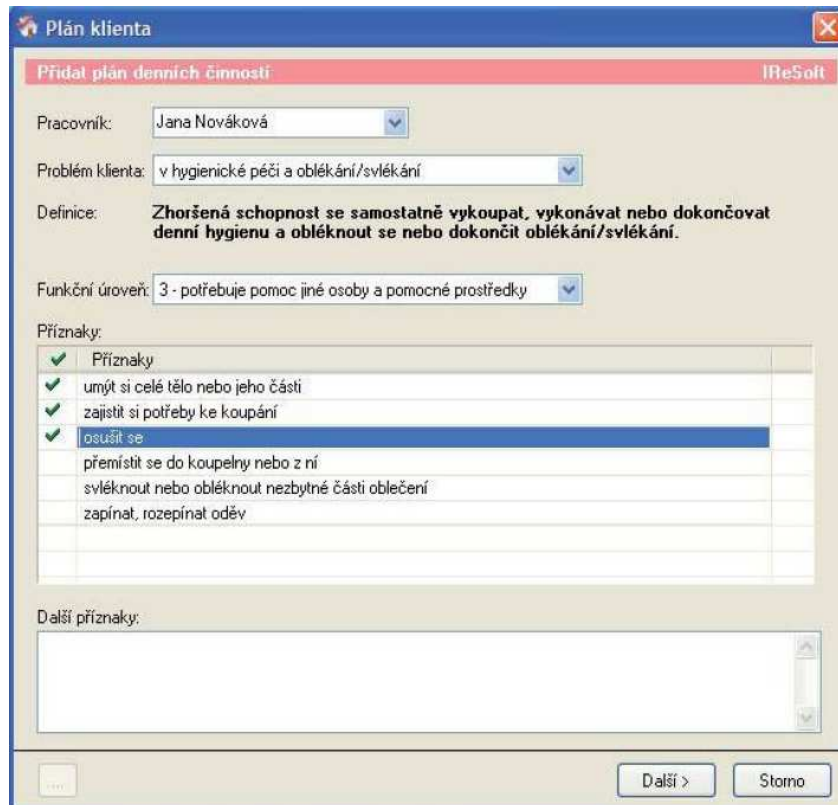
### 2.3.3 Dokumentace klienta

Modul, ve kterém se může vést kompletní ošetrovatelská a sociální dokumentace, je určen pro všeobecné sestry, sociální pracovníky a pracovníky sociální služby. Je zde možnost vytváření různých formulářů, tvorba individuálního plánu, plán péče, plán rizik nebo ošetrovatelský plán.

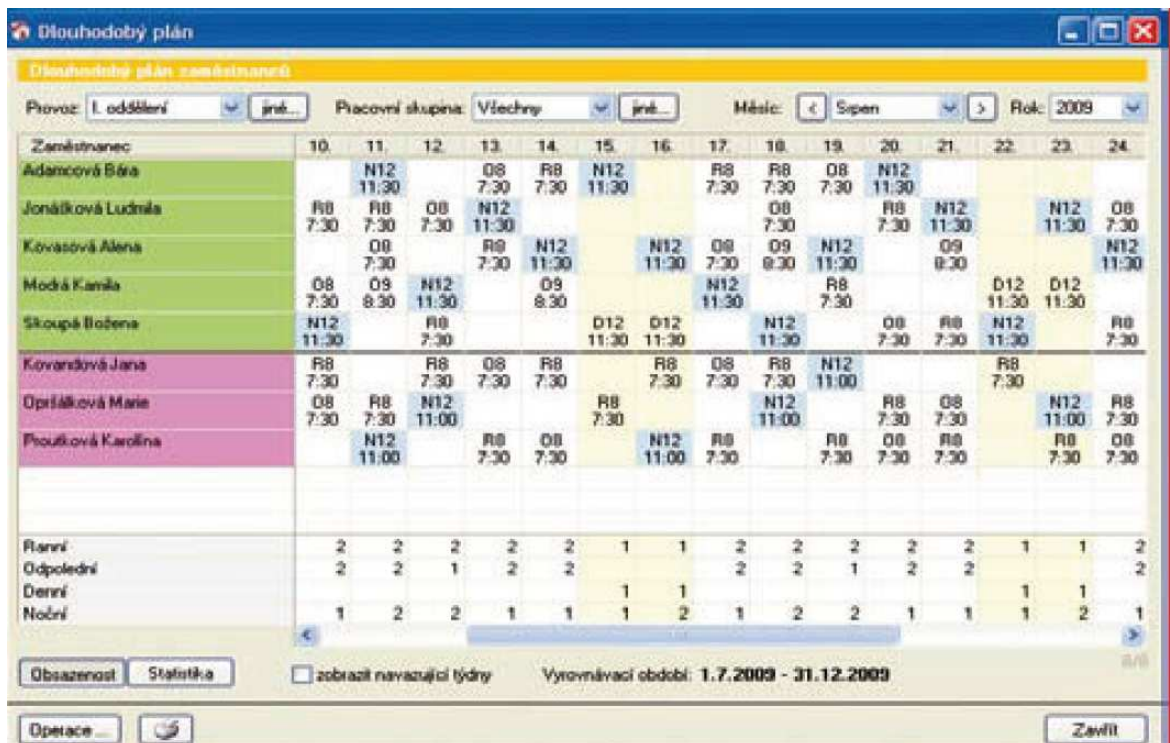
Zajímavá je funkce kontroly provedení péče. V základě se jedná o čtečku, která načte kód, který odpovídá jednotlivému úkonu při sociální péči. Čtečka se pak synchronizuje s aplikací a je možné zobrazit, který úkon byl vykonán a který ne. Úkony se dají přepočítat na body a ty se odesílají na pojišťovny.



*Obr. 2 Cygnus – čtečka [13]*



Obr. 3 Cygnus – čtečka [13]



Obr. 4 Cygnus – dlouhodobý plán[13]



### 2.3.4 Vykazování na ZP

Tento modul slouží pro výkaz úkonů na pojišťovny. Sdílí se zde informace ze sociální části a z modulu dokumentace se vyberou úkony, které byly provedeny.

Informace vývojové společnosti dostupné z [13]:

- Pořizování výkonů odborností 913, 004, 902 a 925 dle datového rozhraní VZP.
- Nástroje pro hromadné zadávání výkonů, ruční editace výkonů v přehledné tabulce.
- Snadné načtení zdokumentovaných intervencí z modulu Dokumentace klienta.
- Evidence a tisk poukazů ORP a DP, hromadné kopírování mezi měsíci, hromadný tisk. Načítání diagnóz a ordinace léků z modulu Dokumentace klienta.
- Vyúčtování cestovních náhrad.
- Vykázání výkonů a vytvoření faktur pomocí jediného tlačítka.
- Hromadný nástroj k uložení dávek na diskety, tisk průvodních listů a faktur.
- Automatické kontroly správnosti pořízených dat – maximální četnost výkonů, kontrola rodných čísel, správnost diagnóz, IČZ, IČP, územní pracoviště apod.
- Možnost pořizovat data na oddělených pracovištích a následně vykázat úkony na jednu fakturu.
- Opravné dávky, vykazování výkonů i zpětně.
- Aktualizace číselníků VZP čtyřikrát ročně.
- Tisky, přehledy, statistiky a mnoho dalšího.

### 2.3.5 Stravovací část

Tímto modulem se komplexně spravuje jídelní část, kdy je možné sledovat nutriční hodnoty, sestavovat jídelní lístky. Kromě klientské části je zde možná správa i zaměstnanecké části stravování.

Informace vývojové společnosti dostupné z [13]:

- Číselník jídel a receptur, normování receptur, možnost hromadného zadání.
- Tvorba jídelních lístků, více variant pro každý druh jídla. Rozbor podle jednotlivých diet strážníků, podle ceny nebo nutričních hodnot.

- Hromadné zadávání počtu strážníků s možností načtení porcí z modulu Sociální část (počty klientů) a nástroje Objednávka stravy (zaměstnanci a cizí strážníci).
- Vystavení jedné nebo více výdejek pro každý jídelní lístek, úprava výdejek – vložení suroviny nebo celého jídla, odebrání položek výdejky, zarovnání množství dle minimálního výdeje. Výdejky lze snadno odepsat ze skladu potravin.
- Sledování nutričních hodnot jídel, jídelních lístků, rozborů dle druhů jídel, diet, za období apod. Program obsahuje číselník nutričních hodnot pro přibližně 500 základních potravin.
- Vytvoření vlastních kategorií (diet) strážníků, zadávání stravovacích norem pro jednotlivé druhy jídel.
- Denní sledování skutečné spotřeby a stravovacích norem.
- Objednávky stravy pro zaměstnance a cizí strážníky s návazností na stravovací systém (objednávkový a výdejní terminál).
- Možnost porovnání objednávek zaměstnanců s jejich docházkovými výkazy v modulu Zaměstnanci (stravovací povinnost).
- Vyúčtování stravného zahrnuje předpisy a platby, převody mezi měsíci, hromadné nástroje, tisk pokladních dokladů, stravenek, podkladů pro vyúčtování aj.
- Vytváření měsíčních uzávěrek, možnost zrušení uzávěrky bez ztráty dat.
- Tisk výdejek potravin, rozpisů pro kuchaře, jídelních lístků, rozborů nutričních hodnot, přehledů stravovacích norem, skutečné spotřeby a mnoho dalšího.

### 2.3.6 Sklady

Libovolné množství skladů sloužící k evidenci materiálů, provádění inventur a podobně.

Informace vývojové společnosti dostupné z [13]:

- Neomezený počet skladů vedených metodou váženého aritmetického průměru.
- Položky lze evidovat s cenami včetně DPH i bez DPH.
- Jednoduchý nástroj pro hromadnou správu dokladů – rychlé vytvoření příjemky a výdejky, oprava, smazání i tisk.
- Automatické číslování příjemek, výdejek a skladových karet. Možnost nastavit vlastní formát číselných řad, zvláštní číselná řada pro příjemky a výdejky.
- Rozdělení výdejů pomocí číselníku dodavatelů a příznaků.

- Položky skladu lze dělit na podsklady, možnost sledování skladovací doby, minimální a maximální zásoby.
- Vytváření měsíčních uzávěrek, možnost zrušení uzávěrky bez ztráty dat.
- Možnost sledování výdejů na nákladová střediska, export do účetnictví.
- Tisk výkazů, inventur, přehledu příjmů a výdejů, příjemek a výdejek, skladovacích dob a mnoho dalšího.

### 2.3.7 Zaměstnanci

Modul pro kompletní správu zaměstnanců. Je určen jak pro vedoucí zaměstnance, tak zaměstnance účetní správy. Mohou se zde vyplňovat profily, plánovat směny, vytvářet podklady pro mzdy a další.

Informace vývojové společnosti dostupné z <http://www.iscygnus.cz/cz/53/zamestnanci/>:

- Tvorba rozpisů služeb pracovníků v nepřetržitém / směnném provozu pomocí tříúrovňového modelu – dlouhodobý plán, měsíční plán, skutečná docházka.
- Efektivní plánování nerovnoměrně rozvržené pracovní doby v rámci vyrovnávacího období, snadné vytváření směn podle šablon s ohledem na jejich minimální obsazenost.
- Snadné zpracování a vyhodnocení docházky zaměstnanců, tvorba podkladů pro zpracování mezd (příplatky za víkendy, noční, svátky, dovolené, nemoci atd.).
- Možnost využití docházkových čteček pro stanovení skutečné docházky zaměstnanců, včetně porovnání s měsíčním plánem. Součástí je samostatný modul pro automatické stahování dat ze čteček Cygnus – Monitor.
- Samozřejmostí je možnost úpravy měsíčních výkazů, zadávání absencí, uznávání přesčasů, uzamykání zpracovaných dnů, ruční oprava vypočtených hodnot.
- Vlastní nastavení pracovních skupin a směn dle provozních podmínek (např. fond, zaokrouhlování, tolerance, časy a délky směn, přestávky, dělení příplatků), automatické odečítání přestávek.
- Výpočet měsíčních fondů, přenosů mezi měsíci, statistiky, upozornění na rozpory se zákoníkem práce apod.
- Tisk měsíčních výkazů, celkové a souhrnné sestavy, záznamy ze čtečky, přehled aktuálně přítomných zaměstnanců atd.

- Export dlouhodobého plánu a měsíčních výkazů do mzdových programů (Vema, Gordic aj.).
- Sestavení vzdělávacích potřeb zaměstnanců a stanovení osobních cílů. Jednoduchá tvorba individuálních vzdělávacích plánů a vzdělávacího harmonogramu organizace pomocí hromadného nástroje.
- Hodnocení zaměstnanců dle 16 kritérií ve třech úrovních – sebereflexe, hodnocení přímým nadřízeným a mezi kolegy navzájem.
- Přehled naplnění zákonných požadavků na vzdělávání zaměstnanců (hodiny / kredity), včetně sledování nákladů na jejich vzdělávání.
- Možnost ukládání libovolných dokumentů, obrázků a záznamů na kartu zaměstnance.
- Zaměstnanci mají přístup k vlastním rozpisům služeb, měsíčním výkazům, vzdělávacím aktivitám a hodnocení kolegů prostřednictvím modulu Můj Cygnus.
- Údaje evidovaných zaměstnanců jsou nasdíleny do modulů Dokumentace klienta, Stravovací část a Majetek. Program informace využívá i při nastavení přístupových práv do systému.
- Vedoucí jednotlivých provozů mají přístup ke zpracování agendy pouze u svých podřízených.
- Modul také přináší podklady pro statistické vykazování dat na MPSV, tisky, přehledy a mnoho dalšího.

Zajímavou funkcí k tomuto modulu je možnost dokoupení čtečky pro evidenci docházky. Čtečka je propojená s aplikací a automatiky provádí evidenci docházky.



Obr. 5 čtečka pro docházku [13]

### 2.3.8 Majetek

Modul pro správu hmotného i nehmotného majetku. Může se zde vytisknout vyřazovací protokol, udělat inventura a podobně.

Informace vývojové společnosti dostupné z [13]:

- Evidence dlouhodobého hmotného a nehmotného majetku, odepisovaného i neodepisovaného, vlastního i svěřeného. Libovolný počet skupin drobných majetků. Vlastní nastavení názvů a finančních hranic.
- Možnost hromadného uplatnění účetních odpisů (ročně, pololetně, čtvrtletně, měsíčně).
- Hromadný nástroj na pořizování a vyřazování drobného majetku.
- Automatické číslování inventurních čísel. Možnost nastavit vlastní formát číselných řad, zvláštní číselná řada pro každou skupinu majetku.
- Zodpovědné osoby můžete vybírat z pracovníků evidovaných modulem Zaměstnanci.
- Možnost ukládání libovolných dokumentů, obrázků a záznamů na kartu majetku.
- Číselníky středisek, místností, tříd, druhů, odpisových skupin atd.
- Export do Wordu – Zařazovací a Vyřazovací protokol, Karta majetku a jiné.
- Tisk inventurních soupisů, místních seznamů, přesunů majetku, odpisových plánů, uplatněných odpisů a mnoho dalšího.

### 2.3.9 Manažerská část

Speciální modul pro ředitele, nebo vedoucí pozice. Zobrazí se aktuální stav zařízení, ekonomické funkce, grafy, tabulky z ostatních modulů. Tento modul dokáže zobrazit slabá místa nainstalovaného systému, jako je špatně nastavené zálohování atd.

Informace vývojové společnosti dostupné z [13]:

- Tvorba výstupů pro statistické výkazy dat na MPSV – údaje program získává z modulů Sociální část, Dokumentace klienta a Zaměstnanci.
- K dispozici jsou informace o využívání jednotlivých modulů informačního systému, jejich nástrojů a funkcí, přehledy, statistiky a grafy.

- Dostupné informace z modulu Sociální část: počty klientů a jejich členění dle středisek a služeb, přehled nástupů a ukončení, narozeniny klientů, informace o úhradách klientů za pobyt, doplatky příbuzných, neuhrazené částky, rozdělení a výše příspěvků na péči aj.
- Informace z modulu Dokumentace klienta: využití jednotlivých částí modulu, upozornění na nesplnění kritérií standardu kvality č. 5.
- Informace z modulu Vykazování na ZP: fakturované částky na jednotlivé pojišťovny, počty vykazovaných klientů, průměrné částky na klienta.
- Informace z modulu Stravovací část: údaje o jídelních lístcích, aktuální spotřeba a stravovací normy.
- Informace z modulu Sklady: hodnoty jednotlivých skladů, obraty největších dodavatelů ve skladu potravin.
- Další funkce: upozornění na zálohování, aktualizace systému, nastavení přístupových práv, slabé zabezpečení uživatelských hesel a jiné.

## 2.4 Fons enterprise - nemocniční informační systém

Informace od výrobce [14]:

NIS FONS Enterprise obsahuje funkcionality, které umožňují vedení zdravotní dokumentace, a podporuje provozní činnosti na jednotlivých klinických pracovištích. Zajišťuje zadání potřebných administrativních údajů, pořizování výkazných a statistických dat a podporuje činnost lékařů a sester při dokumentaci zdravotního stavu pacienta při ambulantním ošetřování i hospitalizaci.

Obsahově pokrývá všechny klinické procesy řešené ve stávajících NISech, navíc umožňuje definovat jakoukoli strukturovanou dokumentaci pro specifické potřeby jednotlivých odborností.

NIS FONS Enterprise je založen na moderním grafickém uživatelském rozhraní, které co nejvíce podporuje ergonomickou práci koncového uživatele. Snaží se o co největší přehlednost, pracuje s více okny dashboardu na obrazovce, které koncovému uživateli poskytují právě ty informace, které pro danou činnost potřebuje. Například kromě seznamu pacientů ležících na stanici lze zobrazit v dalších oknech i přehled klinické dokumentace

vybraného pacienta, jeho laboratorní výsledky, seznam úkolů přihlášeného uživatele, případně seznam aktuálně doručených výsledků atd.

NIS FONS Enterprise používá v souladu se současnými trendy pro ovládání místo dříve používaného menu s výčtem funkcí přehlednější prvek – ribbon (pás karet). Obsah ribbonu se mění podle aktivního okna, se kterým uživatel právě pracuje – obsahuje jiné možnosti např. v seznamu pacientů a jiné při zápisu dokumentace.

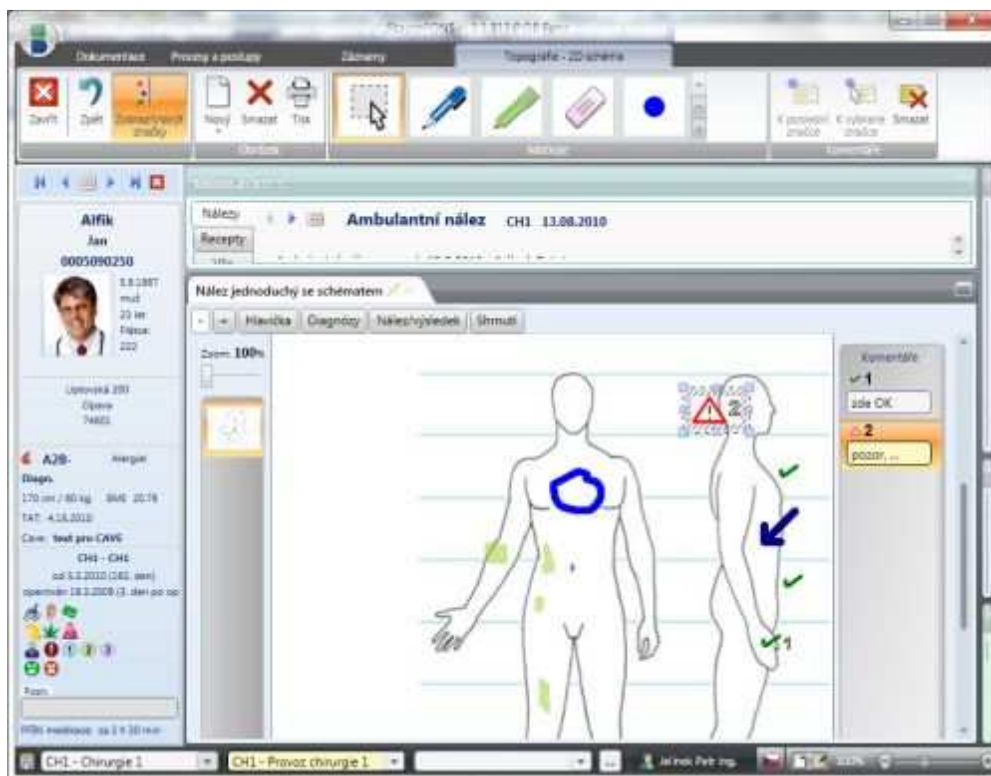
NIS FONS Enterprise používá různé grafické prvky. Nastavení pracovní plochy lze modifikovat pro jednotlivé činnosti koncového uživatele: jinak může vypadat pracovní plocha při zápisu ambulantního vyšetření, jinak při vizitě.

Pracovní plochu lze optimálně přizpůsobit velikosti použitého monitoru. Pro větší monitory lze na pracovní plochu umístit více samostatných oken a optimálně tak využít prostor zobrazením potřebných informací. Uživatel může průběžně podle typu práce měnit velikost zobrazovaných informací (rozlišení) pomocí ovládacího prvku v pravém dolním rohu (+ -).

NIS FONS Enterprise je procesně orientovaný systém. Umožňuje nastavit procesy – od jednoduchých sledů spouštěných funkcí při běžných činnostech (administrativní a lékařský příjem pacienta, propuštění) po složité workflow odpovídající standardním postupům léčby určité diagnózy. Při definici procesů lze pracovat s mnoha faktory, které zajistí vysokou flexibilitu v nastavení procesu.

Procesy se skládají z jednotlivých kroků, u kterých lze nastavit, kdy mají proběhnout, kdo je má vykonat, zda je nutné další krok zahájit až po ukončení předchozího, evidují se časy – kdy byl krok zahájen, ukončen. Proces, je možné větvit do několika směrů, lze nastavit podmíněné rozhodování na základě zadaných dat, povinnost kroků atd. Procesy lze vyhodnocovat – sledovat délku trvání, splnění, resp. odchylky reálného od nastaveného procesu atd.

NIS FONS Enterprise umožňuje pracovat s grafikou přímo v dokumentaci pacienta. Do dokumentace pacienta lze vkládat schematické obrázky, do kterých může uživatel pomocí různých značek a poznámek znázornit např. dekubit, poranění atd. Součástí dokumentace může být i obrazová informace, zvuková informace, video a podobně.



Obr. 6 FONS Enterprise [14]

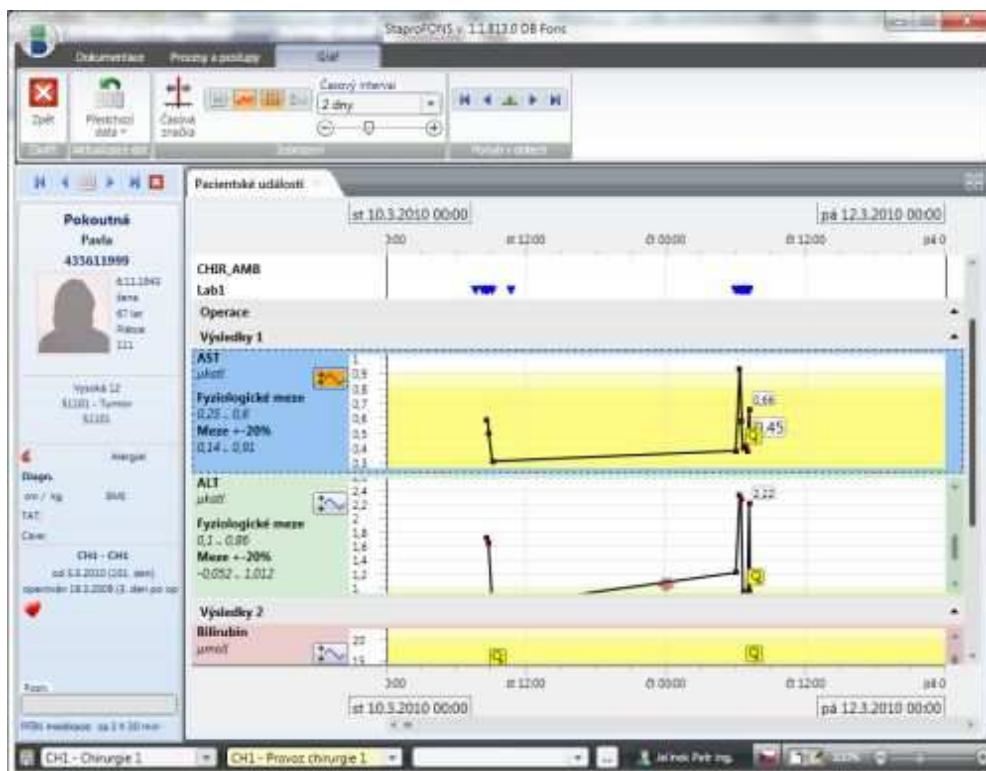
Základním prvkem klinické části FONS Enterprise je klinická událost, na které je založena dokumentace pacienta (anamnéza, příjmová zpráva, ambulantní vyšetření, žádanka na různé druhy vyšetření, ošetrovatelská anamnéza, denní dekurz, operační protokol, atd.).

Mezi obecné vlastnosti klinické události patří vysoký stupeň konfigurovatelnosti, který umožňuje nastavit dokumentaci dle individuálních potřeb. Od velmi jednoduchých dokumentů skládajících se pouze z editoru pro zápis volného textu, po složité strukturované formuláře. Každá klinická událost se skládá z několika oddílů: hlavička, diagnózy, požadavek (tato část je důležitá pro žádanky na různá vyšetření), nález/výsledek (do které uživatel zapisuje nález na vyšetření, výsledek, popisuje stav pacienta atd.), souhrn (klinická událost automaticky generuje ze zapsaných údajů shrnutí pro potřeby různých přehledů a seznamů).

NIS FONS Enterprise umožňuje uživateli otevřít a prohlížet nebo i zapisovat pro vybraného pacienta více klinických událostí současně. Uživatel tak může např. popisovat stav pacienta a současně zadávat žádanky na potřebná vyšetření, zapisovat recept nebo si prohlížet poslední laboratorní výsledky.



Při práci s pacientem je součástí obrazovky i tzv. patientský panel, ve kterém jsou přehledně zobrazeny základní údaje o pacientovi včetně poslední alergie, diagnóz, poslední medikace, informací o operaci, ... Uživatel je na první pohled informován o důležitých faktorech souvisejících se zdravotním stavem pacienta. Přímo z patientského panelu lze otevřít příslušný formulář pro zadání/editaci údaje (diagnózy, krevní skupiny, alergie atd.).



Obr. 7FONS Enterprise [14]

FONS Enterprise umožňuje data zobrazovat na časové ose. Tímto způsobem je možné přehledně graficky zobrazit měřené hodnoty, vývoj laboratorních výsledků pacienta, ordinované léky. Na časovou osu lze promítnout i jednotlivé ambulantní návštěvy, pobyty na lůžkových odděleních, prodělané operace atd. Grafické znázornění na časové ose lze doplnit i číselným a textovým popisem zobrazených dat. Uživatel může měnit časový interval a způsob zobrazení tak, aby pohled na data byl co nejpřehlednější.

## 2.5 Shrnutí

Tyto SW jsou jistě velkým pomocníkem sociálních zařízení. Jak je vidět na dvou výše uvedených případech, nabízí komplexní správu jak zdravotnické a ošetrovatelské dokumentace, tak i komplexní správu zaměstnanců, majetku, či stravování. Je možné

sledovat vykonané úkony, které se přepočtou na body a odešlou pojišťovně k plnění. Pokud bychom se zaměřili na efektivnost elektronické dokumentace, jistě najdeme nesčetně výhod proč takový systém používat. Pokud bychom ale vzali modelovou situaci, kdy nastane nečekaná krizová situace a je potřeba v co nejrychlejší době zobrazit všechny důležité informace o klientovi, které se týkají přímé souvislosti s ohrožením zdraví a života, jako jsou například léky, které bere, nejvážnější choroby, diety, alergie, jeho pohyblivost, orientaci psychický stav a další podobné informace, není vhodné prohledávat všechny tyto karty s obrovským množstvím informací. Jednoduše řečeno se jedná o informace, které mohou usnadnit zásah záchranářů, popřípadě urychlit. Čas, to je veličina, která v takové situaci hraje roli otázky života a smrti. Je tedy vhodné mít tyto informace uložené po několika kategoriích? Existuje nějaká funkce, která personálu v případě potřeby takové informace zobrazí z databáze na jedné obrazovce? Jak dlouho musím pracovat s aplikací, abych byl schopen tyto informace zobrazit? Na otázky lze odpovědět: takové funkce existují, nicméně není vhodné je použít v případě, když není čas hledat. Tyto zařízení mají pro takové případy svůj vlastní systém kde tyto informace vyhledat, například souhrny léků v lékárnice atd. Toto je funkce, která dle mého názoru na takových systémech chybí. Informace jsou zde ve velkém množství, proto není jednoduché se v nich rychle zorientovat v případě nouze.

Dalším trendem je vstup soukromých subjektů do oboru sociální péče. Reformy měnící zdravotnictví stále více ubírají finančních hodnot zařízením a ty jsou pak nuceny svůj podíl měnit ze státního na soukromé. Obecný problém financí v tomto sektoru má za následek úbytek kvalifikovaného personálu. Nekvalifikovaný personál v přímé obslužné péči nesmí vidět dle legislativy zdravotnickou dokumentaci, tudíž ani v elektronické podobě. Elektronická dokumentace je striktně uzamčená pod účtem a není tak žádná reálná možnost do ní nahlédnout. Z jedné strany je to v pořádku, že tyto informace nejsou přístupné neoprávněné osobě, na straně druhé vzniká otázka, co když nastane situace, že i tento personál bude nucen nahlédnout do takové dokumentace? Jelikož dochází již několik let k redukci kvalifikovaného personálu v takových zařízeních, není vzácné najít v sociálním zařízení směnu, kdy například na tři patra pečovatelského domu je pouze jedna sestra, která má pravomoc nahlížet do zdravotnické dokumentace. Pokud nastává nějaká krizová situace, kdy je potřeba rychle nahlédnout do dokumentace, je potřeba zavolat sestru, která odemkne ošetřovnu s přístupem do této dokumentace. A personál se až pak

dozví důležité informace o klientově stavu. Stejný problém vzniká i s elektronickou dokumentací.

## 2.6 Vlastní návrh

Po identifikaci nedostatku funkce rychlého zobrazení důležitých informací v co nejkratší době, jsem se rozhodl sestavit aplikaci, která bude tuto funkci mít. Pokud se rozhlédnete po trhu, zjistíte, že tvořit aplikaci na komplexní správu dokumentace nemá smysl, protože již dnes větší část sociálních zařízení běží na nějakém současném systému. Není tedy primárním úkolem se snažit tvořit aplikaci pro správu co největšího počtu informací z dokumentace. Úkolem je vytvořit aplikaci, která bude obsahovat pouze určitý souhrn, nebo výtah ze zdravotní a ošetřovatelské dokumentace, a udělat tak jednoduchý přehled pro situace, kdy je potřeba tyto informace zobrazit v co nejkratším čase a v co nejstručnější formě. Aplikace nebude sloužit jako náhrada klasické dokumentace, ani jako aplikace vhodná pro používání pro převod klasické dokumentace do elektronické. Spíše se dá klasifikovat, jako aplikace pro zvýšení bezpečnostní úrovně péče.

Představme si modelovou situaci. Probíhá noční směna, na objektu je pouze jedna sestra, na patrech je personál v přímé obslužné péči. Některý z klientů stiskne nouzové tlačítko a po příchodu personálu se zjistí, že klient se dusí. Okamžitě se zavolá zdravotnická záchranná služba a samozřejmě sestra z objektu. První by na místě měla být sestra. Postup je takový že sestra přijde, odemkne se sesterská místnost a začnou se zjišťovat důležité informace. Vyčtou se léky, alergie atd. Při příjezdu ZZS by měli být informace k dispozici. Tento postup je jistým časovým zdržením.

Pokud bychom vzali tuto modelovou situaci a provedli ji ještě jednou, ale tentokrát s přítomností mnou navrhované aplikace byl by postup od zavolání RZP následující. Personál se přihlásí do mé aplikace a do 1 minuty by měl být schopen zobrazit všechny tyto důležité informace a vytisknout je, popřípadě tyto informace podat v telefonickém rozhovoru s dispečinkem ZZS. Při příchodu sestry z jiného patra už bude tento personál mít v ruce vytištěné základní informace, od kterých se lze nějakým způsobem odrazit při řešení této krizové situace. Sestra se na základě informací může rozhodnout, jak tuto situaci řešit, popřípadě při příjezdu ZZS už bude jasnější, jak lze s klientem nakládat. Úplnost a pravdivost vyplněných informací by měla být dána politikou společnosti, aby nemohla nastat situace, že zadané informace nejsou pravdivé. Co se týče rychlosti a přehlednosti zobrazení informací, musí být samozřejmě co nejjednodušší a nejrychlejší.

V databázi budou uložena citlivá data, proto je nutné vytvořit určité oprávnění pro přístup do aplikace. S tím souvisí i určitá správa uživatelů.

#### Uživatelská část:

Uživatelská část je potřebná hned z několika důvodů. Je potřeba zajistit, aby přístup do aplikace neměl každý. Kdo má přístup do aplikace, musí mít určité oprávnění a nedílnou součástí tohoto oprávnění je také možnost sledovat a pojmenovat jednoznačně osobu, která provede příslušnou operaci. Uživatelská oprávnění použité v aplikaci:

- Administrátor
- Zdravotnický personál
- Ostatní oprávnění

Administrátor by měla být důvěryhodná osoba, která bude mít přístup do celé aplikace. Měla by to být staniční sestra případně další ze zaměstnanců z řad zdravotnického personálu pečovatelského domu. Bude mít na starost tvorbu nového uživatelského účtu, změny uživatelských profilů a správu historie. Samozřejmě bude mít přístup i do klientské části a návštěvní knihy, která je popsána na konci této kapitoly.

Zdravotnický personál bude mít svou hlavní úlohu v klientské části. Tvorba, editace a mazání klientských karet. Samozřejmě že i zobrazení karty bude k dispozici.

Personál s označením „Ostatní“ bude mít nejnižší oprávnění a to správu návštěvní knihy, jejíž účel popíšu v samostatné kapitole.

Součástí uživatelské části, jak již bylo řečeno, bude hlavně správa uživatelů a to přidání, editace, mazání.

#### Klientská část:

Klientská část – informace které aplikace bude spravovat, byla konzultována se dvěma osobami, pohybující se v této problematice, konkrétně se staniční sestrou sociálního zařízení a studentkou oboru sociální péče. Momentálně je nastaveno pár základních informací, které při nečekané události mohou ušetřit čas při záchraně.

Jedná se o tyto oblasti:

#### VŠEOBECNÉ INFORMACE

Jméno a příjmení.....

Datum narození.....Věk.....

## FYZIOLOGIE

Výška.....

Hmotnost.....

## VNÍMÁNÍ ZDRAVÍ

Celková úroveň zdraví (nemocnost, vleklá choroba)

.....

Alergie a diety (léky, pyly, potraviny apod.)

.....

## ÚROVEŇ SEBEPÉČE

Celková pohyblivost            Mobilní - imobilní

Potřeba speciálních pomůcek?

## VNÍMÁNÍ A POZNÁVÁNÍ

Potíže se sluchem    ano    ne

kompenzační pomůcka    ano    ne

Potíže se zrakem    ano    ne

Pomůcka: .....

Poruchy řeči            řeč je srozumitelná            řeč je nesrozumitelná

## Orientace

osobou    ano    ne

časem    ano    ne

místem    ano    ne

situací    ano    ne

## LÉKY

Vypište veškeré léky, které užívá (název, kolik, kolikrát, jakým způsobem, v kolik hodin):

1) .....

LÉKAŘI (jméno a příjmení, adresa ordinace, telefon):

Obvodní lékař.....

Psychiatr.....

Psycholog.....

Neurolog.....

Oční.....

Gynekolog.....

Zdravotní pojišťovna.....

Toto je základní návrh klientské „karty“. Informace by měl vyplňovat pouze zdravotnický personál. Navíc by měla být ošetřena vlastnost, aby nemohl personál editovat klientskou kartu klienta z jiného patra než je on sám. Přístup do této části aplikace je samozřejmě kontrolován.

#### Návštěvní kniha:

Návštěvní kniha je zařazena do aplikace z důvodu monitorování návštěv v objektu. Evidence návštěv by měla být stanovena politikou společnosti z bezpečnostních důvodů, jelikož většina zařízení, pro něž je tato aplikace vhodná k použití, nesmí zakazovat volný vstup osob do objektu. Tato zařízení musím mít zajištěna 24 hodinový režim, avšak v nočních hodinách je personálu v zařízení o poznání méně než při běžných pracovních hodinách. Nachází se zde velké množství léčiv a tak se takové zařízení může stát snadným cílem narkomanů a podobných typů lidí. Další hrozbou u takového typu zařízení je, že klientské pokoje jsou otevřené i přes noc. Spící senioři se tak mohou stát snadným terčem útoku pachatelů. Pokud se v objektu bude pohybovat taková osoba, vzniká tu jisté riziko i pro personál. Proto je do aplikace zařazena i tato funkce, kdy personál může evidovat pohyb osob po objektu bez větších investic do přístupového systému. Návštěva se ohlásí pracovníkovi na recepci, ten jí zapíše do aplikace a při odchodu návštěvy se odepíše jako neaktivní. Personál tak může po návštěvních hodinách zkontrolovat stav aktivních návštěv v objektu. Pokud by se stalo, že nějaká návštěva poruší návštěvní řád, je možné jí uložit do databáze blacklist, a při příští návštěvě bude na tohoto návštěvníka upozorněno jako na osobu v blacklistu. Kontrola údajů, které návštěvník sdělí na vrátnici se skutečností je

téměř nemožná. Proto jsem zařadil systém, kterým lze efektivně a hlavně zákoně ověřit, zda se návštěvník do objektu chystá s čistým úmyslem. Samozřejmě to není 100% ověřitelné, ale jako jistý kontrolní prvek bude sloužit. Jednoduchost spočívá v tom, že pokud se návštěva zapisuje, je nutné vyplnit klienta, za kterým se návštěva chystá. Ale aby bylo možné vyplnit klienta, musí se vybrat patro, na kterém se tento klient nachází. Dřív se jména klientů nezobrazí. Tudíž návštěvník musí znát nejen jméno, ale i patro klienta, za kterým se chystá. Tento nástroj se mě zdá dostačující pro ověření návštěvy a přitom je zcela legální. Přístup k této části aplikaci by měli mít všichni přihlášení uživatelé.

## **II. PRAKTICKÁ ČÁST**



### 3 TECHNICKÁ DOKUMENTACE APLIKACE

#### 3.1 Vývojové prostředky

Pro vývoj aplikace byl použit Golden html editor [15]

Pro testování a databáze byl použit Uniform Server3\_5.

Databázový systém v uniform serveru je phpMyAdmin - 2.10.2 mysql

#### 3.2 PHP

PHP (rekurzivní zkratka PHP: Hypertext Preprocessor, „PHP: Hypertextový preprocesor“, původně Personal Home Page) je skriptovací programovací jazyk, určený především pro programování dynamických internetových stránek. Nejčastěji se začleňuje přímo do struktury jazyka HTML, XHTML, což lze využít při tvorbě webových aplikací. PHP lze použít i k tvorbě konzolových a desktopových aplikací

#### 3.3 Databáze

Pro účely aplikace byla vytvořena databáze „diplomka“, ve které se nacházejí všechny data. Dodržení názvosloví je důležité pro správný chod aplikace. Pokud by aplikace byla nasazena do ostrého provozu, je možné jí exportovat, popřípadě vytvořit vlastní s podmínkou, že by se musel upravit aplikační kód.

##### 3.3.1 Struktura databáze

Sloupec	Typ
<u>id_al</u>	int(10)
klient	varchar(30)
typ	varchar(10)
poznámka	varchar(40)
info	varchar(40)

Obr. 9 Databáze - Alergie

Sloupec	Typ
jmeno	varchar(40)
prijmeni	varchar(40)
<u>id_bl</u>	int(11)
cas	datetime
pridal	int(10)
klient	int(11)

Obr. 8 Databáze - blacklist

Sloupec	Typ
<u>id_lkl</u>	int(10)
klient	int(10)
lek	int(10)
poznamka	varchar(40)

Obr. 11 Databáze – branileku

Sloupec	Typ
klient	int(10)
vyska	int(10)
vaha	int(10)

Obr. 12 Databáze - fyzio

Sloupec	Typ
<u>id_sl</u>	int(10)
skupina	varchar(40)

Obr. 10 Databáze – druhyleku

Sloupec	Typ
<u>id_hk</u>	int(10)
klient	varchar(20)
provedl	varchar(20)
operace	int(10)
cas	datetime

Obr. 13 Databáze - historiek

Sloupec	Typ
<u>id_hl</u>	int(10)
operace	varchar(20)
lek	varchar(20)
provedl	varchar(20)

Obr. 15 Databáze - historieky

Sloupec	Typ
<u>id_hi</u>	int(20)
operace	int(20)
provedl	varchar(10)
uzivatel	varchar(10)

Obr. 14 Databáze – historieu

Sloupec	Typ
<u>id_kt</u>	int(10)
jmeno	varchar(30)
prijmeni	varchar(30)
adresa	varchar(40)
kontakt	varchar(20)
ordinace	varchar(20)
klient	varchar(10)
odbornik	int(11)

Obr. 17 Databáze - kontakt

Sloupec	Typ
<u>id_hn</u>	int(10)
JMENO	varchar(20)
PRIJMENI	varchar(20)
klient	int(10)
casprichodu	datetime
casodchodu	datetime

Obr. 16 Databáze - navstevy

Sloupec	Typ
<u>id_lk</u>	int(11)
nazev	varchar(50)
skupina	int(20)

Obr. 19 Databáze - léky

Sloupec	Typ
klient	int(10)
nemoc	varchar(30)
poznamka	varchar(30)
<u>id_nm</u>	int(11)

Obr. 18 Databáze nemoci

Sloupec	Typ
<u>id_od</u>	int(10)
nazev	varchar(20)

Obr. 20 Databáze - odbornici

Sloupec	Typ
<u>id_op</u>	int(11)
nazev	varchar(20)

Obr. 21 Databáze - operace

Sloupec	Typ
JMENO	varchar(40)
PRIJMENI	varchar(40)
klient	varchar(20)
cas	datetime
pustil	varchar(20)
<u>id_nv</u>	int(11)

Obr. 23 Databáze - osobavstup

Sloupec	Typ
klient	int(10)
mobilni	tinyint(4)
imobilni	tinyint(4)
poznamka	text

Obr. 22 Databáze - pohyb

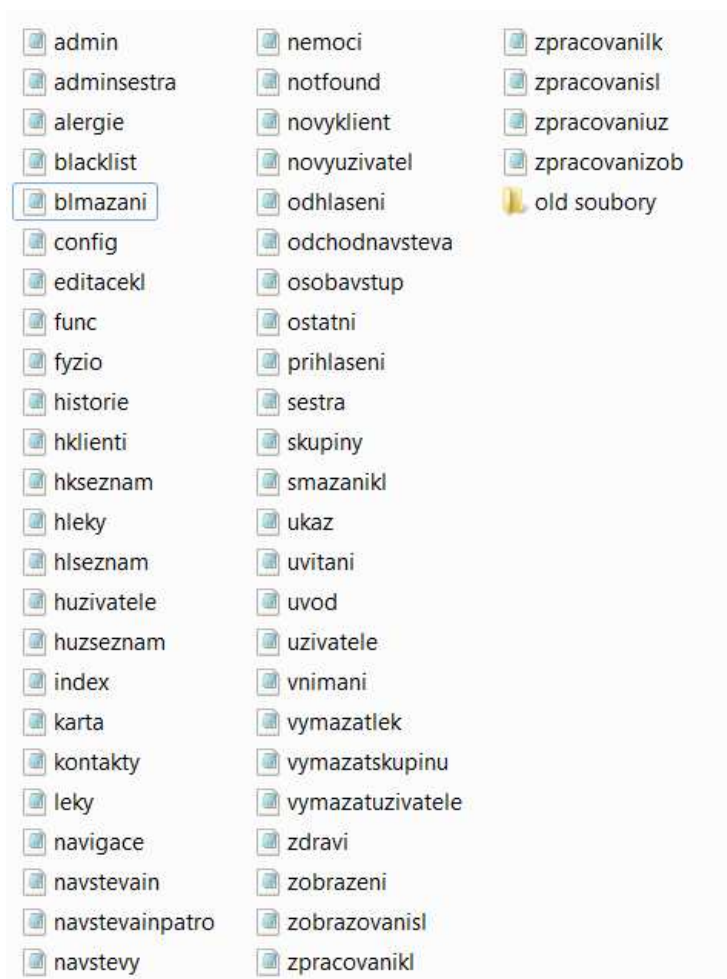
Sloupec	Typ
<b>klient</b>	int(10)
<b>sluch</b>	tinyint(4)
<b>pomucka</b>	tinyint(4)
<b>poznamka</b>	text
<b>zrak</b>	tinyint(4)
<b>bryle</b>	tinyint(4)
<b>rec</b>	tinyint(4)
<b>orientace_os</b>	tinyint(4)
<b>orientace_cas</b>	tinyint(4)
<b>orientace_ms</b>	tinyint(4)
<b>orientace_sit</b>	tinyint(4)

Obr. 25 Databáze - poznávání

Sloupec	Typ
<b><u>id</u></b>	int(11)
<b>prezdivka</b>	varchar(50)
<b>jmeno</b>	varchar(50)
<b>prijmeni</b>	varchar(50)
<b>patro</b>	int(11)
<b>heslo</b>	varchar(50)
<b>jeadmin</b>	tinyint(4)
<b>jesestra</b>	tinyint(4)
<b>jeostatni</b>	tinyint(4)

Obr. 24 databáze - uživatelé

### 3.4 Kořenový adresář



#### Kořenový adresář

Kompletní kódy souborů jsou obsaženy v příloze. Části kódů a funkce jsou popsány v jednotlivých kapitolách.

### 3.5 Index.php

Po spuštění serveru a zadání požadovaného adresáře je volán právě tento soubor. Je to základní spouštěcí soubor celé aplikace. Volán je soubor funkc.php. Ukládá se id přihlášeného uživatele do `$_SESSION["id"]`.

```

<?
    ob_start();
    session_start();
    require "../func.php";
    if ($_GET["clanek"]=="prihlaseni")
    {
        $id = iduzivatele($_POST["prezdivka"], $_POST["heslo"]);
        if ($id<>0) $_SESSION["id"]=$id;
    }

    elseif ($_GET["clanek"]=="odhlaseni")
    {
        unset($_SESSION["id"]);
    }
    |
?>

```

Obr. 26 Uložení \$\_SESSION["id"].

Obsahuje uspořádání celé grafiky, do levého dolního okna načítá soubor navigace.php, který obsahuje jednotlivé nabídky menu pro uživatele (viz. níže). Do pravého dolního okna načítá funkci ukazclanek(), o které je napsáno v kapitole o souboru funkc.php

```

<div align="center">
  <table width="760" cellpadding="0" cellspacing="0" border="1">
    <tr>
      <td width="760" height="120" colspan="2"></td>
    </tr>
    <tr>
      <td width="160" valign="top"><?require "../navigace.php"?></td>
      <td width="600" valign="top"><?ukazclanek()?></td>
    </tr>
  </table>

```

Obr.27 Grafika aplikace

### 3.6 Funkc.php

Soubor funkc.php obsahuje všechny důležité funkce pro chod aplikace. Je volán do souboru index.php, takže je k dispozici všem souborům spuštěným během práce s aplikací. Základní funkcí je připojení k databázi. Údaje pro přihlášení jsou uloženy v souboru config.php, který je zavolán na začátku souboru. Následuje skript pro přihlášení do databáze.

```

include "config.php";

$GLOBALS["link"]=mysql_connect(SQL_HOST, SQL_USERNAME, SQL_PASSWORD);
mysql_select_db(SQL_DBNAME);

```

Obr. 28 Připojení k DB

Dalším skriptem je funkce ukazclanek(), který má za úkol udržet aplikační logiku. Odkaz ve tvaru `index.php?clanek=uvod` hledá nejprve soubor `uvod.html` a pak soubor `uvod.php` v kořenové adresáři. Díky zavolání funkce v souboru `index.php` se hledaný soubor otevře v pravém dolním okně. Pokud se soubor v kořenovém adresáři nenajde, zavolá se soubor `notfound.php`, který uživateli sdělí chybu při hledání požadované stránky.

Dalšími důležitými skripty jsou funkce `jeadmin()`, `jesestra()`, `jeostatni()`, `jeboji()`. Slouží pro identifikaci oprávnění uživatele, kdy se z databáze načte informace o uživatelově účtu na základě `$_SESSION["id"]` čísla, a načte se jeho oprávnění. S oprávněním se pak dále pracuje v jednotlivých souborech.

```
{
    if (!isset($_SESSION["id"])) return false;
    $vysledek=mysql_query("select id from uzivatele where id=".$_SESSION["id"]." and jeadmin=1", $GLOBALS["link"]);
    return (boolean) mysql_num_rows($vysledek);
}

function jesestra ()
{
    if (!isset($_SESSION["id"])) return false;
    $vysledek=mysql_query("select id from uzivatele where id=".$_SESSION["id"]." and jesestra=1", $GLOBALS["link"]);
    return (boolean) mysql_num_rows($vysledek);
}
```

Obr. 29 Funkce pro oprávnění

Dalším skriptem jsou funkce pro ověřování duplicity dat v databázi. Pokud je potřeba během funkce aplikace ověřit existenci dat v databázi, volají se právě tyto funkce. Jsou to: klient, přezdívká, skupina léků, existence zápisu v tabulce braní léků vzhledem ke klientovi a léku, lék v databázi, nemoc a alergie v databázi.

```
function alergievdb ($nlek, $odesilatel )
{
    $vysledek=mysql_query("select info, klient from alergie where klient='".$odesilatel."'and info='".$nlek.'",
    return (boolean) mysql_num_rows($vysledek);
}
```

Obr. 30 Ověření existence dat v databázi

### 3.7 Navigace.php

Soubor `navigace.php` je volán v souboru `index.php` do dolního levého okna. Slouží k zobrazení nabídky menu. Je zapotřebí, aby se menu dynamicky měnilo, například podle toho, zda je někdo přihlášen (`$_SESSION["id"]`), v tom případě se menu zobrazí, v opačném případě se vypíše odkaz na přihlašovací skript.

```
<p><a href="index.php?clanek=uvod">Uvod</a></p>
<?if (isset($_SESSION["id"])):??>
<p><a href="index.php?clanek=odhlaseni">Odlasit</a></p>

<?else:??>
<p><a href="index.php?clanek=prihlaseni">Prihlasit</a></p>
<?endif??>
```

*Obr. 31 Dynamické zobrazení menu*

Další možností dynamického zobrazení je výpis menu podle oprávnění přihlášeného uživatele. Do souboru `navigace.php` jsou volány soubory `admin.php`, `sestra.php`, `ostatni.php` a `adminsestra.php`. Obsahem těchto souborů jsou odkazy na jednotlivé soubory, odkazující na funkce přiřazené právě dle typu oprávnění. Na obrázku níže je vidět pro příklad obsah souboru `admin.php`

```
?
  if (!jeadmin()) return;
echo "<a href=\"index.php?clanek=uzivatele\">Sprava uzivatelu</a><br>";
echo "<a href=\"index.php?clanek=vymazatuzivatele\">Vymazat uzivatele</a><br>";
echo "<a href=\"index.php?clanek=novyuzivatel\">Novy uzivatel</a><br>";
echo "<a href=\"index.php?clanek=historie\">Historie</a><br>";
echo "<a href=\"index.php?clanek=osobavstup\">Vstup navstevy</a><br>";
echo "<a href=\"index.php?clanek=navstevainpatro\">Aktivni navstevy</a><br>";
echo "<a href=\"index.php?clanek=odchodnavsteva\">Odchod navstevy</a><br>";
echo "<a href=\"index.php?clanek=blacklist\">Black list</a><br>";
echo "<a href=\"index.php?clanek=navstevy\">Probehle navstevy</a><br>";

>
```

*Obr. 32 Admin.php*

Je vidět, že se administrátorovi zobrazují jen funkce týkající se správy uživatelů a návštěvní knihy. Administrátor má samozřejmě přístup i k funkcím pro lékařskou část personálu, ale jelikož tyto funkce má k dispozici i oprávnění „sestra“, tak jsou uloženy v samostatném souboru `adminsestra.php`



## 3.8 Správa uživatelů

### 3.8.1 Nový uživatel

První z funkcí starající se o správu uživatelů je sada skriptů starající se o přidání nového uživatele uložených v souboru `novyuzivatel.php`. Odkaz na soubor se skripty je dostupný pouze administrátorovi z menu pod názvem „nový uživatel“. Vzhled strany a ovládání je popsáno v uživatelské příručce. Pro technické účely nás bude zajímat pouze struktura stránky a jednotlivé skripty.

Základní funkcí je zobrazení této stránky pouze autorizované osobě a to administrátorovi. K tomu poslouží podmínka funkce `if (!jeadmin()) return`, která zobrazí stránku pouze uživateli s oprávněním `administrátor`. Tímto omezujeme riziko, že uživatel napíše url ručně a ve chvíli kdy vypíše url typu „`index.php?clanek=novyuzivatel`“ například uživatel s nejnižším oprávněním, i přesto by se mu stránka zobrazila, kdyby nezačínala právě touto podmínkou.

Téměř všechny stránky jsou konstruovány tak, že nejprve by se měl zobrazit obsah stránky a na základě uživatelsky zvolených funkcí se odesílá tento obsah do databáze, popřípadě se po odeslání zpět vrací uživateli obsah z databáze k zobrazení. K tomu posloužila stavba stránek, které jsem provedl na následujících pravidlech. Vytvořil jsem si proměnou `$BudemeZobrazovat` a do té ukládám hodnoty „1 a 0“. Pokud je nastavená hodnota 1 zobrazuje se grafika souboru, jako formuláře, tabulky atd. Po odeslání formuláře se většinou aplikace vrací na danou stránku, aby ověřila správnost formulářů a zadaných dat. Pokud jsou data v nepořádku, aplikace vypisuje chybovou hlášku a zůstává na stránce, aby mohl uživatel chyby opravit. Pokud jsou zadaná data v pořádku, nastavuje se pro proměnou `$BudemeZobrazovat=false` a provádí se zápis/čtení z databáze.

```

if (!jeadmin()) return; //kontrola oprávnění

$BudemeZobrazovat=true; //nastavení zobrazení

if (!empty($_POST))
//kontrolní body... ověření formulářů atd...

.
. //další kontrolní body
.

else
{

    $BudemeZobrazovat=false;
    //vynulování zobrazení pak se nezobrazují formuláře ale provádí se skript
    //vlození do databáze..... zpracování dotazu
}

if ($BudemeZobrazovat):?>
//pokud bude zobrazení true - busou se zobrazovat formuláře

//zde budou nějaké formuláře atd.....

```

Obr. 33 Zobrazovací funkce

Pro stránku přidávající uživatele je využito skriptu, který kontroluje proměnou \$BudemeZobrazovat a podle toho buďto zobrazuje stránku s formuláři anebo ukládá do databáze. Stránka se skládá z 3 typů formulářů a to z rozevíracího, zaškrťovacího a textového pole. Rozevírací formulář slouží pro výběr patra, do kterého bude uživatel přiřazen. Je plněn z funkce prikazy()

```

function Prikazy()
{
    $prikazy=Array("1", "2", "3");
    sort ($prikazy);
    echo "<SELECT name=\"patro\">";
    foreach ($prikazy as $patro) echo "\t<OPTION VALUE=\".str_replace('_', '-', $patro).>". $patro.
    echo "</SELECT>";
}

```

Obr. 34 Nový uživatel – funkce příkazy()

Zaškrťovací formulář slouží pro výběr oprávnění a jeho správnost je kontrolována počtem zaškrtnutých políček. Nesmí být uživatel, který má 2 oprávnění. Proto je kontrolováno, aby zaškrtnuto bylo opravdu jen jedno políčko, a naopak každý uživatel musí mít 1 oprávnění, takže není možné odeslat formulář prázdný.

```
elseif ((!empty($_POST["jeadmin"]) and !empty($_POST["jesestra"])) || (!empty($_POST["jeneopraven"])
|and !empty($_POST["jesestra"])) || (!empty($_POST["jeadmin"]) and !empty($_POST["jeneopraven"])))
{
echo "Oznacena musi byt pouze 1 volba";
}

elseif (empty($_POST["jeadmin"]) and empty($_POST["jesestra"]) and empty($_POST["jeneopraven"]))
{
echo "Musi byt oznacena 1 volba";
}
```

*Obr. 35 Nový uživatel – ověření zaškrťovacího formuláře*

Textové formuláře jsou hlídány jednak na počet znaků, většinou 3-15, tak na speciální znaky, aby uživatel nemohl zadávat různé příkazy. V takovém případě se tato proměnná ihned vymaže. Dále je u proměnné přezdívka kontrolována duplicita v databázi pomocí funkce `prezdivkavdb()`

Zápis vložených a ověřených dat do databáze je realizován pomocí skriptu. Do databáze se zapíše nejen nový uživatel, ale i zápis do historie o vytvoření nového uživatele.

### **3.8.2 Smazání uživatele**

Sada skriptů sloužící pro vymazání uživatele z databáze je dostupná opět pouze administrátorovi s menu pod položkou „smazání uživatele“ odkazující na soubor `vymazatuzivatele.php`. Skript je postaven na zobrazení tabulky s uživateli z databáze, kdy v každém řádku bude jeden uživatel. V každém řádku bude odškrťvací formulář. Po odeslání se spustí skript na kontrolu odškrtnutých formulářů a načte se ID číslo každého uživatele, kde je formulář odškrtnutý. Spouští se pak skript na vymazání z databáze a uložení informací do historie.

```

<?
if($_POST['delete']){
//print_r($_POST);
$checkbox=$_POST['checkbox'];
//exit;
for($i=0;$i<count($checkbox);$i++){
$del_id = $checkbox[$i];
$sql="SELECT prezdivka FROM uzivatele WHERE id='$del_id'";

$result = mysql_query($sql);
$nacteni=mysql_fetch_array($result, MYSQL_ASSOC);

$sql = "insert into historieuz ( operace, provedl, uzivatel, cas ) VALUES
( '1', '$_SESSION[id].', '$_nacteni[prezdivka].', now() )";
$result = mysql_query($sql);
$sql = "DELETE FROM uzivatele WHERE id='$del_id'";
$result = mysql_query($sql);
}

```

*Obr. 36 Smazání uživatele - skript*

### 3.8.3 Editace uživatele

Skripty pro editaci uživatelů jsou uloženy v souborech uzivatele.php, který je dostupný pouze administrátorovy z menu pod odkazem „správa uživatelů“, ve kterém je skript pro zobrazení uživatelů dle vybraného patra a odkaz na soubor zpracovaniuz.php, kde je nadále vybraný klient zpracováván.

### 3.8.4 Uzivatele.php

Tento soubor obsahuje sadu skriptů, pro vybrání patra a následné zobrazení uživatelů v tabulce dle vybraného patra. Výběr patra je realizován opět rozevíracím formulářem a funkcí prikazy(), která ho plní. Výběr patra se realizuje odesláním tlačítkem zobrazit. Číslo patra se načte do proměnné \$\_POST["patro"] a provede se výběr z databáze s podmínkou patra, která je uložena v proměnné \$\_POST["patro"].

Uživatelé se vypíší do tabulky, každý uživatel do jednoho řádku. V každém řádku bude tlačítko „upravit uživatele“, na jehož kliknutí aplikace přesměruje uživatele na soubor zpracovaniuz.php. Spolu s přesměrováním se uloží do proměnné \$\_POST["odeslat"] ID uživatele, který byl vypsán na řádku, kde bylo stisknuto tlačítko „upravit uživatele“.

```

<tr>
  <td><div align="center"><?=$nacteni["prezdivka"];?></div></td>
  <td><div align="center"><?=$nacteni["jmeno"];?> <?=$nacteni["prijmeni"];?></div></td>
  <td><div align="center"><?=$nacteni["patro"];?></div></td>
  <td><div align="center"><?=$nacteni["jeadmin"];?></div></td>
  <td><div align="center"><?=$nacteni["jesestra"];?></div></td>
  <td><div align="center"><?=$nacteni["jeostatni"];?></div></td>
  <td> <form method="post" action="index.php?clanek=zpracovaniuz" >
    <input type="hidden" name="odeslat" value="<?=$nacteni["id"];?>" />
    <input type="submit" name="odeslano" value="Upravit uzivatele" />
  </form> </td>

```

Obr. 37 Upravit uživatele – přesměrování na `zpracovaniuz.php`

### 3.8.5 Zpracovaniuz.php

Odkaz na tento soubor zrealizuje kliknutí na tlačítko „upravit uživatele“ v souboru `uzivatele.php`. Spolu s přesměrováním se na tento soubor se uloží id uživatele do proměnné `$_POST["odeslat"]`. Tato proměnná se načte ihned na začátku skriptu souboru `zpracovaniuz.php`, respektive se ověří, zda je tato proměnná načtená a uloží se do proměnné `$odeslatel`. Provede se výběr klienta z databáze s podmínkou vyhledání podle id čísla uživatele, které je uloženo v proměnné `$odeslatel`. Stránka je udělána, aby bylo možné dynamicky měnit všechny informace a vidět okamžitou změnu. Jsou tedy zobrazeny všechny informace i uživateli a pod každou informací je připraven formulář s odesílacím tlačítkem pro změnu. Odesláním tlačítka se provede skript připravený pro jednotlivá tlačítka. Funkce je taková, že při odeslání formuláře se odešle přesměrování na tu samou stránku, čímž dojde ke znovu spuštění všech skriptů. Spolu s přesměrováním se uloží 2 proměnné a to odeslání tlačítka a proměnná ze změnového formuláře. Skripty pro jednotlivé změny jsou pak spouštěny na základě těchto 2 proměnných a to pomocí podmínek `if ($_POST["odeslano"]=="název tlačítka")` a `if (!empty($_POST["proměnná změnového formuláře"]))`.

Pod těmito podmínkami jsou již samotné kontroly odeslaných formulářů, jako jsou duplicity změnových dat a dat uložených v databázi, kontrola počtu znaků či kontrola speciálních znaků. Pokud jsou podmínky kontroly dat v pořádku, následuje přepsání dat v databázi. Dále je zde uložení či přepsání operace v databázi historie.

### 3.9 Historie

Historie je nedílnou součástí aplikace. Zde se administrátor může podívat, kdo vytvářel, mazal nebo upravoval nové uživatele, klienty, skupiny léků, a léky. Historie má 3 podkategorie:

1. Historie uživatelů aplikace
2. Historie klientů
3. Historie skupin léků a léků

Odkaz na jednotlivé typy historií jsou uloženy v souboru historie.php, který je dostupný z menu oprávněním administrátor a zdravotnický personál.

#### 3.9.1 Způsob ukládání historie při běhu aplikace

Způsob uložení do historie je jednoduchý. Ve své podstatě se dá říct, že jakékoliv uložení, aktualizace a smazání dat z databáze je doprovázeno skriptem pro uložení daného kroku do databáze jako historie.

Rozlišujeme 3 typy historie:

- 1) uživatelská
- 2) klientská
- 3) historie léků a skupin

Pro každý typ historie je přidělena samostatná tabulka. Pro identifikaci operace, která je prováděná slouží čísla, která jsou shodná s tabulkou „operace“.

id_op	nazev
1	mazani
2	pridani
3	zmena profilu
4	pridani leku
5	pridani skupiny
6	smazani leku
7	smazani skupiny
8	pridani klienta
9	smazani klienta
10	editace klienta

Obr. 38 Historie – operace

Pro každou prováděnou operaci v aplikaci, která ukládá, mění nebo maže data z databáze je přiřazeno číslo, které ukládám spolu se zápisem do databáze.

```
$sql = "insert into historiekl ( operace, provedl, klient, cas ) VALUES
( '8', '$provadi.', '$nacteni["id_kl"].', now() )";
$result = mysql_query($sql);
```

Obr. 39 Historie – zápis do databáze

### 3.9.2 Historie uživatelů aplikace

Na tuto stránku se dostane uživatel dle oprávnění přes menu – historie – historie uživatelů. V základě se jedná o skript, uložený v souboru huzivatele.php, zobrazující historii dle vybraného kritéria. V této sekci jsou na výběr kritéria – smazání uživatelé, přidání uživatelé a editace profilu. Vše je realizováno pomocí přepínačového formuláře, který obsahuje hodnoty pro jednotlivé operace. Po zvolení hodnoty a kliknutí na tlačítko “zobrazit“ se odešle formulář na tutéž stránku, kde je v první části skript pro výpis z databáze podle hledaného kritéria, v tomto případě se zadává číslo operace která je uložena v proměnné \$\_POST["operace"]. Spojením tabulek pomocí left join se zobrazí hledaná data, jako je jméno a příjmení editovaného uživatele, přezdívka uživatele provádějící operaci a čas provedení.

```
<form method="POST" action="index.php?clanek=historie">
<p><input type="radio" name="operace" value="1">Smazani uzivatele
<input type="radio" name="operace" value="2" >Pridany uzivatele
<input type="radio" name="operace" value="3">Editace
<input type="Submit" name="odesli" value="Zobrazit" ></td></p></td></p>
```

Obr 40 Historie uživatelé – výběr operace

```
$vyber = "SELECT historieuz.id_hi, historieuz.uzivatel, historieuz.cas,uzivatele.jmeno AS provedl
FROM historieuz
LEFT JOIN uzivatele ON historieuz.provedl = uzivatele.id
where operace='$_POST["operace"].'
ORDER BY `id_hi` desc";
$opakovanivyberu = mysql_query($vyber, $GLOBALS["link"]);
```

Obr.

41

### Historie uživatelé – výběr z DB

Na stránce je vložen odkaz, který odkáže uživatele na stránku huzseznam.php, kde je možné zobrazit všechny operace dohromady. Jen z důvodu možného většího počtu dat je zde omezující formulář, s výběrem měsíce, ve kterém byla operace vykonána. Formulář je realizován pomocí funkce Prikazy(). Vybraný měsíc se načte do proměnné

\$\_POST["mesic"] a samotný výběr je poté zrealizován pomocí podmínky where – between – and.

```
function Prikazy()
{
    $prikazy=Array("01", "02", "03","04", "05", "06", "07", "08", "09", "10", "11", "12");
    sort ($prikazy);
    echo "<SELECT name=\"patro\">";
    foreach ($prikazy as $mesic) echo "\t<OPTION VALUE=\".str_replace('_',\"-\",$mesic).\">.$mesic.</OP!";
    echo "</SELECT>";
}

$datum = StrFTime("%Y", Time());
echo "je rok $datum<br>";

$vyber = "SELECT historieuz.id_hi, historieuz.uzivatel, historieuz.cas,uzivatele.jmeno AS provedl, operace.1
FROM historieuz
left JOIN uzivatele ON historieuz.provedl = uzivatele.id
left JOIN operace ON historieuz.operace = operace.id op
where cas between '$datum-".$_POST["mesic"]."-01 00:00:00' and '$datum-".$_POST["mesic"]."-31 00:00:00'
ORDER BY `cas` desc";
```

Obr. 42 Historie uživatelů – huzseznam.php

### 3.9.3 Historie klientů

Historie klientů je řešena naprosto stejným způsobem jako historie uživatelů. Jen se mění názvy souborů a to – hklienti.php a hkseznam.php

### 3.9.4 Historie léky

Historie léky slouží pro zobrazení přidání skupiny léků, nebo přidání léku do skupiny. Skripty jsou vytvořeny naprosto stejným způsobem jako u předchozích dvou. Jediná změna je na úvodní obrazovce, kdy se nabídka operací mění na: přidání léku, přidání skupiny. Mazání skupin a léků je z důvodu možného nesouladu v aplikaci při vymazání zakázáno. Jde smazat pouze z databáze administrátorem.

```
<p><input type="radio" name="operace" value="4">Pridani leku
<input type="radio" name="operace" value="5">Pridani skupiny
<input type="Submit" name="odesli" value="Zobrazit " ></td></p></td></p>
```

Obr. 43 Historie léků – druhy operací

## 3.10 Návštěvní kniha

Návštěvní kniha je zařazena do aplikace z důvodu monitorování návštěv v objektu. Jednotlivé funkce – vstup návštěvy, odchod návštěvy, aktivní návštěvy, proběhlé návštěvy a blacklist.



### 3.10.1 Vstup návštěvy

Sada skriptů starající se o chod části aplikace, kde zapisuje vstup návštěvy je uložena v souboru osobavstup.php, který je zavolán odkazem z menu pod názvem „vstup návštěvy“. Je složen z několika formulářů. Prvním odeslaným formulářem by měl být výběr patra. Je realizovaný pomocí přepínačového formuláře, kdy se do proměnné \$\_POST["patro"] ukládá číslo patra. Následuje spuštění funkce naplnseznam(), která se stará o naplnění rozevíracího formuláře. Ten se naplní hodnotami z databáze a to jmény a příjmeními klientů podle zadaného patra. Id číslo vybraného klienta se uloží do proměnné \$\_POST["data"].

```
function naplnseznam2 ($Klienti, $zvoleno=0)
{
    $options="";
    $vysledek=mysql_query("select id_kl, jmeno, prijmeni from klienti where patro='".$_POST["patro"]."' order by jmeno",$GLOBALS["link"]);
    while ($zaznam=mysql_fetch_array($vysledek)):
        if ($zaznam["id_kl"]==$zvoleno)
            $options.="<option value='".$zaznam["id_kl"]."' selected>".$zaznam["jmeno"]." ".$zaznam["prijmeni"]."</option>\n";
        else
            $options.="<option value='".$zaznam["id_kl"]."'>".$zaznam["jmeno"]." ".$zaznam["prijmeni"]."</option>\n";
    endwhile;
    return $options;
}
```

Obr. 44 Vstup návštěvy – naplnění seznamu

Následují formuláře se jménem a příjmením vstupované osoby. Obě hodnoty jsou uloženy do proměnných a jsou zpracovávány pomocí skriptů. Jsou kontrolovány na obsah a počet znaků. Při špatném zadání opět aplikace vypíše chybu. Zpracování formulářů se děje při splnění podmínky naplnění hodnot.

```
elseif (uzivatelin ($_POST["prijmeni"], $_POST["jmeno"]))
{
    echo "Navstevnik se jmenem ".$_POST["jmeno"]." ".$_POST["prijmeni"]." je v evidenci aktivni navstevy";

    elseif ((strlen ($_POST["jmeno"])>15) | strlen ($_POST["jmeno"])<3)echo "Jmeno musi mit 3 - 15 zaku";
    elseif ((strlen ($_POST["prijmeni"])>15) | strlen ($_POST["prijmeni"])<3) echo "Prijmeni musi mit 3 - 15znaku";
    elseif (empty($_POST["data"])) echo "Musi byt zadan klient";
    else {

        mysql_query ("insert into osobavstup (jmeno, prijmeni, klient, cas, pustil) values
        ('".$_POST["jmeno"]."', '".$_POST["prijmeni"]."', '".$_POST["data"]."', now(), '".$_SESSION["id"]."',, $GLOBALS["link"]);
        echo "<meta http-equiv='refresh' content='0';URL=http://localhost:8000/diplomka/index.php?clanek=navstevainpatro\>";

    }
}
```

Obr. 45 Vstup návštěvy – zpracování formulářů

### 3.10.2 Odchod návštěvy

Skripty zpracování funkce odchod návštěvy jsou uloženy v souboru odchodnavsteva.php. V podstatě se jedná o jednoduchý skript, velmi podobný skriptu použitého v souboru vymazatuzivatele.php. Skriptem se zobrazí aktivní návštěvy z tabulky osoba vstup do tabulky, kde je v každém řádku jedna návštěva a zaškrťovací formulář.

```
$query="
SELECT osobavstup.id_nv, osobavstup.jmeno, osobavstup.prijmeni, osobavstup.cas, klienti.jmeno
klienti.prijmeni AS nazev, klienti.pokoj AS pokoj,
uzivatele.prezdivka AS uzivatel
FROM osobavstup
LEFT JOIN klienti ON osobavstup.klient = klienti.id_kl
LEFT JOIN uzivatele ON osobavstup.pustil = uzivatele.id";
$result=mysql_query($query);
```

Obr. 46 Odchod návštěvy – zobrazení návštěv

Po odškrtnutí vybraných návštěv se tlačítkem „odchod návštěvy“ nebo „black list“ odešlou data ke zpracování. Načítají se jednotlivá id čísla návštěv do proměnné \$\_POST['checkbox']. Dle vybraného tlačítka se vybírá skript pro uložení do databáze.

```
if($_POST['blacklist'])
{
    $checkbox=$_POST['checkbox'];
    for($i=0;$i<count($checkbox);$i++){
        $del_id = $checkbox[$i];

        $sql = " SELECT id_nv, jmeno, prijmeni, klient
                FROM osobavstup
                where id_nv= '$del_id'";
        $result = mysql_query($sql);

        $nacteni=mysql_fetch_array($result, MYSQL_ASSOC);
        $sql = "insert into blacklist (jmeno, prijmeni, cas, pridal, klient) VALUES
        ( '". $nacteni["jmeno"]."', '". $nacteni["prijmeni"]."', now(), '". $_SESSION["id"]."',
        '". $nacteni["klient"]."' )";
        $result = mysql_query($sql);

        $sql = "DELETE FROM osobavstup WHERE id_nv='$del_id'";
        $result = mysql_query($sql);
    }
}
```

Obr. 47 Odchod návštěvy – zpracování odeslaného formuláře

### 3.10.3 Aktivní návštěvy

Skript uložený v souboru navstevainpatro.php je dostupný oprávněnému uživateli z menu pod položkou „aktivní návštěvy“. Jedná se o skript, který zobrazí data z databáze podle vybraného patra. Patro se vybírá z rozevíracího formuláře, který je plněn z funkce

prikazy() a hodnota patra je uložena do proměnné \$\_POST["patro"]. Na základě této proměnné se zobrazí aktivní návštěvy z tabulky osobavstup.

```
function Prikazy()
{
    $prikazy=Array("1", "2", "3");
    sort ($prikazy);
    echo "<SELECT name=\\"patro\\">";
    foreach ($prikazy as $patro) echo "\t<OPTION VALUE=\".str_replace(\"_\", \"-\", $patro).\">";
    ". $patro."</OPTION>\n";
    echo "</SELECT>";
}

$vyber = "
SELECT osobavstup.jmeno, osobavstup.prijmeni, osobavstup.cas, klienti.jmeno AS skupina,
klienti.prijmeni AS nazev, klienti.pokoj AS pokoj
FROM osobavstup
inner JOIN klienti ON osobavstup.klient = klienti.id_kl
where patro='".$_POST["patro"]."';

$opakovanivyberu = mysql_query($vyber, $GLOBALS["link"]);
```

Obr. 48 Aktivní návštěvy – zobrazení

V těle skriptu se nachází odkaz na stránku navstevain.php, kde se zobrazí všechny aktivní návštěvy. Skript je postaven pouze na výpisu všech dat z databáze.

### 3.10.4 Proběhlé návštěvy

Skript dostupný z menu pod položkou „proběhlé návštěvy“ a uložený v souboru navstevy.php, je jednoduchým skriptem, kdy se pomocí rozevíracích formulářů zjišťuje od uživatele datum a čas příchodu návštěvy a datum a čas odchodu návštěvy. Formuláře jsou plněny jednotlivými funkcemi pro měsíc, den a hodinu.

```
function mes()
{
    $mes=Array("01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12");
    sort ($mes);
    echo "<SELECT name=\\"mesic\\">";
    foreach ($mes as $mesic) echo "\t<OPTION VALUE=\".str_replace(\"_\", \"-\", $mesic).\">". $mesic."</OPTION>\n";
    echo "</SELECT>";
}
```

Obr. 49 Proběhlé návštěvy – plnění formulářů

Ošetření pro chybné zadání formulářů – například měsíc příchodu by neměl být větší než měsíc odchodu v jednom kalendářním roce. Ošetření je pomocí jednoduchých funkcí, kdy aplikace kontroluje zadaná data a při větším měsíci příchodu než odchodu vyhodí chybovou hlášku. Pokud je měsíc stejný, kontroluje hodnotu dne – opět by neměla být hodnota dne příchodu větší než hodnota odchodu. Pokud je i den stejný kontroluje se stejným způsobem hodina.

```

if (($_POST["mesic"])>($_POST["dmesic"]))
echo "Mesic prichodu musi byt mensi nez mesic odchodu<br>";
if (($_POST["mesic"])==($_POST["dmesic"]))
{
if (($_POST["den"])>($_POST["dden"]))
echo "Den prichodu nemuze byt vetsi nez den odchodu pri stejnem mesici<br>";
}
if (($_POST["den"])==($_POST["dden"]))
{
if (($_POST["hodina"])>($_POST["dhodina"]))
echo "Hodina prichodu nemuze byt vetsi nez hodina odchodu pri stejnem dni<br>";
}
}

```

Obr. 50 Proběhlé návštěvy – kontrola zadaných dat

Poté už následuje jen jednoduchý skript pro vypsaní dat z databáze pomocí zadaných dat z formuláře. Data jsou uložena v proměnných \$\_POST["(proměnná dle typu.... Den, měsíc hodina atd...)"] a z databáze se vypisují dle podmínky „where čas příchodu between něco and něco“.

```

$vyber = "
SELECT navstevy.id_hn, navstevy.jmeno, navstevy.prijmeni, navstevy.casprichodu, navstevy.casodchodu,
klienti.jmeno AS krestni, klienti.prijmeni AS prijmeniklient
FROM navstevy
inner JOIN klienti ON navstevy.klient = klienti.id_kl
where casprichodu between '$datum-".$_POST["mesic"]."-".$_POST["den"]." ".$_POST["hodina"].":00:00'
and '$datum-".$_POST["dmesic"]."-".$_POST["dden"]." ".$_POST["dhodina"].":59:59'
ORDER BY `casprichodu` desc";

```

Obr. 51 Proběhlé návštěvy – vypsaní z databáze

### 3.11 Klientská část aplikace

Klientskou část tvoří sada souborů se skripty, obdobné jako u uživatelské správy. Jsou tu funkce přidávání, editace a zobrazení.

#### 3.11.1 Nový klient

Funkce dostupná administrátorovi a lékařskému personálu. Sada formulářů pro zadání jména, příjmení, pojišťovny, data narození, patra a pokoje. Všechny formuláře kromě výběru patra jsou textové, jediný formulář pro výběr patra je rozevírací. Rozevírací formulář je opět plněn funkcí prikazy() a patro se ukládá do proměnné \$\_POST["patro"]. Je zde další funkce kontrol(), která převede proměnou \$\_POST["pokoj"] z formuláře „pokoj“ na řetězec a uloží si do proměnné \$kontrol první číslici ze zadaného patra a po té jí porovná s první číslicí patra. Pokud nesouhlasí patro s první číslicí pokoje, aplikace vyhazuje hlášku o chybném zadání pokoje. Kontrola ostatních formulářů je prováděna klasickým způsobem jako u předchozích funkcí.

```

$kontrol=substr($_POST["pokoj"],0, 1);
$_POST["jmeno"]=strip_tags($_POST["jmeno"]);
$_POST["prijmeni"]=strip_tags($_POST["prijmeni"]);
$_POST["pojistovna"]=strip_tags($_POST["pojistovna"]);
$_POST["pokoj"]=strip_tags($_POST["pokoj"]);

$BudemeZobrazovat=true;
if (!empty($_POST["odesli"]))

if (kldvb ($_POST["jmeno"], $_POST["prijmeni"]))
{
echo "Klient se jmenem ".$_POST["jmeno"]." ".$_POST["prijmeni"]." je jiz zareg:
}
|
elseif ((strlen ($_POST["jmeno"])>15) || strlen ($_POST["jmeno"])<3)
echo "Jmeno musi mit 3-15 znaku";
elseif ((strlen ($_POST["prijmeni"])>15) || strlen ($_POST["prijmeni"])<3)
echo "Prijmeni musi mit 3-15 znaku";
elseif ((strlen ($_POST["pojistovna"])>10) || strlen ($_POST["pojistovna"])<3)
echo "Pojistovna musi byt 3-10 znaku";
elseif ((strlen ($_POST["den"])!= 2) or (!is_numeric($_POST["den"])))
echo "Den musi byt 2 mistne cislo<br>";
elseif ((strlen ($_POST["mesic"])!=2) || (!is_numeric($_POST["mesic"])))
echo "Mesic musi byt 2 mistne cislo<br>";
elseif (strlen ($_POST["rok"])!= 4 || (!is_numeric($_POST["rok"])))
echo "Rok musi byt 4 mistne cislo<br>";

```

Obr. 52 Nový klient – kontrola vstupů

Po ověření dat už jen následuje skript pro zapsání do databáze a zapsání do historie. Po zapsání do databáze se stránka automaticky přesměruje na stránku uvitani.php, kde je upozornění o potřebě vyplnit zbývající část klientské dokumentace.

```

$sql = "insert into klienti ( jmeno, prijmeni, patro, pokoj, pojistovna, r_cislo) VALUES
( '".$_POST["jmeno"]."', '".$_POST["prijmeni"]."', '".$_POST["patro"]."',
'".$_POST["pokoj"]."', '".$_POST["pojistovna"]."', '".$_POST["rodne"]."' );
$result = mysql_query($sql);
$sql="SELECT id_kl FROM klienti WHERE r_cislo='".$_POST["rodne"]."' ";
$result = mysql_query($sql);
$nacteni=mysql_fetch_array($result, MYSQL_ASSOC);
$sql = "insert into historiek ( operace, provedl, klient, cas ) VALUES
( '8', '".$_SESSION["id"]."', '$nacteni["id_kl"]."', now() )";
$result = mysql_query($sql);

-EQUIV="Refresh" CONTENT="02; URL=index.php?clanek=uvitani">

```

Obr. 53 Nový klient – zápis do databáze

### 3.11.2 Editace klienta

Editace klienta je dostupná z menu administrátorovi a zdravotnickému personálu pod položkou „editace klienta“ odkazující na soubor editacekl.php. Zde jsou 2 funkce pro

zobrazení klientů. Obě funkce načítají oprávnění uživatele. Po načtení oprávnění se spouští jedna z funkcí. Pokud je přihlášený uživatel sestra, načte se z jejího profilu číslo patra, na kterém působí do proměnné \$stejne. Podle patra se provede výběr klientů s patrem podle proměnné \$stejne. Tato funkce slouží k zabránění editace klienta personálem z jiného patra. Uživatel prostě neuvidí jiné klienty, než na patře které má uvedené na profilu.

Administrátor vidí všechny klienty.

```

if (jesestra ())
{
    $vyber = "
    SELECT uzivatele.patro FROM uzivatele where id='".$_SESSION["id"]."';

    $opakovanivyberu = mysql_query($vyber, $GLOBALS["link"]);
    while($nacteni = mysql_fetch_array($opakovanivyberu))
    {
        $stejne = $nacteni["patro"];
        echo "<br>Odesílatel: ".$stejne;
    }

    $vyber = "SELECT klienti.jmeno, klienti.prijmeni, klienti.patro, klienti.pokoj, klienti.id_kl
    FROM klienti
    where patro='".$_stejne.'"
    ORDER BY `id_kl` desc";

    $opakovanivyberu = mysql_query($vyber, $GLOBALS["link"]);
}

elseif (jeadmin())
{
    $vyber = "
    SELECT klienti.jmeno, klienti.prijmeni, klienti.patro, klienti.pokoj, klienti.id_kl
    FROM klienti
    ORDER BY `id_kl` desc";

    $opakovanivyberu = mysql_query($vyber, $GLOBALS["link"]);
}

```

*Obr. 54 Editace klienta – zobrazení klientů dle oprávnění*

Následuje už jen vypsání klientů dle výběru do tabulky, kde jeden řádek = jeden klient. V každém řádku je i tlačítko „upravit klienta“, které automaticky přesměruje uživatele na stránku zpracovanikl.php a zároveň uloží do proměnné \$odelsat id číslo klienta pro další zpracování právě na stránce zpracovanikl.php. Další tlačítko v každém řádku je tlačítko „smazat klienta“, které odkazuje na stránku smazanikl.php a opět sebou odesílá proměnou \$odelsat s id číslem klienta.

### 3.11.3 Zpracovanikl.php

Zpracovanikl.php je pouze „přestupní stanicí“ pro proces upravení klientské části. Na začátku si načte z předchozí stránky id číslo klienta do proměnné \$\_POST['odeslat'] a ve spodní části stránky zobrazí odkazy na jednotlivé části dokumentace, které jsou skryty v souborech podle názvu části dokumentace.

Jsou to:

```
?>
<form method="post" action="index.php?clanek=fyziologie" >
  <input type="hidden" name="odeslat" value="<?=$_POST['odeslat'];?>" />
  <input type="submit" name="odeslano" value="Fyziologie" />
</form>

<form method="post" action="index.php?clanek=vnimani" >
  <input type="hidden" name="odeslat" value="<?=$_POST['odeslat'];?>" />
  <input type="submit" name="odeslano" value="Vnimani a poznávání" />
</form>

<form method="post" action="index.php?clanek=nemoci" >
  <input type="hidden" name="odeslat" value="<?=$_POST['odeslat'];?>" />
  <input type="submit" name="odeslano" value="Nemoci" />
</form>

<form method="post" action="index.php?clanek=alergie" >
  <input type="hidden" name="odeslat" value="<?=$_POST['odeslat'];?>" />
  <input type="submit" name="odeslano" value="Alergie a diety" />
</form>

<form method="post" action="index.php?clanek=leky" >
  <input type="hidden" name="odeslat" value="<?=$_POST['odeslat'];?>" />
  <input type="submit" name="odeslano" value="Leky" />
</form>

<form method="post" action="index.php?clanek=kontakty" >
  <input type="hidden" name="odeslat" value="<?=$_POST['odeslat'];?>" />
  <input type="submit" name="odeslano" value="Lekari a opatrovnici" />
</form>
```

Obr. 55 Zpracovanikl.php - jednotlivé odkazy

Spolu s přesměrováním na stránku se předá proměnná `$_POST['odeslat']` s id číslem klienta pro další zpracování.

### 3.11.4 Podpůrné soubory pro editaci jednotlivých částí klientské dokumentace

Jedná se o sadu souborů, na které se odkazuje ze stránky `zpracovanikl.php`, slouží pro přístup do jednotlivých částí klientské dokumentace. Jejich konstrukce je velmi podobná jako úprava uživatelského profilu. Jsou to sady formulářů, kontroly zadaných dat a skripty pro zápis do databáze. Jedná se o stále opakující se funkce, z toho důvodu zde uvedu pouze 2 soubory, kde ukážu, jakým způsobem jsou data v aplikaci sbírána a upravována.

#### 3.11.4.1.1 Fyziologie.php

Sada skriptů starající se o chod části aplikace, kde se klientovi upravuje jeho karta se zdravotnickými informacemi. V první řadě se načítá z proměnné `$_POST['odeslano']` id číslo klienta pro evidenci „s jakým klientem se pracuje“. Ve spodní části jsou různé typy formulářů, v tomto případě klasické textové a zaškrťovací. Kontrola formulářů je opět

stejná jako u předchozích funkcí a systém zobrazení je také na stejné úrovni. Jediná změna je při zobrazování dat, kdy je v databázi uloženo 1 nebo 0. V aplikaci je to řešené pomocí přepínače, aby uživatel nedostával informace typu 1 0.

```
<form method="post" action="index.php?clanek=fyzio">
  <div align="left">Pohyb:
  <?
  switch ($nacteni["mobilni"]):
    case "1":
      echo "Mobilni";
    endswitch;
  ?>
  <?
  switch ($nacteni["imobilni"]):
    case "1":
      echo "Imobilni";
    endswitch;
  ?>
</div>
```

*Obr. 56 Editace klienta – zobrazení při editaci karty*

### 3.11.5 nemoci.php

Pro příklad malé změny oproti ostatním souborům jsou funkce v souborech nemoci.php a alergie.php. Jedná se o možnost zvolení více řádků pro přidání určitého léku nebo nemoci. Pokud by se zadávalo po jedné a odesílalo, je to jisté zdržení, personál tak může zvolit více řádků pro rychlejší vyplnění dokumentace. Pro tuto funkci jsem použil jediný javaskript v celé aplikaci. Jedná se o funkci přidání vložené funkce, kde jsou vkládány 2 formuláře textové pro proměnou nemoc[] a stav[].



```

<script>
function AddInput()
{
    var div=document.createElement('div');
    div.innerHTML=' Nemoc: <input type="text" name="nemoc[]"><br>Poznamka: <input type="text"
    document.getElementById('inputy').appendChild(div);
}
</script>

<form method="post" action="index.php?clanek=nemoci">
<div id="inputy">

Nemoc: <input type="text" name="nemoc[]"><br>Poznamka: <input type="text" name="stav[]"><hr>
</div>
<input type="button" onclick="AddInput();return false;" value="Pridat radek">
<input type="hidden" name="odeslat" value="<?=$_POST['odeslat'];?>" />
<input type="submit" name="odeslano" value="Vlozit nemoc">
</form>

```

Obr. 57 Nemoci.php – více řádků

Proměnné `$_POST['nemoc']` a `$_POST['stav']` jsou zpracovány pomocí skriptu, jenž se spouští po odeslání tlačítka „vložit nemoc“. Skript tyto proměnné uloží nejprve do proměnné `$stav` a `$nemoc` a poté pomocí cyklu `for($i=0;$i<count($nemoc);$i++)` ukládá proměnné `$_POST['nemoc']` a `$_POST['stav']` do `$Nemoc = $nemoc[$i]` `$Stav = $stav[$i]`; proměnné `$_POST['nemoc']` a `$_POST['stav']` se uloží do proměnných popsanych výše, kde `i` = číslo řádku. Spolu se zpracováním každého řádku se provádí zápis do databáze.

```

if (!empty($_POST["nemoc"]))
{

if ($_POST["odeslano"]=="Vlozit nemoc")
    $nemoc = $_POST['nemoc'];
$stav = $_POST['stav'];
for ($i=0;$i<count($nemoc);$i++)
{
    $Nemoc = $nemoc[$i];
    $Stav = $stav[$i];
    $Poradi = $i+1;
    if($Nemoc)
    {
        if (nemocvdb($nemoc[$i], $odesilatel))
        {

```

Obr. 58 Nemoci.php – cyklus zpracování

### 3.11.6 Zobrazení klientské karty

K zobrazení klientské dokumentace je potřeba nejprve specifikovat klienta kterého chceme zobrazit. K tomu slouží odkaz na stránku `zobrazeni.php`, který je v menu dostupný administrátorovi a zdravotnickému personálu pod položkou „zobrazení klienta“. Jelikož jde vyhledávat podle 2 způsobů, je tady sada formulářů starající se o běh aplikace. První variantou je soubor formulářů, rozdělující písmenka abecedy a odesílající vždy jedno písmeno. Zpracování v souboru `zpracovanizob.php`, na který tyto formuláře odkazují je popsáno níže.

```
<table>
<tr>
<form method="post" action="index.php?clanek=zpracovanizob" >
<input type="hidden" name="prvni" value="a" />
<input type="submit" name="odeslano" value="A - E" />
</form>

<form method="post" action="index.php?clanek=zpracovanizob" >
<input type="hidden" name="prvni" value="f" />
<input type="submit" name="odeslano" value="F - I" />
</form>

<form method="post" action="index.php?clanek=zpracovanizob" >
<input type="hidden" name="prvni" value="j" />
<input type="submit" name="odeslano" value="J - N" />
</form>

<form method="post" action="index.php?clanek=zpracovanizob" >
<input type="hidden" name="prvni" value="o" />
<input type="submit" name="odeslano" value="O - S" />
</form>

<form method="post" action="index.php?clanek=zpracovanizob" >
<input type="hidden" name="prvni" value="t" />
<input type="submit" name="odeslano" value="T - W" />
</form>

<form method="post" action="index.php?clanek=zpracovanizob" >
<input type="hidden" name="prvni" value="x" />
<input type="submit" name="odeslano" value="X - Z" />
</form>
</tr>
```

Obr. 59 Zobrazení klienta – abecední vyhledávání

Druhou variantou je vybrat klienta podle patra a pokoje. Realizováno je to pomocí rozevíracích formulářů, jsou plněny na základě funkcí podobných při výběru patra v jiných funkcích. Výběr pokoje je plněn funkcí obdobnou jako u výběru patra, s tím že se předává proměnná `$_POST["patro"]` a výběr pokojů se provádí s podmínkou `where patro = $_POST["patro"]`, pokoj se opět ukládá do proměnné `$_POST["pokoj"]`, a spouští se poslední výběrové okno, které se naplní z databáze se jmény klientů podle zvoleného

pokoje. Skripty jsou psány tak, aby se jednotlivé formuláře s nabídkou vypsalí až po odeslání toho předchozího, uživatel tak zadává údaje a stránka se mu dynamicky mění.

Skript je obsáhlý, pro vložení obrázku, je proto nutné se na něj podívat v příloze souboru zobrazeni.php

Obě varianty vedou k souboru zpracovanizob.php

### 3.11.7 Zpracovanizob.php

Tento soubor obsahuje skripty pro zpracování \$\_POST['odeslano'], ve kterém se nachází začáteční písmeno vybrané dle tlačítka z předchozí stránky nebo se předá proměnná \$\_POST['klient'], kde je uloženo id číslo klienta a to v případě, že byl klient vybrán pomocí rozevíracích formulářů.

V prvním případě se načte první písmeno z proměnné \$\_POST['odeslano'] a pomocí funkcí if a elseif se zjišťuje, jaké písmenko bylo odesláno. Na základě odeslaného písmenka se spustí jedna z podmínek, v jejímž těle se nachází pole proměnných, do kterých se ukládají jednotlivá písmena abecedy dle intervalu, který byl označen na tlačítku na předchozí stránce. Na obrázku je příklad uložení písmen pro kliknutí na tlačítko A – E.

```
if (!jeobji ()) return;
if (isset($_POST['odeslano']))
{
    $prvni = $_POST['prvni'];
    echo "<br>Prvni: ".$prvni;

    echo "<br><br> ";
    if ($prvni==a)
    {
        $a='A';
        $b='B';
        $c='C';
        $d='&#268;'; //č
        $e='&#269;'; //č
        $f='D';
        $g='&#270;'; //ď
        $h='&#271;'; //ď
        $i='&#282;'; //ě
        $j='&#283;'; //ě
        $k='E';
    }
}
```

Obr. 60 Zobrazení klienta – vybrání podle abecedy

K této funkci jsem přistoupil, ačkoliv není úplně nejvhodnějším řešením, nicméně je jediná, která dokázala zobrazit písmenka podle abecedy dle vybraného kritéria. Problém

nastal v případě, kdy některý z uživatelů zaeviduje nového klienta s příjmením čermák a zaeviduje ho s malým „č“ na začátku. Pak by funkce s vyhledáváním where between and nefungovala správně a takového klienta by nezobrazila. Znaková sada databáze byla nastavena na UTF8-czech\_ci, což by měly být české znaky a při vyhledávání nerozlišovat malé a velké znaky. Nicméně při uložení například již zmíněného písmenka „č“ byl tento znak v databázi uložen jako „&#269;“. Při zpětném přečtení je tento znak normálně zobrazen jako malé „č“. Proto jsem otestoval všechny znaky české abecedy, abych zjistil jejich znakovou sadu v databázi a uložil jak malé, tak velké české písmena do pole, jak je vidět na obrázku výše. Takto uložené pole jsem využil při prohledávání databáze a to tak, že jsem použil podmínku where příjmení začíná na písmeno uložené v proměnné a, nebo v proměnné b atd....

```
$vyber = "
SELECT klienti.jmeno, klienti.prijmeni, klienti.patro, klienti.pokoj, klienti.id_kl
FROM klienti
WHERE prijmeni LIKE '$a.%' or
prijmeni LIKE '$b.%' or
prijmeni LIKE '$c.%' or
prijmeni LIKE '$d.%' or
prijmeni LIKE '$e.%' or
prijmeni LIKE '$f.%' or
prijmeni LIKE '$g.%' or
prijmeni LIKE '$h.%' or
prijmeni LIKE '$i.%' or
prijmeni LIKE '$j.%' or
prijmeni LIKE '$k.%'

ORDER BY `prijmeni` desc";
```

Obr. 61 Zobrazení klienta – výpis z databáze

Při variantě s rozevíracími formuláři je situace poněkud jednodušší a to v tom, že do proměnné \$\_POST['klient'] je uložen id číslo konkrétního klienta a výběr z databáze podle tohoto čísla už není žádným problémem.

```
$prvni = $_POST['klient'];
echo "<br>Prvni: ".$prvni;

echo "<br><br> ";

$vyber = "
SELECT klienti.jmeno, klienti.prijmeni, klienti.patro, klienti.pokoj, klienti.id_kl
FROM klienti
where id_kl='".$_POST["klient"]."'

ORDER BY `id_kl` desc";
```

Obr. 62 Zobrazení klienta – výběr z databáze pomocí formulářů

Po vypsání klienta, v případě abecedního seznamu klientů, do tabulky, je v každé řádce jeden klient a jedno tlačítko „zobrazit dokumentaci“, které odkazuje na stránku karta.php, kde se zobrazí kompletní dokumentace klienta. Odeslána je opět proměnná \$\_POST['odeslano'], kde je uloženo id číslo klienta. Na základě id čísla je už potřeba jen pospojovat jednotlivé tabulky a vypsát všechny informace týkající se zdravotní karty klienta. Skript pro zobrazení je obsáhlý, proto je uveden pouze v příloze. Jednotlivé funkce a s pojení tabulek je popsáno v kapitole „relace“.

### 3.11.8 Karta.php

Soubor sloužící k zobrazení kompletních informací o klientovi na jedné stránce. Vše se zobrazuje do tabulky do řádků. Spojení tabulek pro výpis informací je vidět na obrázku níže.

```
$vyber = "SELECT klienti.jmeno, klienti.prijmeni, klienti.narozeni, klienti.pojistovna,
klienti.patro, klienti.pokoj,
fyzio.vaha as vaha, fyzio.vyska as vyska,
pohyb.mobilni, pohyb.imobilni, pohyb.poznamka,
poznvani.sluch as sluch, poznvani.pomucka as pomucka, poznvani.poznamka as poznamkasluch,
poznvani.zrak as zrak, poznvani.bryle as bryle, poznvani.rec as rec, poznvani.orientace_os as or_os,
poznvani.orientace_cas as or_cas, poznvani.orientace_ms as or_ms, poznvani.orientace_sit as or_sit
FROM klienti
LEFT JOIN fyzio ON klienti.id_kl = fyzio.klient
LEFT JOIN pohyb ON klienti.id_kl = pohyb.klient
LEFT JOIN poznvani ON klienti.id_kl = poznvani.klient

where id_kl='".$.$klient.'"";
$zobrazeni = mysql_query($vyber, $GLOBALS["link"]);
while($nacteni = mysql_fetch_array($zobrazeni))
{
```

*Obr. 63 Zobrazení klienta – celkový pohled*

Zobrazení opakujících se věcí, jako mohou být například nemoci, je řešen samostatným cyklem while. Výsledkem je zobrazení opakujících se informací do řádku v návaznosti na ostatní informace.

```

$vyber = "SELECT nemoci.nemoc, nemoci.poznamka
          FROM nemoci
          where klient='".$klient."'";
$zobrazeni = mysql_query($vyber, $GLOBALS["link"]);
if (mysql_num_rows($zobrazeni)==0)
echo "Neni zadna nemoc v databazi";
else
{
  while($nacteni = mysql_fetch_array($zobrazeni))
  {
    <div align="left">Nemoc: <?=$nacteni["nemoc"];?><br>
    Poznamka: <?=$nacteni["poznamka"];?> <br><br>
  }
}

```

Obr. 64 Zobrazení klienta – opakující se informace

Informace uložené v databázi formou binárního data – 1 nebo 0, jsou zobrazovány pomocí přepínačů.

```

<div align="left">Poruchy reci: <?
switch ($nacteni["rec"]):
  case "0":
    echo "Rec je srozumitelna";
    break;
  case "1":
    echo "Rec neni srozumitelna";
    break;
  default:
    echo "-";
endswitch;
</div>

```

Obr. 65 Zobrazení klienta – přepínače

## 4 UŽIVATELSKÝ MANUÁL K APLIKACI

### 4.1 Přístup k aplikaci

Přístup k aplikaci záleží na nastavení serveru, u cílového klienta. Při vývoji byl použit již zmíněný uniform server, který má základní přístup z adresy: `http://localhost:8000/`

Pro mé použití je adresa `http://localhost:8000/diplomka/`. Název „diplomka“ značí kořenovou složku souborů celé aplikace.

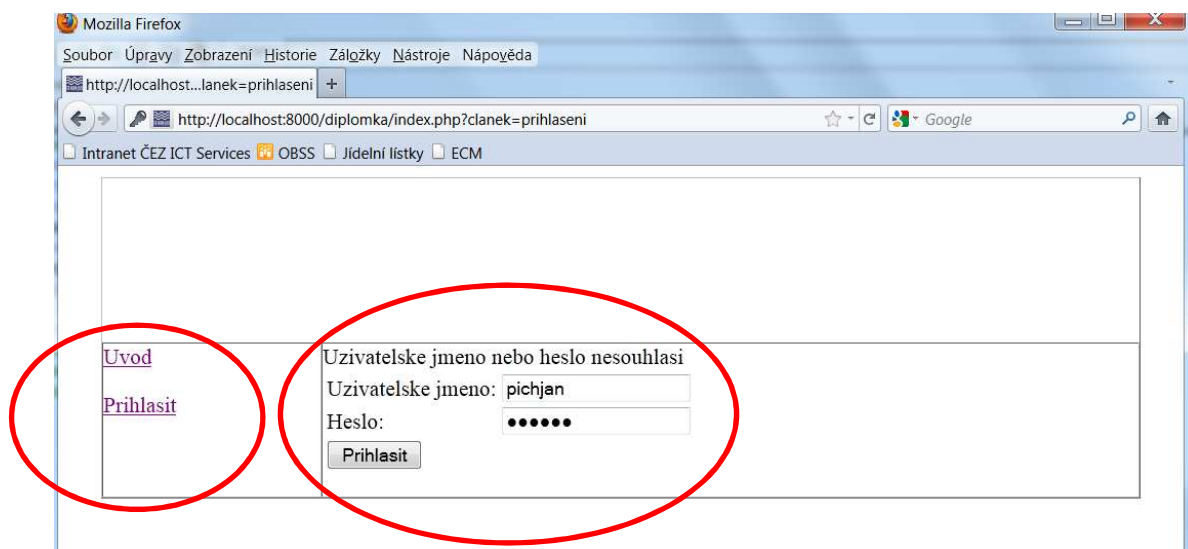
### 4.2 Přihlášení

Uživatel se po spuštění dostane buď přímo na stránku přihlášení, nebo se na ní přihlásí kliknutím na „přihlásit“.

Do pole „uživatelské jméno“ zadá uživatel svou přezdívku, která mu byla nastavena při jeho registraci nebo podle nastavení podle politiky firmy.

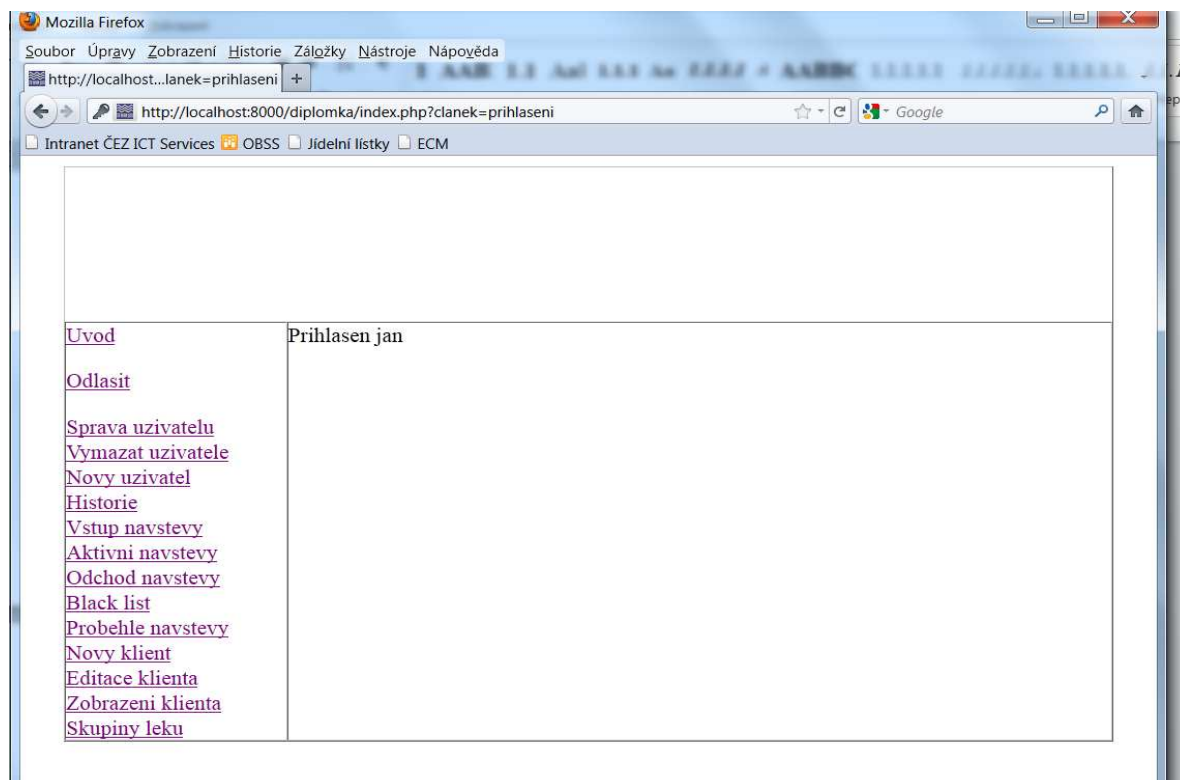
Do pole „heslo“ zadá uživatel své nastavené heslo. Poté kliknutím na tlačítko „přihlásit“ se uživatel přihlásí do aplikace.

Pokud uživatel zadá nesprávné uživatelské jméno nebo heslo, aplikace ho automaticky upozorní.



Obr. 66 Manuál - přihlášení

Po přihlášení uživatel dostane zprávu, že přihlášení proběhlo v pořádku. V levém dolním okně rozbalí nabídka „menu“, podle uživatelských pravomocí. Na obrázku je vidět nabídka uživatele administrátor.



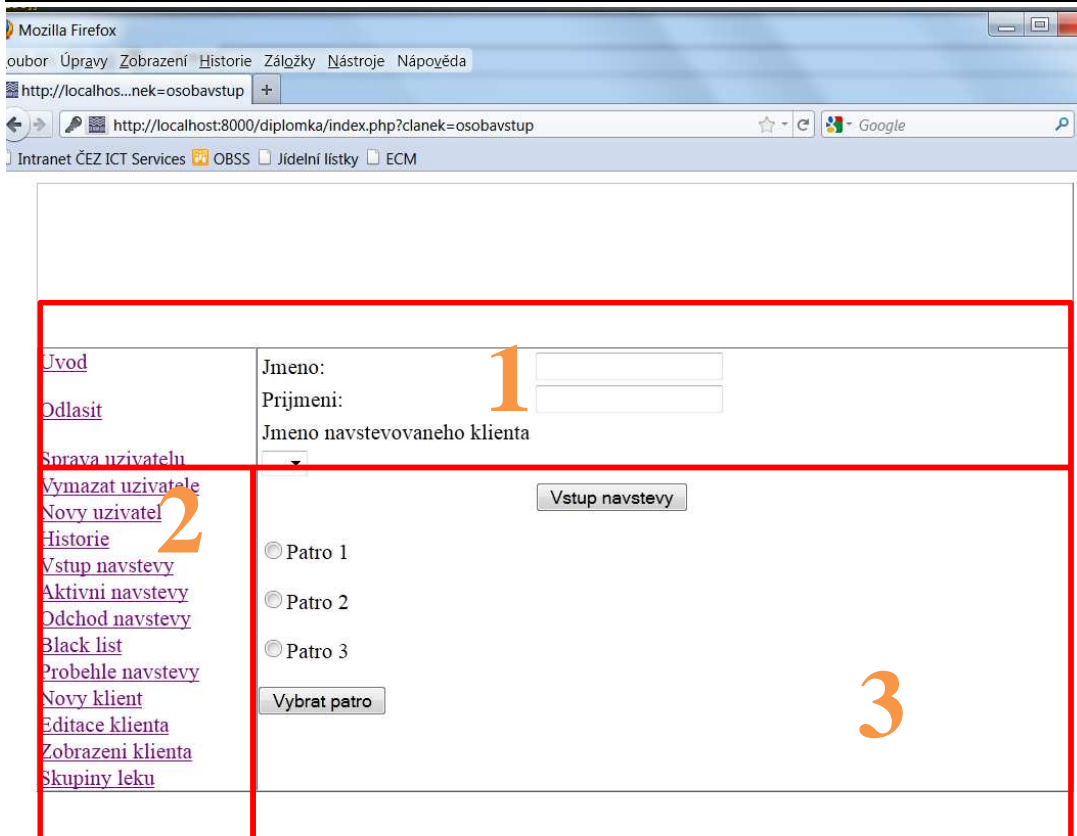
Obr. 67 Manuál - menu

### 4.3 Základní vzhled

Základní vzhled aplikace sestává z 3 navázaných oken, kdy vrchní okno (1) slouží k napsu aplikace, logu společnosti nebo jinému grafickému útvaru. Levá dolní část (2) zastává funkci hlavního menu a pravá dolní část (3) slouží jako aktivní zobrazovací okno. Pro uživatelské použití jsou důležitá 2 spodní okna, kdy v levém okně vybíráme požadované funkce, a v pravém okně se zobrazují výsledky funkcí.

1. Okno pro grafiku
2. Menu
3. Zobrazovací okno





Obr. 68 Manuál – základní vzhled

#### 4.4 Menu

Menu uživatel najde v levém dolním okně (2). Každý uživatel má přidělenou jednu ze tří rolí a k ní vztahující se oprávnění. Podle role uživatele se mění i obsah menu. Administrátor má k dispozici všechna oprávnění, včetně správy uživatelů aplikace. Uživatel z řad zdravotnického personálu má k dispozici funkce vztahující se k zdravotní části a funkce návštěvní knihy. Uživatelé s nejnižším oprávněním mají k dispozici pouze omezené množství funkcí návštěvní knihy a to vstup návštěvy a odchod návštěvy, popřípadě zaevidování návštěvníka do blacklistu návštěvní knihy.

<a href="#">Uvod</a>	<a href="#">Uvod</a>	<a href="#">Uvod</a>
<a href="#">Odlasit</a>	<a href="#">Odlasit</a>	<a href="#">Odlasit</a>
<a href="#">Sprava uzivatelu</a>	<a href="#">Vstup navstevy</a>	<a href="#">Vstup navstevy</a>
<a href="#">Vymazat uzivatele</a>	<a href="#">Aktivni navstevy</a>	<a href="#">Aktivni navstevy</a>
<a href="#">Novy uzivatel</a>	<a href="#">Odchod navstevy</a>	<a href="#">Odchod navstevy</a>
<a href="#">Historie</a>	<a href="#">Black list</a>	
<a href="#">Vstup navstevy</a>	<a href="#">Probehle navstevy</a>	
<a href="#">Aktivni navstevy</a>	<a href="#">Novy klient</a>	
<a href="#">Odchod navstevy</a>	<a href="#">Editace klienta</a>	
<a href="#">Black list</a>	<a href="#">Zobrazeni klienta</a>	
<a href="#">Probehle navstevy</a>	<a href="#">Skupiny leku</a>	
<a href="#">Novy klient</a>		
<a href="#">Editace klienta</a>		
<a href="#">Zobrazeni klienta</a>		
<a href="#">Skupiny leku</a>		

Obr. 69 Manuál – jednotlivé menu

## 4.5 Aktivní okno

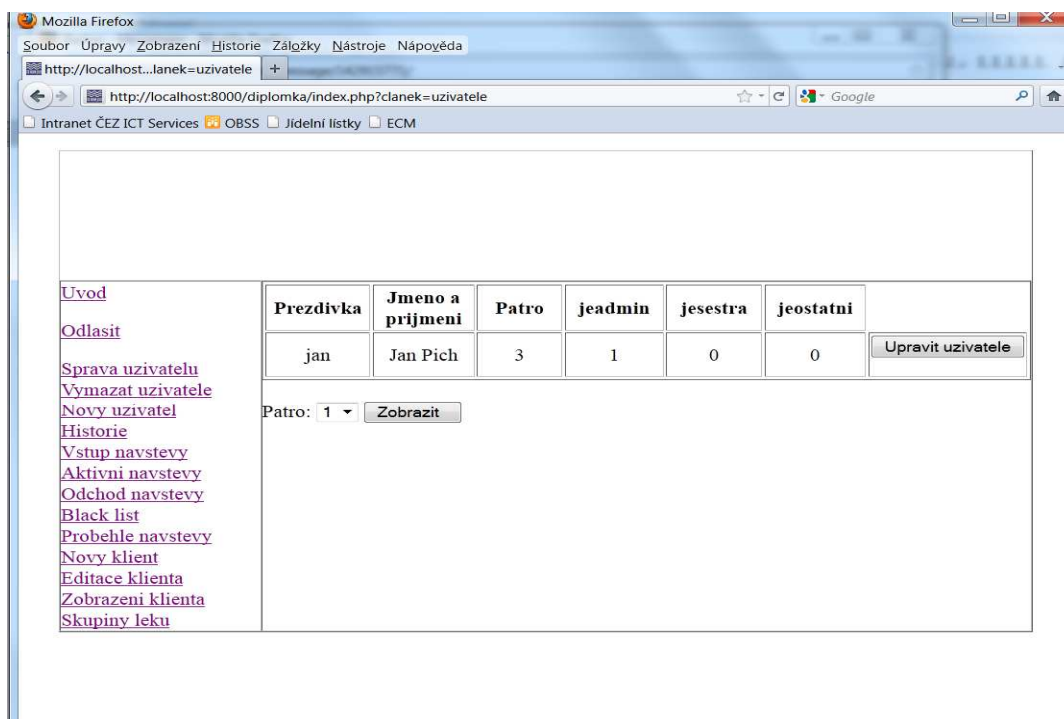
Toto okno slouží pro zobrazení výstupu vybrané funkce z menu, nebo i funkce, kterou jsme použili přímo v aktivním okně. V tomto okně se bude zobrazovat veškerý obsah. Slouží jako výstupní zobrazení.

## 4.6 Jednotlivé funkce

V této kapitole budou popsány všechny funkce pro jednotlivé typy účtů.

### 4.6.1 Správa uživatelů

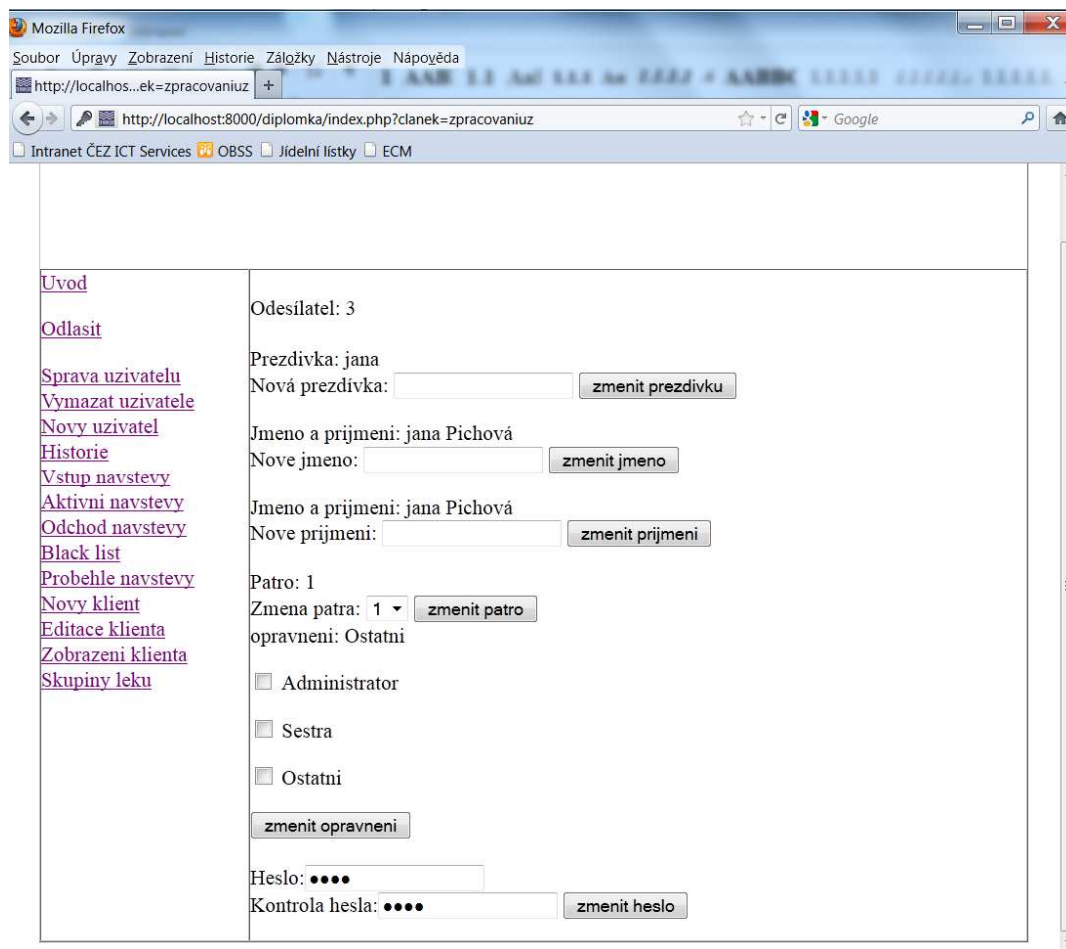
V této funkci administrátor spravuje uživatele aplikace. Na začátku si administrátor vybere patro, aby se eliminoval problém s větším počtem záznamů. Po vybrání patra z rozevřacího seznamu se dotaz odešle tlačítkem „zobrazit“. Pokud není pro dané patro žádný záznam, vypíše aplikace hlášku „není žádný záznam v databázi“. Pokud jsou k dispozici nějaké záznamy, aplikace po té vypíše v tabulce uživatele z vybraného patra s informacemi o přezdívkě, jménu, příjmení a oprávnění. V každém řádku je jeden klient a jedno tlačítko „upravit uživatele“.



Obr. 70 Manuál – správa uživatelů

#### 4.6.2 Upravit uživatele

Tato funkce spravuje informace o uživateli a umožňuje je dynamicky měnit. Měnit lze: jméno, příjmení, přezdívka, patro, oprávnění a heslo. Každá z informací se mění samostatně, kdy je v prvním řádku vypsána aktuální hodnota a v druhém řádku je formulář a tlačítko pro změnu hodnoty. Při odeslání změny uvidí administrátor okamžitou změnu hodnoty.

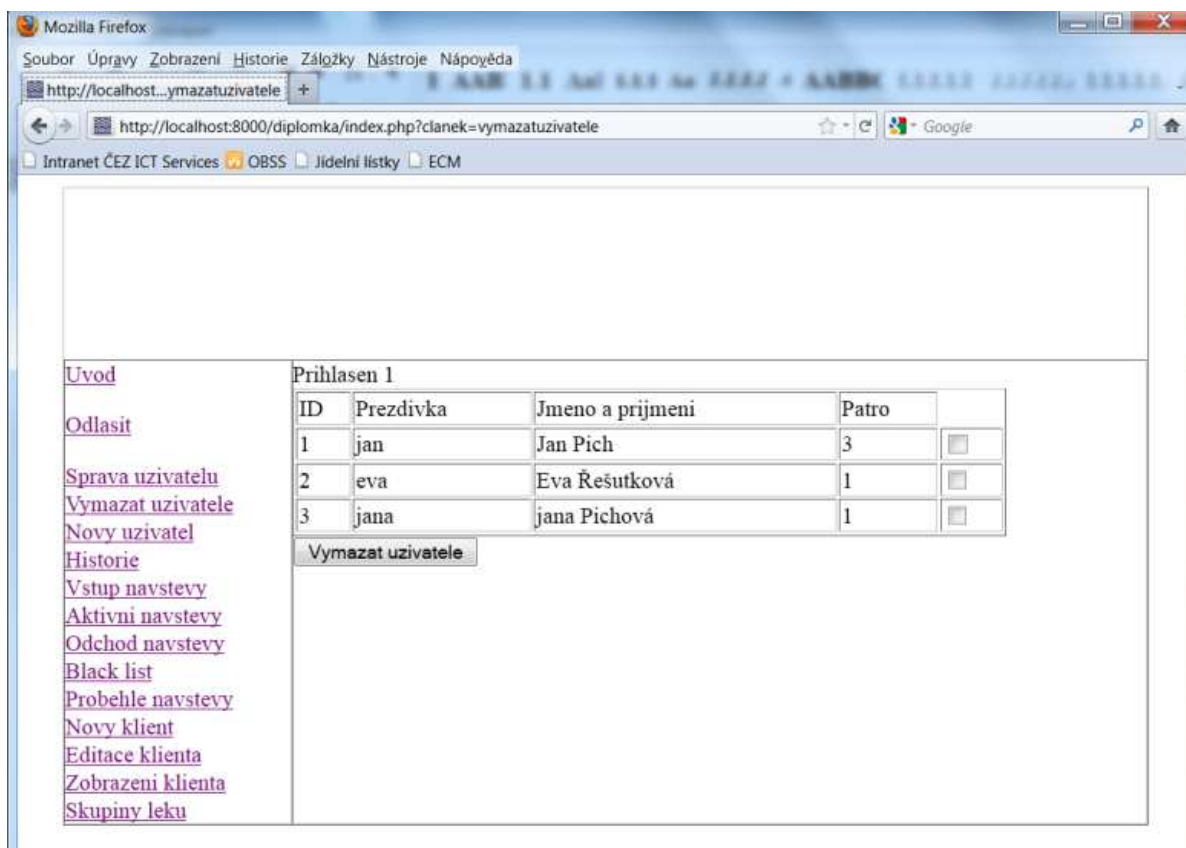


Obr. 71 Manuál – editace uživatele

Formuláře jsou ošetřené na počet znaků. Přesný počet se mění dle potřeby, na příslušnou délku aplikace vždy upozorní. Dále jsou ošetřené zaškrtačkové formuláře, proti zaškrtnutí více než jedné hodnoty. Formulář na změnu přezdívky dále kontroluje existenci duplicitní přezdívky v databázi. Formulář na změnu hesla je kontrolován na správnost zadaného hesla v ověřovacím formuláři.

### 4.6.3 Vymazat uživatele

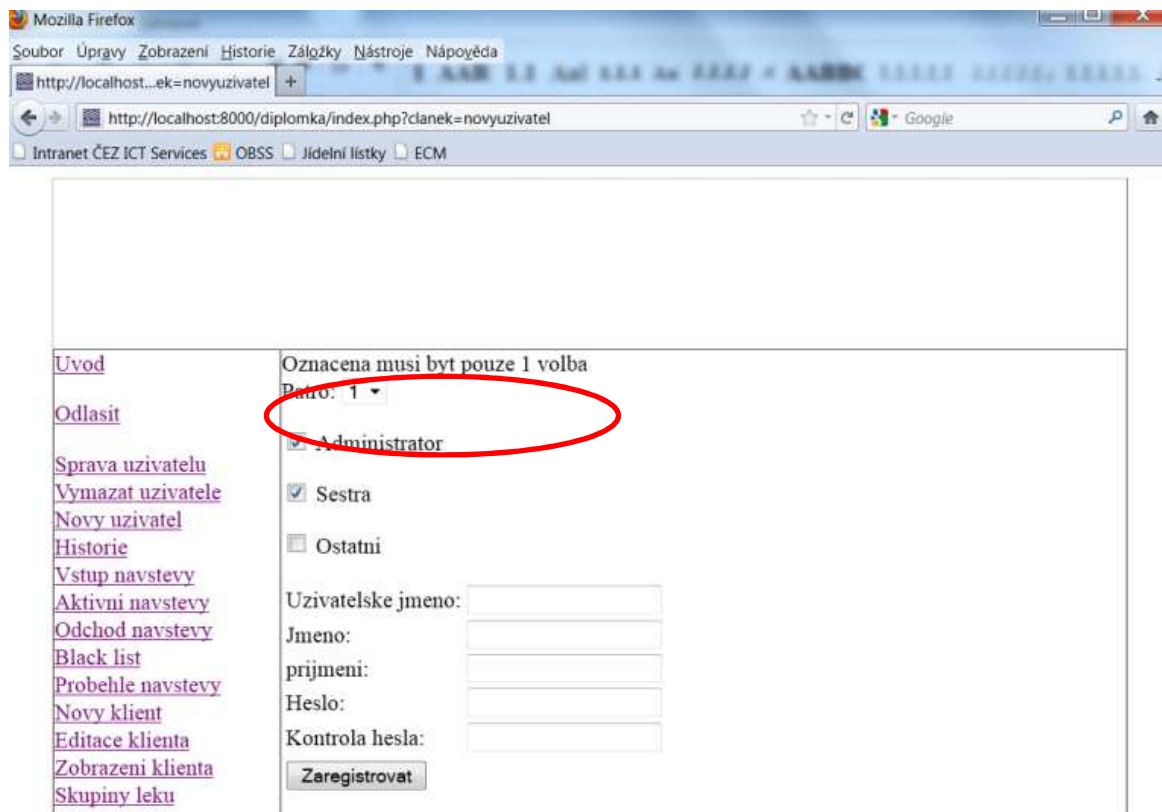
Tato funkce, jak je již dle názvu vypovídající, slouží pro smazání uživatelů. Je tvořena pomocí tabulky, kde je vypsáno ID číslo klienta, jméno a příjmení, přezdívkou a patro. V každém řádku je jeden klient a jedno zaškrtačací políčko. Administrátor zaškrtně políčka u vybraných uživatelů a tlačítkem „vymazat uživatele“ je odstraní z databáze.



Obr. 72 Menu – mazání uživatele

### 4.6.4 Nový uživatel

Přes tuto stránku může uživatel s rolí administrátora přidávat uživatele aplikace. Zadává se: jméno a příjmení, přezdívkou, patro, oprávnění a samozřejmě heslo. Formuláře jsou opět ošetřeny na počet znaků, duplicitu přezdívkou, počet odškrtnutých políček s oprávněním nebo kontrolu zadaného hesla. Pokud bude jakákoliv hodnota v rozporu, objeví se hláška o kolizi nad zadávaným formulářem.



Obr. 73 Manuál – nový uživatel

#### 4.6.5 Historie

Historie je nedílnou součástí této aplikace. Někdy je potřeba dohledat kdo vytvářel nového uživatele, popřípadě klienta, kdo ho upravoval, kdo ho smazal, atd. Jakákoliv operace by měla být monitorována. V rámci aplikace jsou monitorovány 3 oblasti:

Klienti - smazání, přidání a editace.

Uživatelé – přidání smazání a editace.

Léky – přidání léku, smazání léku, přidání skupiny a smazání skupiny.



Obr. 74 Manuál - historie

## 4.6.5.1.1 Historie uživatelé

Uživatel se pod touto nabídkou dostane na úvodní okno historie uživatelů. V nabídce jsou přepínačové formuláře, kterými lze zvolit operaci, jejíž historii chceme vidět. V nabídce jsou výše zmíněné možnosti. Další možností je zobrazení celkového seznamu, kde se zobrazují všechny operace v závislosti na vybraném měsíci. Zvolíme měsíc, odešleme a aplikace zobrazí všechny operace, které byly provedeny od 1 do posledního dne měsíce.

Operace ID	Uzivatel	Provedl	Cas
------------	----------	---------	-----

[Kompletní seznam](#)

Smazani uzivatele  Pridany uzivatele  Editace

Obr. 75 Manuál – Historie uživatelé

je rok 2012  
Zadny zaznam

Operace	Uzivatel	Provedl	Cas
---------	----------	---------	-----

Mesic: 01

- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10
- 11
- 12

Obr. 76 Manuál – Historie uživatelé seznam

#### 4.6.6 Historie klienti

Historie klientů je řešená stejným způsobem jako historie uživatelů a to vstupní obrazovkou s výběrem operace (mazání, přidání, editace) a odkazem na kompletní seznam operací s výběrem měsíce.

Operace	Klient	Provedl	Cas
editace klienta	jan Č Čermák 4	jan	28/4 2012 v 13:15:50
editace klienta	Jana hhhh 13	jan	28/4 2012 v 12:50:06
smazani klienta	šanÁ		28/4 2012 v 11:26:05

Obr. 77 Manuál – historie klienti zobrazení

Pokud je políčko provedl prázdné, uživatel, který tuto operaci provedl už je smazaný.

#### 4.6.7 Historie léky

Historie léků má 2 volitelné položky operací a to smazání a přidání léku, a smazání a přidání skupiny léků. Seznam celkových operací podle měsíců je v této části také implementován.

Operace ID	Lek nebo skupina	Provedl	Cas
1	test lék 3	jan	05/5 2012 v 12:39:15

#### Kompletní seznam

Pridani leku
  Pridani skupiny

Obr. 78 Manuál - historie léky

#### 4.6.8 Nový klient

V této části zdravotnický personál přidává nové klienty, avšak pouze základní identifikační údaje jako jsou jméno, příjmení, rodné číslo, pojišťovna, patro a pokoj. Opět jsou zde ošetřené formuláře na počet znaků a formuláře, data narození na počet znaků + znaky musí být čísla. Datum narození je kontrolováno speciální funkcí, proto je nutné zadat datum narození správně. Aplikace opět sama upozorní uživatele na chybné zadání. Duplicita je zde kontrolována pomocí jména a příjmení klienta. Dále je kontrolována správnost zadání patra a pokoje. Musí souhlasit první číslice pokoje a patra. Po odeslání aplikace uživatele automaticky nasměruje na stránku s upozorněním a instrukcemi, aby byla klientská karta doplněna o zdravotnické informace. Zde uživatel najde informace jak a kde vyplnit zbylé informace, aby mohla aplikace plnit svůj účel.

Datum není v pořádku

Jmeno:	<input type="text" value="Petr"/>
Prijmeni:	<input type="text" value="Pich"/>
Den narození (01-31):	<input type="text" value="30"/>
Mesic narození (01-12):	<input type="text" value="13"/>
Rok narození (yyyy):	<input type="text" value="1980"/>
Pojistovna:	<input type="text" value="vzp"/>
Patro: 1 ▾	
Pokoj:	<input type="text" value="125"/>

Obr. 79 Manuál - nový klient



## Dokonceni editace klienta.

1. K dokončení je potřeba vyplnit zdravotní část karty klienta.
2. Pokračujte prosím na stránku "[editace klienta](#)", kterou najdete v menu a doplňte prosím zdravotní část dokumentace.

*Obr. 80 Manuál – dokončení registrace*

### 4.6.9 Editace klienta

Zdravotnický personál zde upravuje informace o klientovi. Administrátor zde má možnost vidět všechny klienty ze všech pater. Zdravotnický personál zde uvidí pouze klienty z patra, které má uveden v profilu. Tato funkce je zde proto, aby nemohl zdravotnický personál upravovat klienty z jiného patra, než on sám působí. V každém řádku je jeden klient, a 2 tlačítka – smazat a editovat.

Pokud klient odešle tlačítko editovat, aplikace ho přesměruje na stránku, kde je na výběr z 6 podkategorií, které může uživatel editovat.

Jedná se o:

- Fyziologie
- Vnímání a poznávání
- Nemoci
- Léky
- Alergie a diety
- Kontakty

Jednotlivé možnosti nastavení v podkategoriích jsou popsány v popisu aplikace pro zdravotnický personál.

klient	Patro	Pokoj		
Petr Čunek	2	236	Upravit klineta	Smazat klineta!!!
Jana hhhhh	2	225	Upravit klineta	Smazat klineta!!!
žžž Žžžž	1	111	Upravit klineta	Smazat klineta!!!
JANA ý Ýcxycy	1	123	Upravit klineta	Smazat klineta!!!
jan Ě ě	1	112	Upravit klineta	Smazat klineta!!!

Obr. 81 Manuál – editace klienta

Odesílatel: 14

#### 4.6.10 Smazání klienta

Na odkaz pro smazání klienta se dostaneme přes menu – editace klienta. Možnosti výběru jsou popsány ve výše uvedené kapitole. Po kliknutí na „smazat klienta“, aplikace přesměruje uživatele na stránku, kde se zobrazí upozornění o možných následcích, které nastanou po smazání. Jedná se zejména o kompletní smazání klientské karty a zdravotní části uložené v databázi. Pro smazání je nutné odškrtnout tlačítko „souhlasím“, aby nedošlo k mylnému smazání.

#### **Upozornění!!!!**

**Smazáním klienta vymazete všechny údaje o něm obsazené v databázi.  
Vše bude nenávratně smazáno**

**Klient: Petr Čunek**

Rozumím s informacemi o smazání

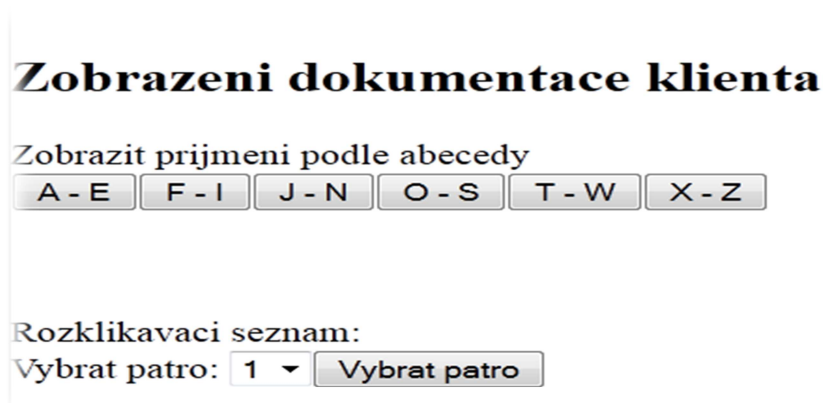
Obr. 82 Manuál – smazání klienta

#### 4.6.11 Zobrazení klienta

Toto je nejdůležitější funkce aplikace, respektive tato funkce je klíčová pro obecné naplnění předpokladu aplikace. Je vytvořena tak, aby v případě krizové situace bylo možné co nejrychleji vyhledat požadovaného klienta a zobrazit jeho dokumentaci. V této aplikaci lze vyhledat požadovaného klienta dvěma způsoby:

1. Podle abecedy – tlačítka A – E, F – I, J – N, O – S, T – W, X – Z
2. Výběrem patra, následuje výběr dostupných pokojů, a následuje výběr klienta ze zvoleného pokoje.

Dvě funkce jsou zvoleny z důvodu zvýšení možnosti nalezení klienta, kdy by například bylo špatně zadané příjmení, aby bylo možné najít klienta i podle pokoje.



**Zobrazení dokumentace klienta**

Zobrazit příjmení podle abecedy

A - E   F - I   J - N   O - S   T - W   X - Z

Rozklikavací seznam:

Vybrat patro: 1 ▾   Vybrat patro

*Obr. 83 Manuál – zobrazení klientské dokumentace*

Dle zvolené varianty vybrání klienta aplikace zobrazí novou stránku s tabulkou, kde budou vypsaní klienti, řazení podle abecedy příjmení, nebo konkrétně vybraný klient.

Budou zde informace jméno, příjmení, patro, pokoj a tlačítko „zobraz dokumentaci“. Tímto tlačítkem se zvolí zobrazení kompletního souboru všech informací, které má klient zadané v databázi. Dle tohoto seznamu se poté může zdravotnický personál řídit.

klieut	Patro	Pokoj	
jan eivh	1	125	Zobrazit dokumentaci
jan Áhoj	1	123	Zobrazit dokumentaci
jan Ď ě	1	125	Zobrazit dokumentaci
Petr Čunek	2	236	Zobrazit dokumentaci
jan Č Čermák	1	125	Zobrazit dokumentaci

Obr. 84 Manuál – zobrazení dokumentace

**Jmeno a prijmeni: jan jjj**

Datum narozeni:: 1972-12-05

Pojistovna: 1255

Patro: 2

Pokoj: 125

**Fyzi**

Vyska: 100 cm

Vaha: 100 Kg

**Pohyb**

Pohyb: Mobilni

Poznamka:

**Vnimani a poznavani**

Potiz se sluchem: -

Pomucka: -

Poznamka:

Obr. 85 Manuál - zobrazení karty

#### 4.6.12 Skupiny léků

Tuto funkci, na kterou se odkazuje z menu, personál použije pro vytvoření vlastní databáze skupin léků a jím přiřazených konkrétních léků. V první obrazovce, na kterou aplikace přeměruje, jsou zobrazeny konkrétní skupiny léků v každém řádku spolu s tlačítkem „zobraz léky“, které zobrazí léky přidružené konkrétní skupině. Pod tabulkou je i možnost přidání nové skupiny. Stačí zadat název a odeslat tlačítkem „přidat skupinu“.

Skupina musí být 3 - 15 znaku		
ID skupiny	Název skupiny	Zobrazení leku
17	tesz	<input type="button" value="zobraz leky"/>
19	Test skupina 2	<input type="button" value="zobraz leky"/>

Nova skupina

Obr. 86 Manuál – skupiny léků

Po kliknutí na tlačítko „zobraz léky“ aplikace uživatele přeměruje na stránku sloužící správě léků. Zde uživatel uvidí všechny léky přidružené skupině. Má možnost přidat nový lék do skupiny nebo některé léky smazat. Je potřeba dbát zvýšené opatrnosti, protože smazáním léku se vymaže nejen tento lék z databáze, ale vymažou se i všechny záznamy tohoto léku ke vztahu ke klientům. Proto je u každého léku přidáné tlačítko „zobraz klienty“, kde se zobrazí všichni klienti, kteří daný lék mají přiřazen v databázi jako aktivní. Uživatel se tak může přesvědčit, jaká data budou smazána.

ID leku	Název leku	Klienti
19	lek2	<input type="button" value="Zobrazit klienty"/>

Novy lek

Obr. 87 Manuál -léky

Prvni: 21

klient	lek	poznámka
Jana hhhhh	test2lek1	2x za den 1/2 prášku
jan Č Čermák	test2lek1	2x za den 1/2 prášku

Obr. 88 Manuál – léky a klienti

#### 4.6.13 Vstup návštěvy

Funkce sloužící k zaevidování návštěvy při vstupu. Uživatel vybere patro, na které chce návštěvník jít a odešle tlačítkem „vybrat patro“. Poté může vyplnit jméno, příjmení. Jméno klienta, za kterým návštěva hodlá jít, se vybere z rozevíracího seznamu na základě vybraného patra. Jméno a příjmení by mělo být kontrolováno podle dokladu, protože kniha návštěv porovnává databázi, zda není návštěvník v evidenci blacklistu. Tlačítkem „vstup návštěvy“ odešle uživatel návštěvu do databáze jako aktivní návštěvu.

Jmeno:

Prijmeni:

Jmeno navstevovaneho klienta

Patro 1

Patro 2

Patro 3

Obr. 89 manuál – návštěva vstup

#### 4.6.14 Aktivní návštěvy

Evidenci pohybujících se osob po objektu může sledovat touto funkcí. Úvodní strana, na kterou aplikace uživatele přesměruje, zobrazí informace o aktivních návštěvách. Z důvodu možného většího počtu dat je zde filtrace podle pater. Uživatel zvolí patro a zobrazí se mu informace o aktivních návštěvách na zvoleném patře. Uživatel má možnost zobrazit

kompletní seznam všech aktivních návštěv, ale jak již bylo řečeno, je tady velká pravděpodobnost většího počtu dat.

Jmeno a prijmeni navstevy	Klient	Pokoj	Cas
Jan Pich	jan Č Čermák	125	29/4 2012 v 15:29:39

[Kompletní seznam](#)

Patro:

- 1
- 2
- 3

Obr. 90 Manuál – aktivní návštěvy

#### 4.6.15 Odchod návštěvy

Při odchodu návštěvy je potřeba návštěvu odepsat z evidence aktivních návštěv a to buď formou bezproblémového odchodu, nebo zařazení návštěvy do blacklistu. Zařazení do blacklistu se může použít v případě, že v průběhu návštěvy se vyskytnou problémy s porušováním návštěvního řádu nebo jiného rázu.

Id navstevy	Jmeno a prijmeni navstevy	Klient	Pokoj	Cas	Pustil	
18	Jan Pich	jan Č Čermák	125	2012-04-29 15:29:39	jan	<input type="checkbox"/>

Obr. 91 Manuál – odchod návštěvy

#### 4.6.16 Black list

Evidence návštěvníků, kteří nějakým způsobem porušili návštěvní řád. Politiky společnosti již určí jak s takovými návštěvníky nakládat. Smazání z blacklistu může provádět pouze oprávněný administrátor.

je rok 2012

Jmeno a prijmeni navstevy	Klient	Pokoj	Cas prichodu	Pustil z personalu
jan pich	jan Č Čermák	125	28/4 2012 v 09:36:51	

Mesic: 01

- 01
- 02
- 03
- 04
- 05
- 06
- 07
- 08
- 09
- 10
- 11
- 12

Obr. 92 Manuál - blacklist



## ZÁVĚR

Závěrem bych rád shrnul poznatky, které vyplynuly z této práce. Ošetřovatelská dokumentace je spolu se zdravotnickou dokumentací základním dokumentem používaným v sociálních zařízeních. Obsahem této dokumentace je klíč, který zdravotníkům odemkne ten správný postup pro péči, nebo v případě krizové situace recept pro záchranu života. Dostupnost této dokumentace by proto měla být co nejrychlejší. Jelikož je tato dokumentace velice obsáhlá a podrobná, je problém v ní rychle vyhledat soubor těch nejdůležitějších informací. Výstupem této práce je aplikace, která shromažďuje souhrn nejdůležitějších informací z těchto dokumentací a dokáže jej zobrazit v mnohem kratší době. Personál tak může zobrazit všechny informace přehledně na jedné stránce. Ty mohou posloužit při krizové situaci, kdy zdravotník potřebuje vědět aktuální stav klienta a není čas na hledání těchto informací v dokumentaci. Uspoří se tak čas, který v určitých situacích je pro klienta otázkou života a smrti.

## ZÁVĚR V ANGLIČTINĚ

Finally, I would like to summarize the findings that emerged from this work. Nursing documentation is medical documentation along with the basic document used in social settings. The content of this documentation is the key that unlocks the health professionals the right procedure for the care, or in case of emergency life-saving recipe. The availability of this documentation should be as fast as possible. Since the documentation is very comprehensive and detailed, it is a problem quickly locate the most important set of information. The outcome of this work is an application that collects the most important summary information from these documents and can be viewed in a much shorter time. Staff can clearly see all the information on one page. You can serve in a crisis situation, when the nurse needs to know the current status of the client and not the time for seeking such information in the documentation. Thus saving time in certain situations, the client matter of life and death.

**SEZNAM POUŽITÉ LITERATURY**

- [1] KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Vyd. 1. Brno: Computer Press, 2007, 607 s. ISBN 978-802-5118-139.
- [2] LAVIN, Peter. *PHP - objektivě orientované: koncepty, techniky a kód*. 1. vyd. Praha: Grada, 2009, 211 s. Průvodce (Grada). ISBN 978-802-4721-378.
- [3] *MySQL profesionálně: optimalizace pro vysoký výkon*. Vyd. 1. Brno: Zoner Press, 2009, 712 s. ISBN 978-807-4130-359.
- [4] CHAMROVÁ, Anna. *Vytvoření ošetrovatelské dokumentace a její využití v Domově pro seniory Chýnov*. České Budějovice, 2010. Bakalářská práce. Jihočeská univerzita. Vedoucí práce Mgr. Ivana Chloubová.
- [5] RAK, Michal. *Ošetrovatelská dokumentace v ČR*. České Budějovice, 2009. Diplomová práce. Jihočeská univerzita. Vedoucí práce Mgr. Helena Michálková.
- [6] KRÝDLOVÁ, Bc. Michaela. *Elektronická dokumentace v ošetrovatelské praxi*. České Budějovice, 2009. Diplomová práce. Jihočeská univerzita. Vedoucí práce Mgr. Lenka Šedová, R.N.
- [7] Vyhláška č. 385/2006 Sb., o zdravotnické dokumentaci
- [8] Vyhláška č. 366/2001 Sb.,
- [9] Zákon č. 20/1966 Sb., o péči o zdraví lidu
- [10] Zákon č. 101/2000 Sb., o ochraně osobních údajů

- [11] Zákon č. 227/2007 Sb., o elektronickém podpisu
- [12] Zákon č. 111/2007 Sb., kterým se mění zákon č. 20/1966 Sb., o péči o zdraví lidu
- [13] IRESOFT. *IC Cygnus* [online]. 2002. vyd. 2002, 2012 [cit. 2012-05-11]. Dostupné z: <http://www.iscygnus.cz/>
- [14] STAPRO S.R.O. *Stapro* [online]. 2002, 2012 [cit. 2012-05-11]. Dostupné z: <http://www.stapro.cz/>
- [15] PAVEL KYSILKA, IČ: 72868490. *Linuxsoft.cz* [online]. 2003, 2012 [cit. 2012]. Dostupné z: <http://www.linuxsoft.cz/>

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

SW	Software – program
HW	Hardware – fyzické prvky IT systému
MySQL	Význam třetí zkratky
PHP	Skriptovací programovací jazyk
Dashboard	Řídící panel, kontrolní panel
HTML	Jazyk pro tvorbu webových stránek
XHTML	HTML jazyk přeformulovaný podle XML syntaxe
XML	Protokol pro usnadnění vzdáleného volání metod
IT	Informační technologie

**SEZNAM OBRÁZKŮ**

<i>Obr. 1 Cygnus – moduly [13]</i> .....	21
<i>Obr. 2 Cygnus – čtečka [13]</i> .....	23
<i>Obr. 3 Cygnus – čtečka [13]</i> .....	24
<i>Obr. 4 Cygnus – dlouhodobý plán[13]</i> .....	24
<i>Obr. 5 čtečka pro docházku [13]</i> .....	28
<i>Obr. 6 FONS Enterprise [14]</i> .....	32
<i>Obr. 7 FONS Enterprise [14]</i> .....	33
<i>Obr. 9 Databáze - Alergie</i> .....	41
<i>Obr. 8 Databáze - blacklist</i> .....	41
<i>Obr. 10 Databáze – druhyleku</i> .....	42
<i>Obr. 11 Databáze – branileku</i> .....	42
<i>Obr. 12 Databáze - fyzio</i> .....	42
<i>Obr. 13 Databáze - historiek1</i> .....	42
<i>Obr. 15 Databáze – historieuz</i> .....	42
<i>Obr. 14 Databáze - historieleky</i> .....	42
<i>Obr. 16 Databáze - navstevy</i> .....	43
<i>Obr. 17 Databáze - kontakt</i> .....	43
<i>Obr. 18 Databáze nemoci</i> .....	43
<i>Obr. 19 Databáze - léky</i> .....	43
<i>Obr. 20 Databáze - odbornici</i> .....	43
<i>Obr. 21 Databáze - operace</i> .....	43
<i>Obr. 22 Databáze - pohyb</i> .....	43
<i>Obr. 23 Databáze - osobavstup</i> .....	43
<i>Obr. 25 databáze - uzivatele</i> .....	44
<i>Obr. 24 Databáze - poznavani</i> .....	44
<i>Obr. 27 Uložení \$_SESSION["id"]</i> .....	46
<i>Obr.28 Grafika aplikace</i> .....	46
<i>Obr. 29 Připojení k DB</i> .....	46
<i>Obr. 30 Funkce pro oprávnění</i> .....	47
<i>Obr. 31 Ověření existence dat v databázi</i> .....	47
<i>Obr. 32 Dynamické zobrazení menu</i> .....	48
<i>Obr. 33 Admin.php</i> .....	48

<i>Obr. 34</i> Zobrazovací funkce.....	50
<i>Obr. 35</i> Nový uživatel – funkce příkazy() .....	50
<i>Obr. 36</i> Nový uživatel – ověření zaškrtačacího formuláře .....	51
<i>Obr. 36</i> Smazání uživatele - skript.....	52
<i>Obr. 37</i> Upravit uživatele – přesměrování na <i>zpracovaniuz.php</i> .....	53
<i>Obr. 38</i> Historie – operace .....	54
<i>Obr. 39</i> Historie – zápis do databáze .....	55
<i>Obr. 40</i> Historie uživatelé – výběr operace .....	55
<i>Obr. 41</i> Historie uživatelé – výběr z DB.....	55
<i>Obr. 42</i> Historie uživatelů – <i>huzseznam.php</i> .....	56
<i>Obr. 43</i> Historie léků – druhy operací.....	56
<i>Obr. 44</i> Vstup návštěvy – naplnění seznamu .....	57
<i>Obr. 45</i> Vstup návštěvy – zpracování formulářů .....	57
<i>Obr. 46</i> Odchod návštěvy – zobrazení návštěv .....	58
<i>Obr. 47</i> Odchod návštěvy – zpracování odeslaného formuláře.....	58
<i>Obr. 48</i> Aktivní návštěvy – zobrazení .....	59
<i>Obr. 49</i> Proběhlé návštěvy – plnění formulářů .....	59
<i>Obr. 50</i> Proběhlé návštěvy – kontrola zadaných dat .....	60
<i>Obr. 51</i> Proběhlé návštěvy – vypsání z databáze .....	60
<i>Obr. 52</i> Nový klient – kontrola vstupů .....	61
<i>Obr. 53</i> Nový klient – zápis do databáze .....	61
<i>Obr. 54</i> Editace klienta – zobrazení klientů dle oprávnění .....	62
<i>Obr. 55</i> <i>Zpracovani.kl.php</i> - jednotlivé odkazy .....	63
<i>Obr. 56</i> Editace klienta – zobrazení při editaci karty.....	64
<i>Obr. 57</i> <i>Nemoci.php</i> – více řádků .....	65
<i>Obr. 58</i> <i>Nemoci.php</i> – cyklus zpracování .....	65
<i>Obr. 59</i> Zobrazení klienta – abecední vyhledávání .....	66
<i>Obr. 60</i> Zobrazení klienta – vybrání podle abecedy.....	67
<i>Obr. 61</i> Zobrazení klienta – výpis z databáze .....	68
<i>Obr. 62</i> Zobrazení klienta – výběr z databáze pomocí formulářů .....	68
<i>Obr. 63</i> Zobrazení klienta – celkový pohled .....	69
<i>Obr. 64</i> Zobrazení klienta – opakující se informace .....	70
<i>Obr. 65</i> Zobrazení klienta – přepínače .....	70

<i>Obr. 66</i> Manuál - přihlášení .....	71
<i>Obr. 67</i> Manuál - menu.....	72
<i>Obr. 68</i> Manuál – základní vzhled.....	73
<i>Obr. 69</i> Manuál – jednotlivé menu .....	73
<i>Obr. 70</i> Manuál – správa uživatelů .....	74
<i>Obr. 71</i> Manuál – editace uživatele.....	75
<i>Obr. 72</i> Menu – mazání uživatele .....	76
<i>Obr. 73</i> Manuál – nový uživatel.....	77
<i>Obr. 74</i> Manuál - historie .....	77
<i>Obr. 75</i> Manuál – Historie uživatelé .....	78
<i>Obr. 76</i> Manuál – Historie uživatelé seznam .....	78
<i>Obr. 77</i> Manuál – historie klienti zobrazení.....	79
<i>Obr. 78</i> Manuál - historie léky.....	79
<i>Obr. 79</i> Manuál - nový klient.....	80
<i>Obr. 80</i> Manuál – dokončení registrace .....	81
<i>Obr. 81</i> Manuál – editace klienta .....	82
<i>Obr. 82</i> Manuál – smazání klienta.....	82
<i>Obr. 83</i> Manuál – zobrazení klientské dokumentace .....	83
<i>Obr. 84</i> Manuál – zobrazení dokumentace .....	84
<i>Obr. 85</i> Manuál - zobrazení karty .....	84
<i>Obr. 86</i> Manuál – skupiny léků.....	85
<i>Obr. 87</i> Manuál -léky.....	85
<i>Obr. 88</i> Manuál – léky a klienti .....	86
<i>Obr. 89</i> manuál – návštěva vstup.....	86
<i>Obr. 90</i> Manuál – aktivní návštěvy .....	87
<i>Obr. 91</i> Manuál – odchod návštěvy .....	87
<i>Obr. 92</i> Manuál - blacklist.....	88



## SEZNAM TABULEK

## SEZNAM PŘÍLOH

1. Hodnocení aplikace personálem pečovatelského domu pro seniory.

**PŘÍLOHA P I: NÁZEV PŘÍLOHY**