

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

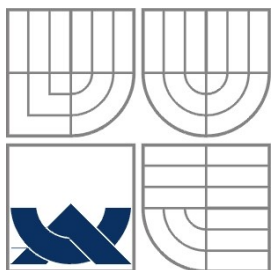
INFORMAČNÍ SYSTÉM REVIZNÍ FIRMY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

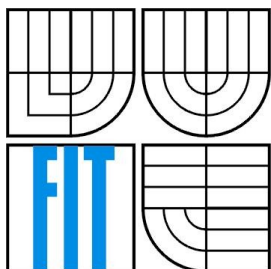
AUTOR PRÁCE
AUTHOR

Bc. RADEK PINKAVA

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INFORMAČNÍ SYSTÉM REVIZNÍ FIRMY

INFORMATION SYSTEM OF THE SERVICE COMPANY

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. RADEK PINKAVA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PAVEL OČENÁŠEK

BRNO 2007

Abstrakt

Diplomová práce se zabývá problematikou návrhu a implementace informačního systému pro firmu HNS mechanix s. r. o. Tato společnost provádí revizní činnost optických výrobků. Při návrhu se vychází z konkrétních požadavků nejen na funkčnosti a vlastností informačního systému, ale také na použité implementační nástrojů. Je vyžadováno, aby byl systém dostupný pomocí internetové služby World Wide Web a poskytoval multilevel rozlišení uživatelů.

Klíčová slova

Informační systém, WWW (World Wide Web), XHTML (eXtensible Hypertext Markup Language), jazyk PHP, databáze, MySQL, kaskádové styly CSS (Cascading Style Sheets).

Abstract

This diploma thesis covers a solution of designing the information system for HNS mechanix ltd.

This company provides auditorial services of optical products.

During the desining not only the requirements of functionality and attributes of the system itself but also implementation and developing tools are necessary to be followed.

The system is required to be available as WWW service with multilevel user access.

Keywords

Information system, WWW (World Wide Web), XHTML (eXtensible Hypertext Markup Language), PHP language, database, MySQL, CSS (Cascading Style Sheets).

Citace

Pinkava Radek: Informační systém revizní firmy. Brno, 2007, diplomová práce, FIT VUT v Brně.

Informační systém revizní firmy

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Pavla Očenáška. Další informace mi poskytl Ing. Martin Honeš (externí konzultant – zástupce firmy HNS mechanix s.r.o.).

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Radek Pinkava
22. května 2007

Poděkování

Děkuji tímto vedoucímu svého semestrálního projektu panu Ing. Pavlu Očenáškov, za odborné vedení a konzultace.

Dále bych chtěl poděkovat externímu zadavateli projektu Ing. Marku Honešovi za ochotu při konzultacích a za správné směřování práce.

© Radek Pinkava, 2007.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	4
2 IS v prostředí internetu.....	6
2.1 Architektura informačního systému.....	6
2.2 XHTML.....	7
2.2.1 Historie.....	7
2.2.2 Zdůvodnění volby.....	8
2.3 XML.....	8
2.4 PHP.....	9
2.4.1 Historie.....	9
2.4.2 Zdůvodnění volby.....	9
2.5 MySQL.....	10
2.5.1 Historie.....	10
2.5.2 Zdůvodnění volby.....	10
2.6 CSS.....	10
2.7 JavaScript.....	10
2.7.1 AJAX.....	11
2.8 Apache.....	11
2.9 UML.....	11
3 Analýza požadavků aplikace.....	12
3.1 Společnost HNS mechanix s. r. o.....	12
3.1.1 O společnosti.....	12
3.1.2 Popis činnosti laboratoře.....	12
3.2 Stávající systém.....	13
3.3 Požadavky na nový systém.....	14
4 Specifikace požadavků.....	17
4.1 Model případů použití.....	17
4.1.1 Aktér Uživatel.....	18
4.1.2 Aktér Klient.....	18
4.1.3 Aktér Zaměstnanec.....	19
4.1.4 Aktér Technik.....	21
4.1.5 Aktér Administrátor.....	21
4.1.6 Aktér Ředitel.....	23
4.2 Zpracovávaná data laboratoře.....	23

4.2.1 Typy tonometrů.....	24
4.2.2 Typy formulářů.....	24
4.2.3 Požadavky na zpracování.....	24
5 Návrh aplikace.....	25
5.1 Architektura aplikace.....	25
5.2 Model ERD.....	25
5.2.1 Popis entit diagramu.....	26
5.2.2 Popis vazeb diagramu.....	28
5.3 Návrh uživatelského rozhraní.....	28
5.3.1 Layout.....	28
5.3.2 Navigace.....	29
6 Implementace aplikace.....	30
6.1 Struktura aplikace.....	30
6.2 Moduly systému.....	30
6.2.1 Uživatelé.....	31
6.2.2 Tonometry.....	33
6.2.3 Revize.....	33
6.2.4 Nastavení.....	34
6.2.5 Testy.....	35
6.2.6 Přehledy.....	35
6.2.7 Dokumenty.....	36
6.3 Aplikační logika.....	36
6.4 Použité techniky a rozšíření.....	37
6.4.1 Databáze.....	37
6.4.2 Šablonovací systém.....	38
6.4.3 Navigace.....	38
6.4.4 HTML formuláře.....	39
6.4.5 Výstupy aplikace.....	40
6.4.6 Export dat.....	41
6.5 Uživatelské rozhraní.....	41
6.5.1 Layout a design.....	41
6.5.2 Navigace.....	42
6.6 Zpracování dat laboratoře.....	43
6.7 Zabezpečení aplikace.....	44
6.7.1 Přihlašování.....	44
6.7.2 Uložení hesel.....	44
6.7.3 Zabezpečení stránek.....	44

6.7.4 Ošetření vstupů.....	45
6.7.5 Další zabezpečení.....	45
6.7.6 Zhodnocení zabezpečení.....	46
7 Rozšiřitelnost aplikace.....	47
7.1 Požadavky.....	47
7.2 Diskuze rozšíření.....	47
8 Závěr.....	48
Literatura.....	49
Seznam použitých zkratk.....	50
Seznam příloh.....	51
Příloha 1 – formuláře AMS.....	52
Příloha 2 – Uživatelská příručka.....	55
Příloha 3 – obsah přiloženého CD.....	65

1 Úvod

Počítače a informační technologie (IT) stále více ovlivňují lidskou společnost. Jejich používání je stále komplexnější a klade stále větší důraz na ukládání, vyhledávání a prezentaci informací. Základním prvkem využívání IT jsou softwarové produkty pokrývající tuto činnost – informační systémy. Informační systémy (IS) jsou základním nástrojem změn organizace práce a zlepšení kvalitativních charakteristik podniků a organizací. Kvalita informačních systémů, rozsah a míra jejich využívání do značné míry rozhoduje o úspěchu dané společnosti.

Cílem této diplomové práce bylo navrhnout a implementovat informační systém pro společnost HNS mechanix s.r.o., která se zabývá revizní činností optických výrobků. Stávající aplikace totiž nepokrývala potřebné požadavky na ukládání dat a nenabízela svým uživatelům dostatečnou funkcionalitu a prostředky pro správu informací. Z toho důvodu bylo nutné navrhnout nový informační systém, který tyto nedostatky odstraní. Důraz byl především kladen na vhodné ucelení systému, centralizaci uložených dat a zajištění jejich lepší přístupnosti. Od nasazení nového IS se předpokládá vyšší stupeň využití IT v laboratořích AMS (autorizované metrologické středisko firmy HNS mechanix s.r.o.), zefektivnění práce, rozšíření možností správy apod. S tím také souvisí vylepšení a rozšíření poskytovaných služeb zákazníkům. Analýzy a specifikace požadavků projektu byla zpracována v rámci semestrálního projektu, na který navázala fáze detailního návrhu a implementace.

Tento dokument je rozdělen do osmi základních, tematicky oddělených kapitol. První část se zabývá rozborem a odůvodněním výběru pracovních nástrojů, technologií a programovacích jazyků, které budou použity k implementaci navrženého informačního systému. Podrobněji jsou zde popsány především použité technologie pro tvorbu uživatelského rozhraní XHTML spolu se skriptovacím jazykem PHP a databázovým systémem MySQL.

V další kapitole je uvedena analýza požadavků na vytvářenou aplikaci. Je zde představena společnost HNS mechanix s.r.o., obor její působnosti, kvalita i úroveň poskytovaných služeb. Zmíněna je činnost laboratoře AMS, její účel a dále jsou uvedeny její pracovní postupy. Následuje stručný popis systému, který je aktuálně v HNS mechanix s.r.o. používán, jsou zhodnoceny jeho vlastnosti, výhody a nevýhody. Dále jsou neformálně popsány funkční i nefunkční požadavky, které jsou kladeny na nově navrhovaný informační systém.

V rámci čtvrté kapitoly je zmíněna podrobná specifikace požadavků na vyvíjený informační systém. Pro potřeby znázornění a modelování funkcí aplikace je uveden diagram případů použití z notace jazyka UML. Následuje rozbor formulářů a dokumentů laboratoře AMS, které musí být v aplikaci zpracovány.

Pátá část tohoto dokumentu se zabývá detailním návrhem vyvíjeného informačního systému. Nejprve je zmíněna architektura aplikace a její vlastnosti. S využitím nástrojů UML je dále uveden

konceptuální logický model dat v podobě diagramu ERD. Díky tomuto modelovacímu prostředku je schématicky znázorněna struktura databáze, která bude v systému použita. Následuje návrh grafického uživatelského rozhraní aplikace.

Šestá kapitola obsahuje popis procesu implementace vyvíjeného informačního systému. V první části je znázorněna struktura aplikace a jejích jednotlivých součástí. Poté jsou jednotlivé moduly podrobně popsány včetně jejich funkcí a vlastností. Jsou zde zmíněny konkrétní techniky a rozšíření, které byly při implementaci použity a jejich stručný popis. Následuje znázornění zpracování uživatelského rozhraní a jeho jednotlivých prvků. Poté je popsána realizace zpracování a uložení dat laboratoře AMS. V závěrečné části této kapitoly je formulována a diskutována otázka zabezpečení vyvíjeného informačního systému. Vedle popisu jsou zde uvedeny návrhy, které by vedly ke zvýšení bezpečnosti aplikace.

V sedmé kapitole jsou zmíněny možnosti rozšíření informačního systému. Dále jsou diskutovány konkrétní nápady pro rozšíření a další pokračování tohoto projektu.

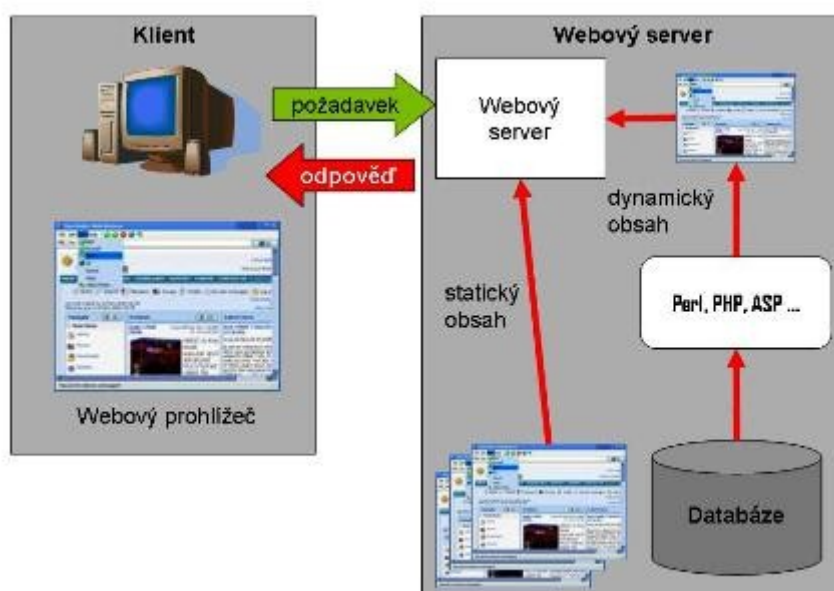
Závěr práce obsahuje zhodnocení provedených etap, nastiňuje budoucí vývoj realizovaného projektu a shrnuje jeho celkový přínos.

2 IS v prostředí internetu

Tato kapitola se nejprve zabývá obecnou strukturou informačního systému v prostředí Internetu. Jsou zde stručně popsány technologie a prostředky, které budou při řešení diplomové práce využity. Mimo jiné je uvedena jejich stručná historie, zhodnocení a zdůvodnění výběru konkrétního prostředku či technologie.

2.1 Architektura informačního systému

V dnešní době již převládá nejen v prostřední síti Internetu třívrstvá architektura komunikace klient / server. Nejinak je tomu u vyvíjeného informačního systému. Ten je konstruován jako internetová aplikace, která bude spuštěna na webovém serveru Apache (viz. kap. 2.8) s nainstalovaným modulem PHP (viz. kap. 2.4). Webový server komunikuje s databázovým serverem se systémem MySQL (viz. kap. 2.5). Pomocí skriptovacího jazyka PHP se na straně webového serveru dynamicky generují WWW stránky aplikace, které vytvářejí grafické uživatelské rozhraní informačního systému. K těmto stránkám, které jsou v podobě XHTML dokumentů (viz. kap. 2.2), uživatelé přistupují pomocí svých webových prohlížečů. Rozvržení a grafický design uživatelského rozhraní aplikace jsou popsány pomocí kaskádových stylů CSS (kap. 2.6). Validace zadaných dat ve formulářích, změny zobrazení a jednoduché operace, které mohou být vyřešeny na straně klienta bez nutné komunikace se serverem, jsou naprogramovány pomocí jazyku JavaScript (kap. 2.7).



Obrázek 2-1 Schéma IS v síti Internet (převzato z (1))

2.2 XHTML

2.2.1 Historie

Historie značkovacích jazyků pro vytváření webových stránek začala specifikací HTML 2.0, kterou vydala internetová standardizační organizace Internet Engineering Task Force (IETF) v roce 1995 jako dokument RFC¹. Tento jazyk byl podmnožinou SGML, univerzálního značkovacího jazyka.

Specifikace HTML 2.0 neobsahovala příliš mnoho prvků k vytváření pohledných grafických stránek, a tak byla hlavně kvůli tlaku ze strany společností Netscape a Microsoft uvolněna specifikace HTML 3.2. V této verzi byly bohužel do jazyka začleněny mnohé prvky, které naprosto postrádají strukturální význam a slouží pouze k definici vzhledu. Tuto specifikaci již vydala webová standardizační organizace Webové konsorcium - W3C (World Wide Web Consortium). W3C se snaží vést web k tomu, aby byl přístupný z nejrůznějších zařízení a ne pouze z osobních počítačů. Toho lze nejlépe dosáhnout pomocí strukturálních prvků, nikoliv zmiňovaných prvků vzhledových, které jsou specifické pro konkrétní zařízení a kterých bylo HTML 3.2 plné (2).

Proto W3C vydalo další specifikaci - HTML 4.0. Zde definuje poměrně mocný jazyk pro definici struktury dokumentu, vzhled dokumentu nechává téměř výhradně na CSS (popsáno v kap. 2.6). Webové konsorcium chtělo zároveň nechat výrobcům prohlížečů a autorům stránek čas k přechodu na tuto normu, proto rozdělilo HTML 4.0 na: HTML 4.0 Strict; HTML 4.0 Transitional – přechodná norma, zachovává vzhledové atributy HTML 3.2 a zároveň obsahuje všechny výhody a vylepšení HTML 4.0 Strict, především v oblasti přístupnosti a CSS; HTML 4.0 Frameset – tato norma se téměř shoduje s HTML 4.0 Transitional, je pouze rozšířena o elementy pro definici rámu (jedná se o možnost vkládat více HTML dokumentů do jednoho) (2).

Specifikace HTML 4.0 byla později revidována, byly opraveny některé chyby a stala se z ní specifikace HTML 4.01. Tato verze obsahuje kompletní popis jazyka a všechny následující specifikace jazyků pro tvorbu webu, které dosud vyšly, se na ni odkazují.

V další fázi bylo potřeba definovat HTML jako podmnožinu jazyka XML (popsáno v kap. 2.3), který vychází ze SGML, avšak je odlehčen o některé zbytečné vlastnosti a některé nové vlastnosti naopak přidává. W3C prosazuje XML jako hlavní a jediný značkovací jazyk nejen pro web, a proto byla reformulace HTML do XML logickým vyústěním této snahy. Nově vytvořený jazyk dostal jméno XHTML (eXtensible HyperText Markup Language - rozšiřitelný hypertextový značkovací jazyk) a jeho první verze se označuje XHTML 1.0. Tato specifikace je stejná jako specifikace HTML 4.01, jsou zde pouze integrována pravidla XML (2).

¹ RFC (request for comments) - označení řady standardů a dalších dokumentů popisujících Internetové protokoly, systémy, specifikace apod.

W3C chce XHTML prosazovat jako jediný jazyk pro definici webových stránek, bez ohledu na typ zařízení, které je zpracovává. Expanze webu na různá alternativní zařízení ukázala, že tato zařízení nemohou kvůli svým omezeným možnostem podporovat všechny vlastnosti XHTML a že podporují jen nějakou jeho část. Tyto podmnožiny bylo třeba definovat a standardizovat, proto vznikla specifikace Modularization of XHTML, která rozděluje všechny prvky XHTML 1.0 do modulů, ze kterých se následně skládají značkovací jazyky. Skupiny zabývající se webem v alternativních zařízeních mohou definovat nové moduly s prvky specifickými pro dané zařízení, mohou stávající moduly modifikovat, ale hlavně z nich mohou skládat nové kompletní značkovací jazyky, které vyhovují jejich potřebám a možnostem interpretace webu. Z těchto jazyků zatím W3C uznalo XHTML 1.1 (až na několik detailů se shoduje s XHTML 1.0 Strict, hlavní rozdíl spočívá v tom, že XHTML 1.1 je definováno pomocí modulů) a XHTML Basic (skládá se pouze ze základních modulů, k použití hlavně na mobilních telefonech, PDA a podobně) (2).

2.2.2 Zdůvodnění volby

Jazyk XHTML zvolen při řešení informačního systému proto, že nejlépe vyhovuje požadavkům na implementaci neexistuje mnoho jeho alternativ. V kombinaci s CSS (viz. kap. 2.6) splňuje požadavky jak z hlediska správné strukturovanosti dokumentu, tj. umožňuje odloučení vzhledu od vlastního obsahu, tak z hlediska jeho čistoty a validity. Používání tohoto jazyka je velmi rozšířeno, přestože jeho podpora webovými prohlížeči zatím neumožňuje plné využití všech jeho vlastností. Konkrétně ve vyvíjeném systému bude použita specifikace XHTML 1.0 strict, která se maximálně přibližuje zatím stále nejrozšířenějšímu formátu WWW dokumentů verzi HTML 4.01.

2.3 XML

XML (eXtensible Markup Language - rozšiřitelný značkovací jazyk) je obecný značkovací jazyk, který byl vyvinut a standardizován konsorciem W3C. Umožňuje snadné vytváření konkrétních značkovacích jazyků pro různé účely a široké spektrum různých typů dat.

Jazyk je určen především pro výměnu dat mezi aplikacemi a pro publikování dokumentů. XML umožňuje popsat strukturu dokumentu z hlediska věcného obsahu jednotlivých částí, nezabývá se sám o sobě vzhledem dokumentu nebo jeho částí. Prezentace dokumentu (vzhled) se potom definuje připojeným stylem. Velkou výhodou XML získává tím, že se data v něm uložená dají lehce konvertovat do jiných formátů nebo jiné struktury a bezproblémově se tedy dají využít i v databázích. Z vyjmenovaných výhod plyne, že se dá XML výhodně použít především pro uchovávání a výměnu dat, a proto byla tato metoda uložení upřednostněna i v této aplikaci.

2.4 PHP

2.4.1 Historie

PHP (v dnešní době označováno celým názvem: PHP Hypertext preprocessor) je skriptovací jazyk na straně serveru pro tvorbu dynamického webu a jeho počátky spadají do roku 1994. Rasmus Lerdorf vytvořil jednoduchý systém pro počítání přístupů ke svým stránkám v jazyce PERL, ale systém kvůli velké náročnosti a zatěžování serveru přepsal do jazyka C. Sada těchto skriptů byla ještě později téhož roku rozšířena a vydána pod názvem „Personal Home Page Tools“, zkráceně PHP. Později byl název upraven „Personal Home Page Construction Kit“ (3).

V polovině roku 1995 se systém PHP spojil s jiným programem stejného autora — nástrojem „Form Interpreter“ (FI), který umožňoval začleňování dotazů jazyka SQL do stránek, vytváření formulářů a zobrazování výsledků dotazů. Tak vzniklo PHP/FI 2.0, systém, který si postupně získal celosvětovou proslulost a byl velmi rozšířen (3).

Koncem roku 1998 byla již k dispozici verze PHP 3.0, která byla mnohem rychlejší a vybavenější a byla k dispozici rovněž pod operačními systémy Windows. Jednou z nejsilnějších zbraní PHP 3.0 byly jeho obrovské možnosti rozšíření. Mnoho vývojářů tak mohlo vytvářet rozšiřující moduly pro mnoho různých databází a protokolů. Jiným klíčovým prvkem byla podpora objektivě orientované syntaxe a mnohem silnější a konzistentnější syntaxe jazyka (3).

Počet webů které používaly PHP se stále zvyšoval, a proto se nezastavil ani vývoj čtvrté verze PHP. Do jazyka přibylo opět mnoho nových funkcí a rovněž přepracované a tudíž podstatně rychlejší jádro, které dostalo označení Zend. Cílem návrhu nové verze bylo zvýšit výkon pro složité aplikace a zlepšit modularitu kódové báze. PHP 4.0 bylo oficiálně uvolněno v roce 2000 a přibyly i další vlastnosti jako podpora pro mnoho WWW serverů, HTTP sessions, bezpečnější způsoby zpracování výstupu uživatele a další (3).

V polovině roku 2004 byla uvolněna další verze PHP 5.0 vystavěná na novém jádru Zend 2 s novým objektivním modelem. Je zde kompletně přepracovaná podpora XML, nové objektivní rozhraní pro práci s databází MySQL a mnoho dalšího (4).

2.4.2 Zdůvodnění volby

Jazyk PHP má řadu výhod. Jde zejména o jeho blízkost a podobnost programovacímu jazyku C++, relativní jednoduchost a srozumitelnost. Další výhodou PHP je to, že je šířen pod volnou (Open Source) licenci a je tedy zdarma. O volbě použití tohoto jazyka při zpracování informačního systému rozhodl rovněž požadavek ze strany zadávající firmy HNS mechanix s.r.o., která v PHP provozuje stávající řešení, které bude nahrazeno novým IS. Úzká vazba PHP k databázovému systému MySQL,

který je rovněž ve stávajícím systému využit, jen potvrdila tuto volbu. Během implementace je použita verze 5.1.4, která dovoluje použití potřebných OOP technik.

2.5 MySQL

2.5.1 Historie

MySQL je otevřený systém správy relačních databází podnikové úrovně. V roce 1996 jej vyvinula švédská společnost TcX pro potřebu rychlého a pružného databázového systému. Tento produkt byl rychlý, spolehlivý a neobyčejně pružný. MySQL začal být od své verze 3.23 šířen zdarma v souladu s pravidly GNU (General Public Licence), tj. včetně zdrojových kódů. Kromě toho, že neklade na uživatele finanční nároky, je další nespornou výhodou MySQL nezávislost na operačním systému, což se o konkurenčních produktech říci nedá. Nejrozšířenější verze MySQL má nyní pořadové číslo 5.0 a stále jsou přidávány nové vlastnosti a funkce (5).

2.5.2 Zdůvodnění volby

Databázový systém MySQL byl vybrán z několika důvodů. Tento systém je společností HNS mechanix s.r.o. využíván ve stávajícím řešení IS, a proto bylo vhodné toto řešení zachovat. Důvodem byla i cena produktu, která je nulová, a již zmíněná provázanost s výše jmenovaným jazykem PHP. MySQL plně postačuje nárokům, které vytvářený informační systém vyžaduje, je navíc v dnešní době velmi rozšířený a oblíbený. Při implementaci je použita aktuální verze 5.0.22.

2.6 CSS

Kaskádové styly (CSS) jsou nedílnou součástí XHTML dokumentů a popisují jeho vzhled a formátování. Přináší i mnoho výhod, především širší formátovací možnosti, snadnou tvorbu a údržbu konzistentního stylu celé aplikace. Oddělují především strukturu a styl dokumentu, čímž se zvyší přístupnost a přehlednost vlastních dokumentů. Pro všechny tyto vlastnosti je použití kaskádových stylů v informačním systému nezbytné. V aplikaci je použita specifikace CSS verze 2.0, která je nyní nejvíce používána a je podporována ze strany nejrozšířenějších internetových prohlížečů.

2.7 JavaScript

JavaScript je interpretovaný programovací jazyk. Program v něm napsaný je klientský skript, což znamená, že se jeho kód odesílá se stránkou na klienta (do prohlížeče) a teprve tam je vykonáván. Těchto vlastností se v systému využívá k validaci formulářů, jednoduchým výpočtům a dynamickému zobrazování částí dokumentu, kde není nutná komunikace se serverem.

2.7.1 AJAX

AJAX (Asynchronous JavaScript and XML) je obecné označení pro technologie vývoje interaktivních webových aplikací, které mění obsah svých stránek bez nutnosti jejich znovunačítání. Poskytují uživatelsky příjemnější prostředí, ale na druhou stranu vyžadují použití moderních webových prohlížečů. Použitím této technologie došlo ke zlepšení některých operací, které nabízí vyšší komfort při práci s vyvíjenou aplikací.

2.8 Apache

Apache je dlouhodobě nejrozšířenější webový server, který je možno provozovat na různých platformách operačních systémů. Díky jednoduché konfiguraci a malým systémovým nárokům na operační systém se objevuje nejen na platformě Unix/Linux, ale i na počítačích se systémy Microsoft Windows (6). Jeho hlavní výhoda spočívá v tom, že je šířen zcela zdarma jak pro soukromé, tak komerční použití. To byl jeden z hlavních důvodů výběru tohoto produktu pro tento projekt. Použitá verze je Apache 2.0.

2.9 UML

UML (Unified Modeling Language) je v softwarovém inženýrství grafický jazyk pro vizualizaci, specifikaci, navrhování a dokumentaci programových systémů. UML nabízí standardní způsob zápisu návrhů systému včetně konceptuálních prvků jako jsou business procesy a systémové funkce. Nabízí rovněž prostředky ke znázornění konkrétních prvků systémů jako jsou příkazy programovacího jazyka, databázová schémata a znovupoužitelné programové komponenty (7). Tento jazyk byl použit pro modelování a znázornění požadavků a při analýze a návrhu vyvíjené aplikace.

3 Analýza požadavků aplikace

V této kapitole jsou uvedeny základní informace o společnosti HNS mechanix s.r.o., která je zadavatelem projektu. Nejprve je daná společnost stručně charakterizována a poté jsou popsány činnosti její laboratoře (AMS), které jsou výchozími pilíři při navrhování systému. Jsou zde shrnuty požadavky, které byly na systém vzneseny ze strany zadavatele.

3.1 Společnost HNS mechanix s. r. o.

3.1.1 O společnosti

Informace uvedené v této kapitole jsou převzaty z (7). Firma HNS mechanix s.r.o. vznikla v roce 1993. Je zaměřena na vývoj, výrobu a prodej přístrojů pro oční optiky. V současné době nabízí téměř kompletní technické vybavení pro oční dílnu a je největším výrobcem broušících a leštících přístrojů v České republice.

Kromě produktů pro optiky vyrábí přístrojovou techniku určenou pro široký okruh zájemců jako: ultrazvukové čističky s množstvím různých typů van, vrtací-frézovací přístroje aj. Společnost v roce 1996 rozšířila působnost na metrologické služby. Disponuje laboratoří autorizovaného metrologického střediska AMS. V souladu s ustanovením §19 zákona o metrologii č.505/1990 Sb. v platném znění je registrována pro servis mechanických očních tonometrů podle nejnovějších nařízení českých norem a předpisů EU (ověřování jakosti dle ČSN EN ISO 10012 a požadavků ČSN EN ISO/IEC 17025).

Společnost zajišťuje veškeré opravy, seřízení a úřední ověřování všech typově schválených Schiötzových a aplanačních Goldmannových očních tonometrů. Laboratoř HNS mechanix s.r.o. jako autorizované metrologické středisko (AMS) disponuje certifikací, profesionálními pracovníky s osvědčením o odborné způsobilosti. Pro výkon ověřování těchto měřidel je na pracovišti AMS k dispozici jednak speciální měřicí zařízení pro ověřování aplanačních Goldmannových tonometrů, a také měřicí zařízení pro ověřování impresních Schiötzových tonometrů.

AMS zajišťuje metrologické služby pro všechny státní i privátní ordinace a tím jednotnost, přesnost a spolehlivost měření nitroočního tlaku očními tonometry. Všechny oční tonometry typu Schiötz a Goldmann se kontrolují a ověřují přesně podle závazných příslušných metodik schválených ČMI, ÚNMZ, které jsou validovány a verifikovány.

3.1.2 Popis činnosti laboratoře

Činnost laboratoře autorizovaného metrologického střediska (AMS) je postavena na úředním ověřování a servisu mechanických očních tonometrů. Vychází ze zákonných nařízení definujících

povinnost ověřovat oční tonometry všem jejich provozovatelům, tj. soukromým či státním očním ordinacím. Laboratoři jsou průběžně z těchto zařízení zasílány oční tonometry, práce je relativně stálá po celý rok.

Zákonná lhůta ověření je stanovena na 1 rok (dle vyhlášky Ministerstva průmyslu a obchodu ČR č.345/2002 Sb.pol.2.3.1a) přílohy ve znění vyhlášky č.65/2006 sb.), poté je nutné znovu provést test (kalibraci) přístroje. Posloupnost procesů v laboratoři je ve svém principu jednoduchá a přesně definovaná.

Oční tonometry doručené na adresu AMS prochází vstupní prohlídkou, na jejímž základě jsou následně rozděleny do skupin na ověření, kdy nejsou dále opravovány, a opravu po jejímž dokončení dochází k ověření. Laboratorní procesy lze rozdělit na dva základní proudy činností - agenda a měřicí úkony. Tyto činnosti jsou navzájem provázány v tom smyslu, že agenda je tvořena různými zápisy o tonometrech před, v průběhu i po proměření. Je nutné provést zápis o vstupním testu, zápis o zkoušce (protokol o měření) a následně vydat potvrzení ověření platné 12 měsíců. V případě, že se tonometr nepodaří opravit a docílit tak naměření požadovaných hodnot, je vydáno rozhodnutí o zamítnutí. Po dokončení popsáných úkonů jsou tonometry spolu s výsledky měření odeslány zpět klientovi.

3.2 Stávající systém

V současné době se v AMS nepoužívá žádný centrální systém, který by prováděl správu zákazníků, revizí, atd. Dokumenty jsou vesměs zpracovávány nezávisle na databázi, formou jednoduchých protokolů v textovém editoru. Jde např. o předvyplněné formuláře, do kterých se později dopisují naměřené hodnoty apod. Informace o výsledcích provedených měřeních se později přepisují do jednoduché tabulky databáze, se kterou pracuje několik jednoduchých PHP skriptů. Ty ale poskytují pouze základní funkce jsou zobrazení, přidání a mazání uložených dat.

Výhodou tohoto řešení je jeho relativní jednoduchost a srozumitelnost. Neklade žádné velké požadavky na obsluhu, její postupy jsou již zaběhlé. Malé nároky na zaměstnance jsou rovněž přínosem.

Mezi jednu z hlavních nevýhod stávajícího aplikace patří absence jakéhokoli systému zabezpečení. K systému a všem jejích funkcím (včetně editace a mazání) má přístup kdokoli. Mezi další nevýhody patří to, že už je systém v dnešní době poněkud zastaralý a těžkopádný. Je neefektivní a neposkytuje řadu dnes již vyžadovaných funkcí. Velkou nevýhodou aplikace je rovněž to, že neposkytuje žádné možnosti vzdálené kontroly nad procesy prováděné v laboratoři. Systém neumožňuje filtrování klientů dle zadaných kritérií, jejich správu a je bez jakýchkoli tiskových výstupů apod. Rovněž schéma databáze, které je použito, nese mnohé nevýhody a obsahuje nejen redundance, ale i neatomické hodnoty apod.

3.3 Požadavky na nový systém

Požadavky na funkčnost systému vychází z výše uvedeného postupného procesu zpracování zakázky (revize). Systém by měl poskytovat prostředky pro správu jednotlivých kroků tak, jak jdou za sebou, a provést automatické vyhodnocování určitých úkonů. Rovněž databáze nového systému by měla být vhodně navržena tak, aby umožnila efektivní uložení většího počtu záznamů a neobsahovala žádné redundance a nekonzistence.

Zmíněným pojmem „správa“ je dále v tomto dokumentu myšleno zadávání či odebrání záznamů a jejich případná editace. Používaný termínem „revize“ je označen proces přijetí tonometru, vyplnění vstupního testu a záznamu o zkoušce včetně příslušných měření a následné vygenerování ověření.

Jednou z hlavních funkcí IS by tak měla být možnost zobrazení těch tonometrů (klientů), kterým uplynula lhůta ověření a možnost automaticky generovat upozornění na tuto skutečnost (např. dopis, email, apod.). S tím je spojena i funkčnost IS se „statistickým“ modulem, který by umožňoval zobrazovat klienty, jejich tonometry atd. dle nejrůznějších filtrů (např. ověření v daném roce, z dané nemocnice, z daného města, z určitého časového období, atp.).

Dále by měl systém umožnit správu klientů, aktuálních stavů tonometrů a jednotlivých ověření, generování reportů na základě předem zvolených filtrů, vytváření tiskových šablon (např. pro proší tonometry, atd.) apod.

Vyjmenované požadavky by měl systém poskytovat uživateli na základě toho, v jaké roli vystupuje. Měla by být odlišena klientská část od části, kam mají přístup zaměstnanci. Mělo by být umožněno odlišit jednu osobu, které bude umožněno nastavení systému a dále jednu osobu, která bude mít přístup ke sledování některých údajů o ostatních zaměstnancích.

Souhrnně lze požadavky na aplikaci shrnout následovně:

Nefunkční požadavky:

- minimální náročnost na uživatele aplikace a zároveň rychlé zaučení – určena pro techniky se znalostí IT na úrovni uživatele začátečníka
- rozhraní aplikace řešit „user-friendly“
- možnost současného přístupu do aplikace z více počítačů (nejen v jedné síti) – využití IT internetového přístupu
- implementace aplikace v jazyku PHP s využitím databáze MySQL.

Funkční požadavky obecně:

- řešení aplikace z víceúrovňového pohledu (administrátor, uživatel, atd.)
- správa klientů

- správa tonometrů
- správa zaměstnanců
- správa provedených a stávajících revizí a jejich součástí
- standardní možnosti správy databáze - vyhledávání, filtrování záznamů
- možnosti statistik a reportů - např. za zvolené časové období zjistit počet obslužených klientů, tonometrů, ověření, atd.
- tiskové výstupy laboratoře (protokoly o ověření, apod.)
- odlišení sekcí pro klienty a různé role zaměstnanců.

Základní funkční moduly systému:

- zpracování aktuálních záznamů o revizi s tiskovými výstupy. Jsou to dílčí části:
 - zadání základních údajů k revizi poté, co je tonometr přijat do AMS
 - zápis o vstupním testu - do systému bude uloženo datum testování, osoba, která výkon prováděla, získané hodnoty měření atd.
 - vyplnění informací o případné opravě či seřízení tonometru, které bylo na základě vstupního testu vyžádáno
 - zápis o zkoušce (protokol o měření) - přístroj byl odměřen – naměřené hodnoty budou pro zavedeny do systému
 - potvrzení ověření - systém vygeneruje certifikát v případě, že revize proběhla úspěšně a tonometr splňuje požadované limity
 - rozhodnutí o zamítnutí - systém vygeneruje zprávu v případě, že tonometr nesplňuje podmínky k udělení certifikátu
 - revize je dokončena, je zaznamenáno datum ukončení a přístroj je připraven k odeslání klientovi
- procházení a generování výpisů záznamů o klientech, tonometrech a revizích dle zvolených kritérií. Systém zobrazí např.:
 - seznamy klientů, přístrojů (tonometrů) a jednotlivých revizí
 - seznamy klientů, kteří mají přístroj aktuálně v laboratoři
 - informace o přístrojích (data ověření, počty a stáří jednotlivých revizí atd.)
 - výpisy klientů, kterým uplynula revize k zvolenému datu s možností generování informačního dopisu o prošlé lhůtě ověření atd.
- možnosti nastavení systému
 - měla by být umožněna jednoduchá správa a nastavení systému a některých jeho parametrů

- je požadováno alespoň částečné spravování tabulek pro jednotlivé etapy revizí, jejich editace a úpravy
- systém by měl umožnit jednoduchou správu zaměstnanců – nastavení přístupů do aplikace atd.

Uživatelské role a jejich potřeby:

- Technik
 - Správa zákazníků
 - Správa tonometrů
 - Správa revizí a jejich součástí
 - Zobrazení přehledů
 - Tiskové výstupy
- Ředitel
 - Přehledy o revizích v závislosti na zaměstnancích
 - Přehledy o přihlášeních do systému
- Administrátor
 - Nastavení aplikace
 - Správa parametrů měření
 - Správa typů tonometrů
 - Správa zaměstnanců
- Klient
 - vyžádání zaslání přihlašovacích údajů do systému
 - zobrazení osobních údajů a možnost jejich editace
 - přehledy o vlastních tonometrech
 - sledování stavu revizí vlastních tonometrů
 - nastavení upozornování

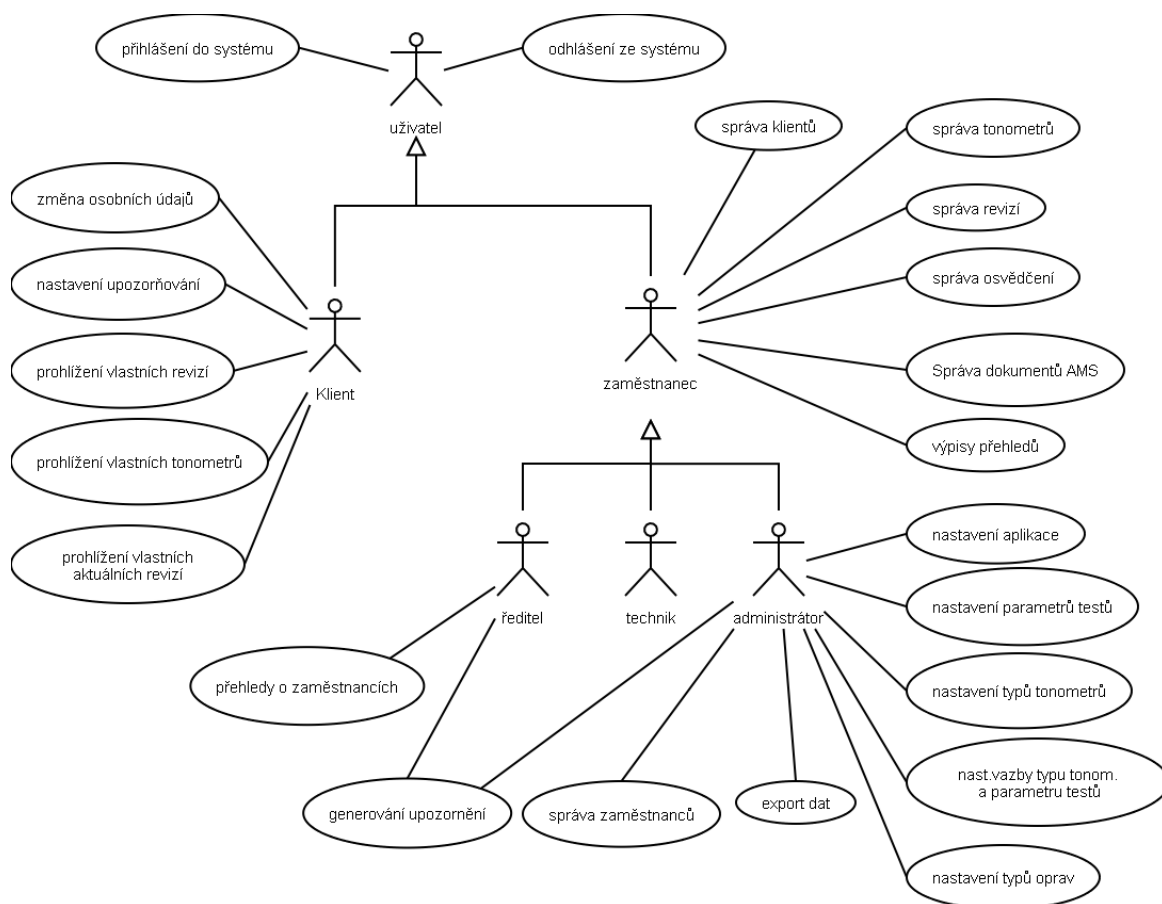
4 Specifikace požadavků

Tato část dokumentu se věnuje detailní analýze a specifikaci požadavků vyvíjeného systému. Pro modelování událostí, na které musí vyvíjený systém reagovat, je v první části uveden a popsán diagram případů použití (USE CASE). Dále následuje diskuse požadavků na uložení formulářových dat z laboratoře AMS.

4.1 Model případů použití

Podle požadavků, které byly uvedeny v předcházející kapitole, jsou nyní podrobně rozepsány jednotlivé role uživatelů navrhované aplikace. Některé případy užití bylo nutné přidat či zobecnit, aby bylo dosaženo potřebných vlastností a funkcí systému.

Pro modelování chování systému, tedy funkcí a možností, které náleží jeho účastníkům, je použit dle notace UML diagram případů užití - USE CASE. V tomto modelu případů použití jsou odlišeny dvě základní role účastníků. První je Zákazník a druhá (virtuální) Zaměstnanec, která je zobecněním role Ředitel, Technik a Administrátor. Model vypadá následovně:



Obrázek 4-2 USE CASE diagram

4.1.1 Aktér Uživatel

Aktér Uživatel představuje obecnou roli jakéhokoli uživatele systému. V případě, že se Uživatel přihlásí do systému, bude tato role zobecněním pro další aktéry (Klient, Zaměstnanec), kteří vzniknou její specializací.

4.1.1.1 Přihlášení uživatele

Předpoklady – Uživatel není přihlášen v systému.

Následné podmínky – přihlášení Uživatele do systému a přiřazení role.

Hlavní tok – Uživatel zvolí akci k přihlášení do systému. Poté vyplní své přihlašovací jméno a heslo a potvrdí volbu. Systém provede autentizaci Uživatele a pokud proběhla správně, přiřadí mu na základě zadaných údajů konkrétní roli. V případě neúspěchu je Uživatel informován a má možnost opětovného zadání přihlašovacích údajů.

4.1.1.2 Odhlášení uživatele

Předpoklady – Uživatel je přihlášen v systému.

Následné podmínky – odhlášení Uživatele se systému.

Hlavní tok – Uživatel zvolí akci k odhlášení ze systému. Systém provede proceduru odhlášení a informuje Uživatele.

4.1.2 Aktér Klient

Uživatel v roli reprezentované aktérem Klient představuje v systému roli zákazníka AMS, který je u společnosti zaregistrován a veden. Klient odesílá k revizi svůj tonometr (tonometry) do laboratoře AMS.

4.1.2.1 Změna osobních údajů

Předpoklady – žádné.

Následné podmínky – aktualizace osobních nebo kontaktních údajů o Klientovi.

Hlavní tok – Klient zvolí akci ke změně osobních údajů. Systém nabídne formulář, kde je možné editovat jednotlivé osobní i kontaktní položky včetně přístupového hesla do systému. Systém provede aktualizaci údajů a informuje Klienta.

4.1.2.2 Prohlížení vlastních revizí

Předpoklady – existuje alespoň jedna revize.

Následné podmínky – informace o dané revizi.

Hlavní tok – Klient zvolí akci k prohlížení revizí. Systém nabídne formulář, kde Klient vybere revizi, kterou si chce prohlédnout. Systém zobrazí detailní informace ke zvolené revizi.

4.1.2.3 Prohlížení vlastních tonometrů

Předpoklady – existuje alespoň jeden tonometr

Následné podmínky – informace o daném tonometru.

Hlavní tok – Klient zvolí akci k prohlížení tonometrů. Systém nabídne seznam všech jeho tonometrů, které jsou v AMS registrovány. Klient má možnost zobrazit detail každého tonometru pro konkrétnější informace.

4.1.2.4 Prohlížení aktuálních revizí

Předpoklady – existuje alespoň jedna aktuálně zpracovávaná revize.

Následné podmínky – informace o dané revizi.

Hlavní tok – Klient zvolí akci k prohlížení aktuální revize. Systém vypíše seznam a stav aktuálně zpracovávaných revizí v laboratoři AMS. Klient má možnost zobrazit detail každé revize pro konkrétnější informace.

4.1.3 Aktér Zaměstnanec

Aktér Zaměstnanec realizuje roli zaměstnance laboratoře AMS. Je modelován jako abstraktní role pro zobecnění aktérů Ředitel, Technik a Administrátor, kteří vzniknou jeho specializací. Sdružuje případy užití, které budou mít všichni tito aktéři společné. Tyto případy budou v závislosti na konkrétní roli mírně upraveny.

4.1.3.1 Správa klientů

Předpoklady – žádné.

Následné podmínky – přidání či odebrání klienta nebo změna jeho údajů.

Hlavní tok – Zaměstnanec zvolí akci k zadání nového klienta a poté vyplní jeho údaje do zobrazeného formuláře, nebo vybere ze seznamu klientů jednoho, u kterého může změnit jednotlivé údaje, anebo vybere klienta a po potvrzení volby jej vymaže.

4.1.3.2 Správa tonometrů

Předpoklady – musí existovat alespoň jeden klient.

Následné podmínky – přidání či odebrání tonometru nebo změna údajů o něm.

Hlavní tok – Zaměstnanec zvolí akci k zadání nového tonometru a poté vyplní údaje o něm do zobrazeného formuláře, nebo vybere ze seznamu tonometrů jednoho, u kterého může změnit jednotlivé údaje, anebo vybere tonometr a po potvrzení volby jej vymaže.

4.1.3.3 Správa revizí

Předpoklady – musí existovat alespoň jeden tonometr.

Následné podmínky – přidání či odebrání tonometru nebo změna údajů o něm.

Hlavní tok – Zaměstnanec zvolí akci k zadání nové revize a poté vyplní její údaje do zobrazeného formuláře, nebo vybere ze seznamu revizí jednu, které může změnit jednotlivé údaje, anebo vybere revizi a po potvrzení volby ji vymaže. V případě zadávání nové revize musí Zaměstnanec vybrat tonometr, který bude předmětem revize. Po zadání obecných informací o revizi je možné vytvořit záznam o vstupním testu. U toho musí Zaměstnanec vyplnit potřebné údaje a zhodnotit další vývoj revize. Systém umožňuje zobrazit již uložené vstupní testy a provést jejich případnou editaci. Po dokončení zadání vstupního testu může Zaměstnanec přistoupit ke kroku zadávání naměřených hodnot do záznamu o zkoušce. Zadané hodnoty může rovněž později editovat po vybrání konkrétního záznamu o zkoušce. Kdykoliv během procesu zadávání hodnot může Zaměstnanec vložit k danému kroku poznámku a případně změnit typ opravy či seřízení tonometru, vyžádá-li si to měření. Po dokončení vyplnění formulářů testů a v závislosti na výsledku měření systém vygeneruje informace o ověření, jehož detaily a položky lze rovněž editovat.

4.1.3.4 Správa osvědčení

Předpoklady – musí existovat ověření anebo dokončený záznam o zkoušce.

Následné podmínky – přidání ověření, zobrazení informací o něm, anebo vygenerování ověření.

Hlavní tok – Zaměstnanec zvolí akci k zadání nového ověření a vybere, pro kterou revizi má být ověření vydáno a poté vyplní dodatečné údaje, nebo zvolí akci k prohlížení ověření a systém zobrazí jejich seznam s možným detailním náhledem a příp. editací vybrané položky, anebo Zaměstnanec zvolí akci pro vygenerování ověření, kdy systém provede zobrazení vybraného ověření s možností výstupu ve formě určené k tisku anebo k uložení do souboru.

4.1.3.5 Výpisy přehledů

Předpoklady – žádné.

Následné podmínky – výpis určených přehledů a statistik o zákaznících, tonometrech a revizích.

Hlavní tok – Zaměstnanec vybere akci k zobrazení přehledů a vybere, které přehledy chce zobrazit. Systém zobrazí formulář, ve kterém požaduje zadání parametrů na základě kterých se vypíší přehledy revizí, tonometrů atd.

4.1.3.6 Správa dokumentů AMS

Předpoklady – na serveru jsou uloženy soubory dokumentů.

Následné podmínky – získání dokumentů laboratoře.

Hlavní tok – Zaměstnanec zvolí akci k získání dokumentů laboratoře. Systém zobrazí obsah adresáře na serveru, kde jsou uloženy dokumenty určené k tisku (formuláře laboratoře apod.) a nabídne existující soubory ke stažení (pozn. upload a mazání jsou povolena rolím Administrátor a Ředitel).

4.1.4 Aktér Technik

Role, kterou reprezentuje aktér Technik, odpovídá roli technického zaměstnance laboratoře AMS. Má v systému všechny vlastnosti, možnosti a funkce svého předka – aktéra Zaměstnanec.

Pod touto rolí budou do systému přistupovat všichni zaměstnanci kromě Administrátora a Ředitele, kteří mají více možností. Technikům bude umožněno zadávat a editovat nové zákazníky, tonometry, revize a jejich součásti. Mazání a editace stávajících dat, tedy těch, která nejsou nově vytvořena, anebo jsou již ve stavu dokončení, umožněna této roli není. Je to z bezpečnostních důvodů - jak proti nechtěným změnám hodnot, tak proti úmyslným zásahům do uložených dat.

4.1.5 Aktér Administrátor

Aktér Administrátor realizuje zaměstnanec, který je zodpovědný za údržbu systému a má pověření k jeho nastavení a dalším funkcím. Tato role je odvozena od aktéra Zaměstnanec - má tedy všechny jeho možnosti a některé další přidává.

Na rozdíl od Techniků má Administrátor oprávnění k tomu, aby mohl editovat a mazat jakékoli položky revizí, údaje o tonometrech a klientech.

4.1.5.1 Správa zaměstnanců

Předpoklady – žádné.

Následné podmínky – přidání či odebrání zaměstnance nebo změna jeho údajů.

Hlavní tok – Administrátor zvolí akci k zadání nového zaměstnance a poté vyplní jeho údaje do zobrazeného formuláře, nebo vybere ze seznamu zaměstnanců jednoho, u kterého může změnit jednotlivé údaje, anebo vybere zaměstnance a po potvrzení volby jej vymaže.

4.1.5.2 Správa dokumentů AMS

Předpoklady – žádné.

Následné podmínky – uložení a smazání dokumentů na server.

Hlavní tok – Administrátor vybere akci k uložení dokumentů na server nebo jejich smazání. Systém zobrazí formulář pro upload souboru. Po zadání souboru a potvrzení formuláře se soubor uloží do zadaného adresáře na serveru. V případě, že chce Administrátor dokument ze serveru smazat, vybere volbu ke smazání souboru u jeho názvu ve výpisů. Poté systém odstraní vybraný dokument z daného adresáře na serveru.

4.1.5.3 Nastavení parametrů testů

Předpoklady – žádné.

Následné podmínky – přidání či odebrání parametru testů nebo změna jeho údajů.

Hlavní tok – Administrátor zvolí akci k zadání nového parametru testů a poté vyplní jeho údaje do zobrazeného formuláře, nebo vybere ze seznamu parametrů jednoho, u kterého může změnit jednotlivé údaje, anebo vybere parametr testů a po potvrzení volby jej vymaže.

4.1.5.4 Nastavení vazeb parametrů testů a tonometrů

Předpoklady – existuje alespoň jeden typ tonometru a test parametru.

Následné podmínky – přidání či odebrání vazby parametrů testů a typů tonometrů nebo změna jejich údajů.

Hlavní tok – Administrátor zvolí akci k zadání nové vazby parametrů testů a typu tonometru a poté vyplní její údaje do zobrazeného formuláře, nebo vybere ze seznamu vazeb jednu, u které může změnit jednotlivé údaje, anebo vybere vazbu a po potvrzení volby ji vymaže.

4.1.5.5 Nastavení typů tonometrů

Předpoklady – existuje alespoň jeden tonometr.

Následné podmínky – přidání či odebrání typu tonometru nebo změna jeho údajů.

Hlavní tok – Administrátor zvolí akci k zadání nového typu tonometru a poté vyplní jeho údaje do zobrazeného formuláře, nebo vybere ze seznamu typů tonometrů jeden, u kterého může změnit jednotlivé údaje, anebo vybere typ tonometru a po potvrzení volby jej vymaže.

4.1.5.6 Nastavení typů oprav

Předpoklady – žádné.

Následné podmínky – přidání či odebrání typu opravy nebo změna jeho údajů.

Hlavní tok – Administrátor zvolí akci k zadání nové opravy a poté vyplní její údaje do zobrazeného formuláře, nebo vybere ze seznamu typů oprav jednu, u které může změnit jednotlivé údaje, anebo vybere typ opravy a po potvrzení volby ji vymaže.

4.1.5.7 Nastavení aplikace

Předpoklady – žádné.

Následné podmínky – nastavení hodnoty časového limitu pro automatické odhlášení uživatele ze systému.

Hlavní tok – Administrátor zvolí akci k nastavení časového údaje pro automatické odhlášení ze systému pro nečinnost a poté nastaví požadovanou hodnotu.

4.1.5.8 Export dat

Předpoklady – žádné.

Následné podmínky – provedení exportu dat určených databázových tabulek do souboru ve vybraném formátu - XML nebo CSV².

Hlavní tok – Administrátor zvolí akci export dat a poté vyplní potřebné údaje (název souboru, formát výstupu) do zobrazeného formuláře a zvolí tabulky, jejichž data chce exportovat. Systém poté vytvoří soubor (XML nebo CSV), do kterého uloží vybrané údaje z databáze.

4.1.5.9 Generování upozornění

Předpoklady – žádné

Následné podmínky – vygenerování dopisů nebo zaslání emailů pro klienty s propadlou revizí.

Hlavní tok – Administrátor zvolí akci pro zobrazení tonometrů, kterým k danému termínu s nastavenou tolerancí propadnou revize. Systém nabídne vygenerování automatických dopisů s informacemi o tonometru, revizi a klientovi a zobrazí tento text v podobě vhodné k tisku. Další možností bude poslat klientům email, který je na skutečnost o propadlých revizích upozorní..

4.1.6 Aktér Ředitel

Uživatel v roli definované aktérem Ředitel představuje zaměstnance, kterému je umožněno sledovat informace o ostatních zaměstnancích. Tato role je odvozena od aktéra Zaměstnanec - má tedy všechny jeho možnosti a některé další přidává.

Ředitel je oprávněn, stejně jako Administrátor, k posílání emailů a generování dopisů o upozornění na propadlé revize.

4.1.6.1 Přehledy o zaměstnancích

Předpoklady – žádné.

Následné podmínky – výpis přehledů o zaměstnancích a jejich práci na revizích.

Hlavní tok – Ředitel zvolí akci prohlížení přehledů o zaměstnancích a vybere parametry, jako je např. časové období, které chce sledovat. Systém zobrazí seznam zaměstnanců a detailní informace o jejich účastech na jednotlivých revizích za určený časový úsek.

4.2 Zpracovávaná data laboratoře

Nyní je stručně nastíněn postup a problematika spojená se zpracováním dat laboratoře AMS. Je zde zmíněno hlavní rozčlenění typů tonometrů a dokumentů, které bylo nutné při specifikaci požadavků ke zpracování dat analyzovat.

² CSV - Comma-separated values (hodnoty oddělené čárkami) je jednoduchý souborový formát určený pro výměnu tabulkových dat

4.2.1 Typy tonometrů

Každý tonometr, který se v AMS měří a testuje, spadá do určité skupiny a má prvky charakteristické pro tuto skupinu.

Společnost HNS provádí revize dvou základních typů tonometrů, které se od sebe značně odlišují. Jedná se tonometry typu: „aplanační - goldmann“ a „schiötz“. Ke každému z těchto typů přísluší jiné vlastnosti, měřicí hodnoty a jejich rozsahy a s tím související dokumenty a formuláře. V rámci každého typu se odlišují tonometry podle modelů a výrobců. Každý model má svá specifika, která se mírně odlišují od ostatních, a je proto nutné brát na tento fakt zřetel při provádění měření, generování formulářů a vystavování výsledků ověření.

4.2.2 Typy formulářů

V testovací laboratoři HNS mechanix s.r.o. se v současné době používá několik typů formulářů, které přísluší různým etapám revize a testování a jsou závislé na typu tonometru, který se testuje.

Jedná se především o formuláře pro „vstupní test“ (ukázka v příloze č 1 obr. 1-1) a formuláře pro „záznam o zkoušce“ (ukázka v příloze č. 1 obr. 1-2). Tyto dva protokoly se vyplňují při daných etapách měření tonometru. Dalšími dokumenty, které se používají v AMS, jsou výsledky dané revize. V případě, že tonometr splňuje dané vlastnosti a limity naměřených hodnot nepřekročí v danou toleranci, je vydáno Osvědčení o ověření (ukázka v příloze č.1 obr. 1-3). V opačném případě je vydáno Rozhodnutí o zamítnutí.

4.2.3 Požadavky na zpracování

Při navrhování vyvíjeného informačního systému je třeba popsané typy protokolů a dokumentů respektovat. Téměř všechny prvky, limity a tolerance testů, resp. popsaných formulářů, jsou pevně dány vyhláškami a nařízeními ČMI (Český metrologický institut) a ÚNMZ (Úřad pro technickou normalizaci, metrologii a státní zkušebnictví). Přesto je nezbytné umožnit alespoň částečnou modifikaci těchto elementů.

Dynamické položky jednotlivých formulářů a dokumentů budou proto uloženy v databázi systému, čímž bude umožněna potřebná flexibilita a variabilita při jejich generování a následném zpracování. Rovněž jejich modifikace bude snadná a přehledná. Statické (tedy neměnné) prvky dokumentů budou uloženy v šablonách jednotlivých dokumentů (formulářů), což také umožní jejich případnou pozdější editaci.

5 Návrh aplikace

Tato část dokumentu je věnována detailnímu návrhu informačního systému. Nejprve je stručně popsána architektura aplikace a jejích součástí spolu s použitými technologiemi. Dále je navržen model dat pomocí diagramu ERD, ze kterého se vychází při tvorbě databázových tabulek. Poté je popsán návrh grafického uživatelského rozhraní.

5.1 Architektura aplikace

Aplikace bude postavena na třívrstvé architektuře klient – server v prostředí sítě Internet. Bude využit webový server Apache, skriptovací jazyk PHP a databázový systém MySQL, jak již bylo zmíněno v kapitole 2.1.

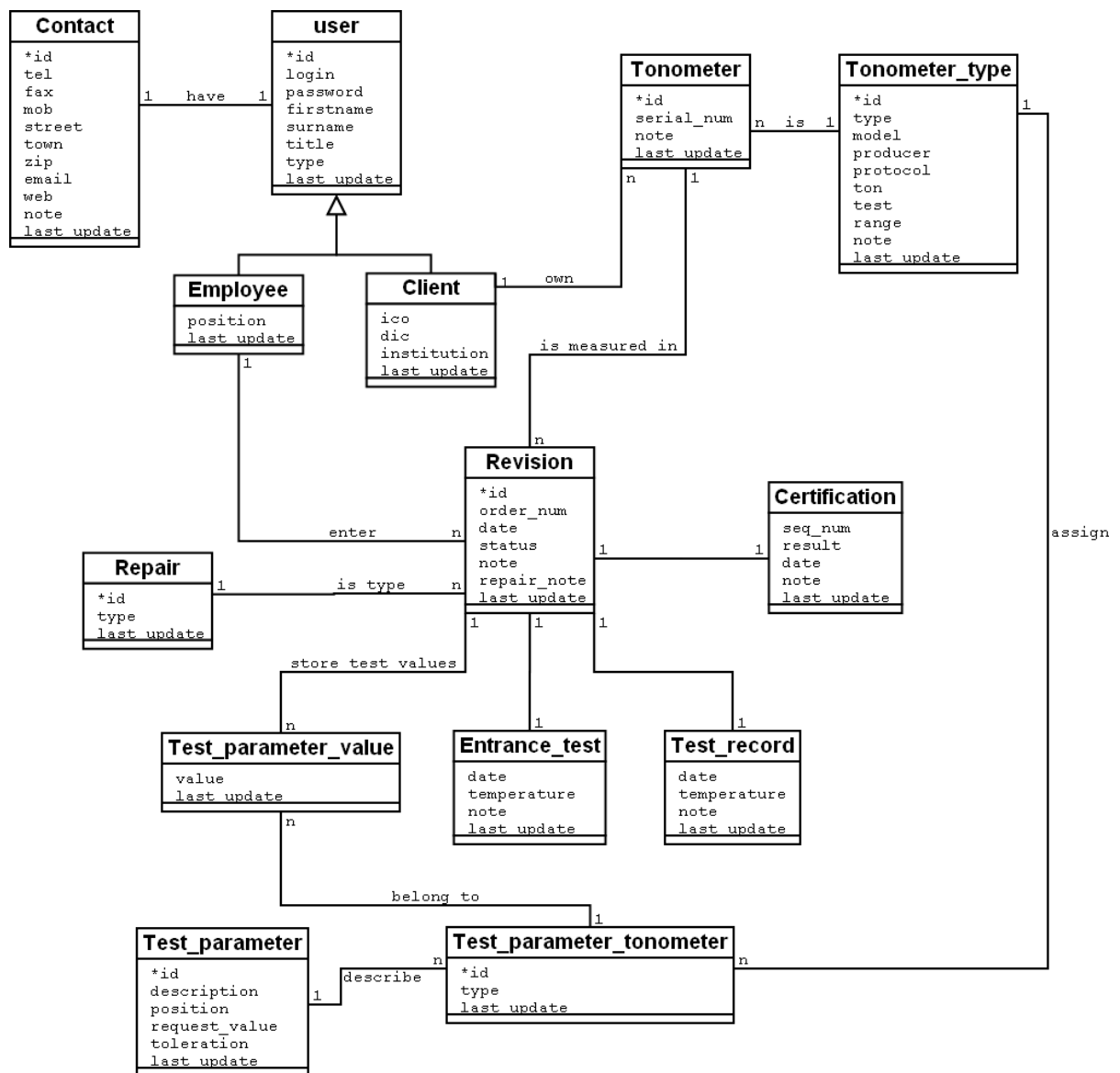
Aplikace bude řešena s využitím objektově orientovaných technik v prostředí PHP. Použití tohoto skriptovacího jazyka v objektovém programování sice přináší oproti jiným programovacím jazykům mnohá omezení a jistý stupeň benevolence, nicméně nesporné výhody a některé užitečné vlastnosti tohoto přístupu jeho nasazení i v objektovém programování ospravedlňují. Oddělení aplikační logiky od prezentační bude částečně realizováno použitím vlastního jednoduchého šablonovacího mechanismu (popsán v kap. 6.4.2). Ačkoliv nejde o komplexní nástroj, zachovává jednoduchost a přehlednost skriptů aplikace s ohledem na její pozdější editaci a rozšiřování.

Aplikace bude rozdělena do několika modulů, které budou každý zodpovídat za určitou sekci funkcí. Tyto moduly spolu budou do jisté míry provázány.

Při implementaci systému bude využito několik volně dostupných rozšiřujících knihoven PHP. V první řadě to bude rozšíření pro práci s databází ADODB (popsáno v kap.6.4.1.1). Dále bude ke zpracování formulářových hodnot použita knihovna PEAR QuickForm (popsáno v kap. 6.4.4.1) a pro realizaci jednotné navigace knihovna HTML Menu (popsáno v kap. 6.4.3.1). Tato rovněž pochází z knihovny PEAR. Pro exportování databázových hodnot do souborů formátu XML a CSV bude využito volně použitelných knihoven PEAR HTML_Serialize (popsáno v kap. 6.4.6.1) a třídy z ADODB (popsáno v kap. 6.4.6.2). Validace vstupních dat, automatické doplňování polí a změny v zobrazení prvků formulářů budou implementovány pomocí jazyku JavaScript.

5.2 Model ERD

Statická struktura dat celé aplikace je modelována pomocí ERD diagramu, kde je každý prvek systému znázorněn pomocí entity se svými atributy. Tyto datové prvky jsou mezi sebou logicky provázány pomocí znázorněných vztahů s určitou kardinalitou (násobností).



Obrázek 5-3 ERD diagram

5.2.1 Popis entit diagramu

5.2.1.1 User (Uživatel)

Entita User obsahuje základní údaje o uživateli systému. Jsou to především přihlašovací údaje: jméno (login) a heslo, dále osobní informace: jméno, příjmení, titul. Je zde zaznamenáno, jakou roli daný uživatel v systému hraje zda jde o klienta nebo zaměstnance. V případě, že jde o zaměstnance je nutné dále rozlišit role technik, administrátor a ředitel.

5.2.1.2 Client (Klient)

Tato entita je specializací entity User a uchovává informace o klientovi společnosti HNS mechanix s.r.o. Jsou to údaje o firmě, resp. instituci, jako jsou IČO, DIČ a název zařízení, které klient zastává.

5.2.1.3 Employee (Zaměstnanec)

Další specializací entity User je Employee. Uchovávat bude informace o zaměstnanci jako je např. popis jeho pozice ve firmě HNS mechanix s.r.o..

5.2.1.4 Contact (Kontakt)

Každý Uživatel systému může mít uvedené kontaktní informace (u Klientů je to nutné). Jsou to především telefonní spojení (pevná linka, mobil, fax), adresa, email a příp. i webová adresa.

5.2.1.5 Tonometer (Tonometr)

O každém Tonometru je nutné uchovat informace o typu, modelu, jeho výrobní číslo a výrobce a identifikaci vlastníka přístroje.

5.2.1.6 Tonometer type (Typ tonometru)

Tato entita nese informace o typu tonometru. V první řadě je jeho typ, model a výrobce. Dále je nutné pro pozdější generování formulářů uchovávat informace pro každý typ tonometru: název testu, testovací protokol, název tonometru a rozsah měření.

5.2.1.7 Revision (Revize)

Entita Revize modeluje celý proces měření i servisu tonometru. Uchovává vazby na jednotlivé zápisy z laboratoře, které jsou v procesu ověření prováděny. Jsou to Vstupní test, záznam o Opravě, Zápis o zkoušce a Ověření.

5.2.1.8 Entrance test (Vstupní test)

Prvním krokem po přijetí Tonometru do AMS je vyplnění Vstupního testu. Je nutné uchovat informace o Tonometru, datum přijetí, popis a výsledky testů z laboratoře, údaj o tom, zda bude muset být Tonometr opravován atd.

5.2.1.9 Repair (Typ opravy)

V případě, že je nutné po vstupním testu provést opravu Tonometru, je nutné uchovat servisní údaje o rozsahu a míře opravy, konkrétní popis, informace o technikovi, který úkon prováděl apod.

5.2.1.10 Test record (Zápis o zkoušce)

Během měření parametrů Tonometru se průběžně zaznamenávají získané hodnoty, informace o tom, kdo Tonometr testoval a další.

5.2.1.11 Certification (Ověření)

Po úspěšném přeměření parametrů Tonometru je vydáno potvrzení o Ověření, které má platnost jeden rok. Údaje, které je nutné uchovat jsou např. vybrané hodnoty, které byly v laboratoři naměřeny, informace o tom, kdo a kdy měření prováděl aj.

5.2.1.12 Test parameter value (Hodnota parametru testu)

Tato entita obsahuje hodnoty, které byly naměřeny v jednotlivých fázích revize. Nese identifikaci o položce formuláře, které náleží.

5.2.1.13 Test parameter (Parametr testu)

Entita Test parameter nese údaje o dynamických polích formulářů. Jde položky popis, případná pozice, požadovaná hodnota a tolerance.

5.2.1.14 Test parameter tonometer (Parametr test tonometru)

Tato entita obsahuje informaci o vztahu mezi entitami Tonometer_type a Test_parameter. Tím určuje, jakou hraje daná vazba u formuláře roli.

5.2.2 Popis vazeb diagramu

Uvedený diagram ERD obsahuje několik typů vazeb mezi jednotlivými entitami. Jejich význam, by měl být patrný z diagramu. V případě uvedených vazeb 1:1 se může nabízet otázka, proč nejsou tyto datové entity spojeny do jedné. Důvod tohoto rozdělení je ten, že tyto entity (a tedy pozdější tabulky databáze) odliší logické celky dat a zjednoduší tak jejich případné rozšiřování. Např. entita Contact byla modelována vazbou 1:1 k entitě User z toho důvodu, aby nenastal problém tehdy, bude-li vyžadováno zaznamenání více kontaktů k jednomu uživateli. A to i přesto, že tento případ nebyl součástí požadavků aplikace apod.

5.3 Návrh uživatelského rozhraní

5.3.1 Layout

Vyvíjená aplikace je webově orientována, proto je nutné zohlednit tento fakt při návrhu uživatelského rozhraní. Dále je nutné přihlídnout k tomu, že se jedná o informační systém, který má jak intranetovou, tak veřejně přístupnou část.

Pro potřeby zaměstnanců AMS je vhodné řešit uživatelské rozhraní v jednodušším a účelném stylu a nabídnout co nejlepší stupeň přehlednosti a ergonomie. Zároveň by však měl layout i design splňovat osvědčené standardy struktury a rozložení pro ostatní uživatele. V neposlední řadě by mělo uživatelské rozhraní působit moderně a reprezentativně.

Způsob rozvržení komponent na webových stránkách systému bude vycházet z osvědčeného dvousloupcového stylu. V horní části bude informační lišta, která ponese informace o přihlášeném uživateli. Pod touto částí budou umístěny navigační prvky – záložky, pomocí kterých se uživatel jednoduše přepne do jednotlivých sekcí informačního systému. Hlavní plocha stránky bude rozdělena na dva sloupce. Levý sloupec bude obsahovat dynamické menu aplikace. Druhá větší část plochy bude pro hlavní výstup stránky. Ve spodní části bude zobrazena patička s kontaktními informacemi.

5.3.2 Navigace

Aby bylo možno aplikaci efektivně využívat, je nutné, aby měla jasně definovanou logickou strukturu a ta byla následně přehledně prezentována. S tím pak souvisí malé nároky na uživatele při pochopení ovládání informačního systému a práci s ním. Hlavním pilířem pro zachování přehlednosti v systému je vhodně zvolená metoda navigace.

Jelikož aplikace obsahuje několik hlavních částí, je vhodné ji rozčlenit do samostatných sekcí. K tomuto účelu poslouží systém záložek, které budou zobrazeny v horní části stránky. Aby uživatel neztrácel přehled, v jaké sekci se právě nachází, bude aktivní záložka viditelně odlišena od ostatních.

Z důvodu vícevrstvé struktury systému je nutné zavést zároveň se zmíněnými záložkami detailnější rozčlenění jednotlivých sekcí. Toho bude dosaženo vytvořením podrobného menu, které se bude dynamicky měnit podle aktuální sekce systému. Toto menu umožní hierarchické zanořování, přičemž zůstanou zobrazeny nadřazené sekce, aby bylo patrné, kde se uživatel právě nachází.

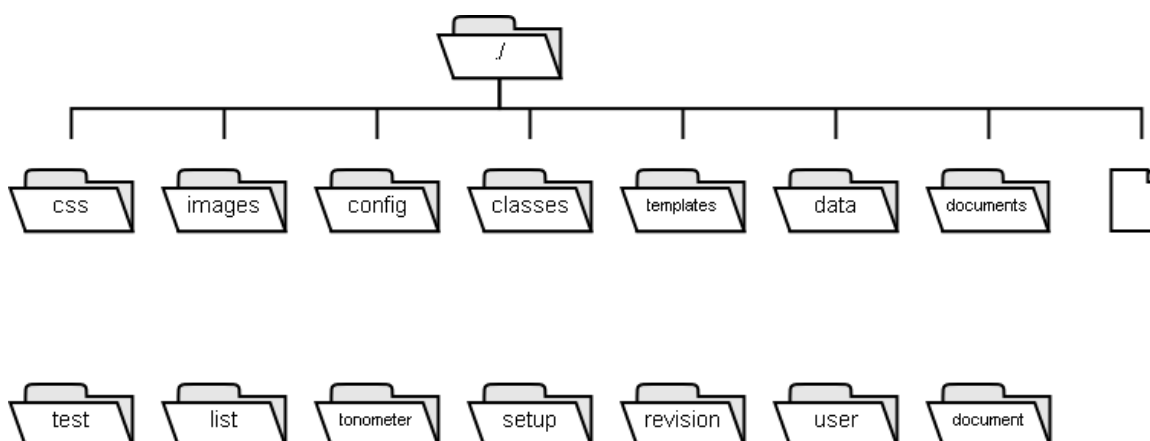
Zmíněné prvky navigace budou na všech stránkách aplikace víceméně stále. Je to z toho důvodu, aby uživatelé měli kdykoli přehled o tom, v jaké sekci systému se nacházejí a jaký má rozsah, a také proto, aby se mohli pohodlně dostat na jiné místo.

6 Implementace aplikace

Tato část dokumentu popisuje etapu implementace. V první části je znázorněna struktura aplikace a popsány její jednotlivé části. Dále jsou zmíněny konkrétní techniky a rozšíření, které byly při různých fázích implementace využívány. Následuje ukázka uživatelského rozhraní a jeho prvků. Na závěr je popsána a diskutována bezpečnost celého systému s návrhem možných vylepšení.

6.1 Struktura aplikace

Pro vyžadované možnosti jednoduché správy systému a z důvodů jeho pozdějšího rozšiřování byla aplikace rozdělena do několika modulů. Každá takto logicky oddělená část má za úkol spravovat jiný úsek systému a jsou na sobě do jisté míry nezávislé. Adresářová struktura a logické členění modulů jsou uvedeny na obr. 6-1. V horní části obrázku jsou znázorněny adresáře, spojené s funkcemi a nastaveními aplikací jako takové, ve spodní části jsou uvedeny jednotlivé funkční moduly systému.



Obrázek 6-4 Struktura aplikace

6.2 Moduly systému

Obecně každý modul obsahuje několik PHP skriptů, které jsou zodpovědné za vygenerování správných komponent výstupní stránky. Je v nich tedy obsažena aplikační logika. Dále každému modulu přísluší implementovaná třída (třídy), které mají za úkol poskytovat vrstvu pro přístup k databázovým datům. Veškeré operace nad databází jsou tedy prováděny v těchto třídách (ve svém názvu obsahují slovo manager).

Metody typu „getter“ a „setter“³ v seznamu uvedené nejsou, neboť jejich význam a funkce jsou pevně dané a je zbytečné je zde uvádět.

³ Getter, resp. setter - metody, které umožňují získání, resp. nastavení, hodnoty atributu objektu.

6.2.1 Uživatelé

Tento modul má za úkol spravovat informace o uživateli systému. Obsahuje tedy skripty ke správě klientů a zaměstnanců a jejich zaznamenaných údajích.

Nyní jsou detailně popsány jednotlivé třídy modulu Uživatelé. U každé třídy jsou uvedeny její atributy a operace.

6.2.1.1 User Manager

Třída *UserManager* zajišťuje operace nad uživateli systému a jejich údaji. Třída je navržena pomocí vzoru singleton⁴. Obsahuje operace pro vytváření, editace a mazání tabulek v databázi, které jsou spjaty s uživateli a dále pro obsluhu procesů používaných při autentizaci do systému.

atributy:

- *instance*⁵ – odkaz na instanci třídy v systému

operace:

- *getInstance*⁶ – vrací instanci třídy, pomocí ní se ke třídě přistupuje. Tím je zajištěn jedinečný výskyt této třídy v celé aplikaci.
- *createUser* – vytvoří instanci objektu User, resp. Klient anebo Employee, v závislosti na procesu autorizace uživatele do systému.
- *loginUser* – provede proceduru přihlášení uživatele do systému. Nejprve provede autentizaci uživatele porovnáním zadaných přihlašovacích údajů s hodnotami v databázi. Poté nastaví časový limit doby nečinnosti uživatele a uloží tento údaj spolu s vytvořeným objektem User (viz. operace *createUser* popsána výše) do session. Nakonec přesměruje uživatele na vstupní menu aplikace.
- *logoutUser* – vykoná proces odhlášení uživatele ze systému, smaže uložené hodnoty session
- *generateLogin* – upraví zadaný řetězec na tvar přihlašovacího jména (loginu) – zkrátí, resp. dolní, jej na požadovanou délku a přidá prefix. Pokud už dané jméno v systému existuje, přidá jako postfix řetězce pořadové číslo výskytu tohoto jména.
- *generatePassword* – vygeneruje pseudonáhodnou posloupnost znaků zvolené délky, která vznikne permutací symbolů ze zadané množiny.
- *createNewUser* – ze zadaných dat vytvoří nový záznam o uživateli v databázi.
- *editUser* – provede aktualizaci údajů uživatele v databázi na základě zadaných hodnot
- *editEmployee* - aktualizuje údaje zaměstnance v databázi na základě zadaných hodnot

⁴ Vzor singleton je jeden z nejznámějších návrhových vzorů OOP. Zajišťuje jedinečnost výskytu objektu v aplikaci. Z celé aplikace je možné k této jediné instanci přistupovat, ale nelze vytvořit její kopii.

⁵ Atribut instance přímo souvisí s návrhovým vzorem singleton a již v rámci zjednodušení nebude o dalších třídách uváděn.

⁶ Metoda *getInstance* rovněž souvisí se vzorem singleton a již nebude u dalších tříd popisována.

- *editClient* - modifikuje údaje o klientovi v databázi na základě zadaných hodnot
- *editContact* – edituje informace o kontaktu uživatele v databázi na základě zadaných hodnot
- *deleteUser* – vymaže údaj o uživateli z databáze
- *deleteEmployee* – smaže informace o zaměstnanci z databáze
- *deleteClient* - smaže informace o zaměstnanci z databáze
- *deleteContact* – odstraní údaj o kontaktu uživatele z databáze
- *getUserInfo* – vrátí objekt uživatele z databáze
- *getEmployeeInfo* – z databáze získá objekt zaměstnance
- *getClientInfo* – načte informace o klientovi a vrátí objekt
- *getContactInfo* – získá údaje o kontaktu uživatele a vrátí objekt
- *listUsers* – vrátí seznam uživatelů jako pole objektů

6.2.1.2 User

Třída *User* obsahuje atributy uživatele systému a poskytuje operace pro naplnění těchto hodnot z databáze. Tato je vytvořena po úspěšné autentizaci uživatele do systému a poté uložena do session, aby k ní bylo možné později přistupovat.

atributy:

- *Id* – jedinečné identifikační číslo uživatele
- *Login* – přihlašovací jméno uživatele
- *Firstname* – křestní jméno uživatele
- *Surname* – příjmení uživatele
- *Title* – titul uživatele
- *Type* – typ uživatel v systému (přiřazení role Klient, Zaměstnanec apod.)

operace:

- *setAttributes* – tato operace naplní uvedené atributy odpovídajícími hodnotami z databáze

6.2.1.3 Client

Třída *Client* je odvozena ze třídy *User*. Poskytuje atributy klienta a operace pro manipulaci s nimi.

atributy:

- *ico* – identifikační číslo osoby
- *dic* – daňové identifikační číslo
- *institution* – název instituce resp. společnosti klienta

operace:

- *setAttributes* – tato operace naplní uvedené atributy příslušnými hodnotami z databáze

6.2.1.4 Employee

Třída *Employee* je odvozena ze třídy *User*. Poskytuje atributy zaměstnance a operace pro manipulaci s nimi.

atributy:

- *Position* – popis pozice zaměstnance v laboratoři AMS

operace:

- *setAttributes* – tato operace naplní uvedené atributy odpovídajícími hodnotami z databáze

6.2.2 Tonometry

Modul Tonometry operuje s veškerými daty, které jsou spjaty s tonometry a jejich správou. Sdružuje skripty aplikace, které jsou k tomuto účelu vytvořené.

Nyní je popsána základní třída, která poskytuje hlavní metody, které tyto skripty využívají.

6.2.2.1 Tonometer Manager

Třída *TonometerManager* poskytuje operace nad tonometry. Třída je navržena pomocí vzoru singleton. Obsahuje operace vytváření, editace a mazání tabulek tonometrů a jejich typů v databázi.

operace:

- *createNewTonometer* - ze zadaných dat vytvoří v databázi nový záznam o tonometru
- *createNewTonometerType* – vytvoří nový údaj o typu tonometru
- *deleteTonometer* – smaže údaje o zadaném tonometru z databáze
- *deleteTonometerType* – odstraní z databáze informace o daném typu tonometru
- *editTonometer* – změní v databázi zadané údaje o tonometru
- *editTonometerType* – modifikuje informace o typu tonometru uložené v databázi
- *getTonometerInfo* – získá z databáze údaje o tonometru a vrátí objekt
- *getTonometerTypeInfo* – načte z databáze informace o typu tonometru
- *listTonometers* – vrátí seznam objektů tonometrů
- *listTonometerTypes* – získá pole objektů uložených typů tonometrů

6.2.3 Revize

Modul Revize spravuje informace a veškeré operace s revizemi. Obsahuje PHP skripty aplikace, které jsou k tomuto účelu vytvořené.

Nyní je popsána základní třída, která poskytuje hlavní metody, které využívají ostatní skripty pro práci s revizemi.

6.2.3.1 Revision Manager

Třída *RevisionManager* poskytuje metody nad revizemi a všemi jejími součástmi. Třída je navržena pomocí vzoru singleton. Obsahuje operace vytváření, editace a mazání tabulek v databázi, které jsou spjaty s revizemi.

operace:

- *createNewRevision* - ze zadaných dat vytvoří v databázi nový záznam o revizi.
- *createNewEntranceTest* – za zadaných informací vytvoří nový záznam v databázi o vstupním testu dané revize
- *createNewTestRecord* – vytvoří nový záznam o zkoušce revize v databázi
- *createNewCertification* – slouží ke tvorbě nového výsledku ověření
- *deleteRevision* – smaže zadanou revizi z databáze
- *deleteEntranceTest* – vymaže zvolený vstupní test z databáze
- *deleteTestRecord* – odstraní z databáze záznam o zkoušce
- *deleteCertification* – smaže v databázi údaje o ověření revize
- *editRevision* – upraví zadané údaje revize v databázi
- *editEntranceTest* – upravuje položky vstupního testu revize v databázi
- *editTestRecord* – modifikuje záznam o zkoušce uložený v databázi
- *editCertification* – slouží k editaci daných položek ověření
- *getRevisionInfo* – získá informace o dané revizi
- *getEntranceTestInfo* – vrátí údaje o vybraném vstupním testu
- *getTestRecordInfo* – získá položky záznamu o zkoušce
- *getCertificationInfo* – poskytne informace o vybraném ověření
- *listRevisions* – vrátí seznam revizí
- *listEntranceTests* – získá seznam vstupních testů
- *listTestRecords* – vrátí výpis záznamů o zkoušce
- *listCertifications* – z databáze získá seznam ověření

6.2.4 Nastavení

Třída *SetupManager* poskytuje operace k nastavení některých položek aplikace a jejich jednotlivých součástí. Třída je navržena pomocí vzoru singleton. Obsahuje především operace vytváření, editace a mazání tabulek v databázi, které jsou spjaty s nastavením systému.

operace:

- *createNewRepair* – vytvoří v databázi novou položku typu opravy
- *deleteRepair* – z databáze odstraní vybraný typ opravy
- *editRepair* – edituje v databázi informaci o opravě

- *getRepairInfo* – načte z databáze informace o typu opravy
- *listRepairs* – poskytne výpis typů oprav z databáze

6.2.5 Testy

Třída *TestManager* poskytuje operace nad prvky všech formulářů testů (vstupní test, zápis o zkoušce, ověření). Třída je navržena pomocí vzoru singleton. Obsahuje operace vytváření, editace a mazání tabulek v databázi, které jsou přímo spjaty s testy.

operace:

- *createNewTestParameter* – vytvoří v databázi nový záznam o parametru testů
- *createNewTestParameterTonometer* – uloží do databáze novou „vazbu“ (popsáno v kap. 6.6) parametru testů a typu tonometru
- *deleteTestParameter* – odstraní položku parametr testů
- *deleteTestParameterTonometer* – smaže z databáze vazbu parametr testů a typ tonometru
- *editTestParameter* – modifikuje na základě zadaných dat v databázi položku parametr testů
- *editTestParameterTonometer* – změní v databázi údaj o vazbě parametr testů a typ tonometru
- *getTestParameterInfo* – vrátí informace o zvoleném parametru testu
- *getTestParameterTonometer* – z databáze získá údaje o vazbě parametr testů a typ tonometru
- *listTestParameters* – vrátí z databáze seznam parametrů testů
- *listTestParameterTonometer* – načte seznam vazeb parametrů testů a typů tonometrů

6.2.6 Přehledy

Třída *ListManager* poskytuje operace nad přehledy a statistikami. Třída je navržena pomocí vzoru singleton. Obsahuje operace na tvorbu složitějších dotazů na databázi, které jsou podstatou zobrazovaných statistik a přehledů.

operace:

- *listLastRevisionsInt* – vrátí seznam posledních revizí s určenými vlastnostmi v zadaném časovém intervalu
- *listStatsInt* – z databáze získá statistické údaje o revizích na základně určených parametrů
- *listLastRevisions* – vrátí seznam revizí, kterým končí revize k zadanému období
- *listQuery* – zavolá obecný dotaz s danými parametry (je využito pro výše uved. metody)
- *listEmployees* – vrátí statistiky o zaměstnancích a jejich účasti na revizích

6.2.7 Dokumenty

Tento modul spravuje dokumenty laboratoře AMS. Zaměstnancům umožňuje ukládat a otevírat dokumenty, které jsou uloženy v příslušném adresáři serveru. Administrátorovi a Řediteli umožní příslušný formulář nahrát libovolný soubor na server a v případě potřeby uložené soubory vymazat.

6.3 Aplikační logika

Nyní je uvedeno několik tříd, které sdružují funkce nad aplikací jako takovou, nemanipulují tedy s údaji v databázi apod. Metody těchto tříd jsou využívány jednotlivými skripty a tvoří aplikační vrstvu systému.

6.3.1.1 Page

Třída *Page* sdružuje operace pro nastavování a editaci vlastností webové stránky aplikace. Obsahuje metody, které slouží pro kontrolu oprávnění k prohlížení, k zobrazení stránky a k nastavení jejích jednotlivých součástí.

atributy:

- *Template* - jméno souboru, který slouží jako šablona při vykreslování stránky
- *Html* - vygenerovaný HTML kód stránky
- *Parameters* – pole všech dynamických položky stránky
- *Right* – číselná hodnota určující míru zabezpečení stránky

operace:

- *createPage* – vytvoří stránku na základě zadaných parametrů *Parameters* a šablony *Template*. Šablona stránky se načte do proměnné *Html* a speciálně označené úseky jsou nahrazeny příslušnými hodnotami z pole *Parameters*.
- *displayHMenu* – zobrazí v závislosti na přihlášeném uživateli horizontální menu – záložky. V případě, že je zobrazena stránka z dané podsekce, zvýrazní se aktivní záložka.
- *displayVMenu* - zobrazí v závislosti na přihlášeném uživateli rozbalovací víceúrovňové vertikální menu. Aktuální zobrazená stránka a případné nadsekce jsou zvýrazněny.
- *checkLog* – ověří informaci (na základě *session*), zda je uživatel přihlášen do systému
- *checkRights* – ověří, zda má přihlášený uživatel dostatečná oprávnění k prohlížení aktuální stránky
- *message* – zobrazí hlášku či chybu a případně ukončí zpracování stránky. Tato metoda se volá poté, když se během provádění programu vyskytne nějaká chyba nebo je třeba jinak reagovat. Podle určeného typu zprávy pak zajistí případné ukončení zpracování skriptu, eventuální odhlášení uživatele ze systému apod.
- *getThisFile* – zjistí a naformátuje jméno aktuální stránky resp. skriptu PHP

6.3.1.2 Page_rtf

Třída *Page_rtf* je odvozena ze třídy *Page*. Je navržena pro potřebu výstupu dokumentu do formátu RTF⁷, který vyžaduje odlišný způsob vytvoření stránky.

operace:

createPage – předefinování funkce předka - vytvoří stránku na základě zděděných parametrů *Parameters* a šablony *Template*. Šablona stránky se načte do proměnné *Html* a speciálně označené úseky jsou nahrazeny příslušnými hodnotami z pole *Parameters*. Před výstupem jsou odeslány hlavičky, které jsou potřebné ke správnému zobrazení ve formátu RTF.

6.4 Použité techniky a rozšíření

Nyní jsou popsány techniky, konkrétní knihovny a nástroje, které byly během implementace systému použity.

6.4.1 Databáze

Jeden z nejdůležitějších faktorů, které ovlivňují komplexní funkcionalitu systému, je komunikace s databází. Po prodiskutování a uvážení alternativ implementace přístupu k databázi byla nakonec s přihlédnutím na flexibilitu a komfort používání zvolena možnost nasazení již existující a odladěné knihovny ADODB.

6.4.1.1 ADODB

ADODB je knihovna ve stylu databázového obalu, která v PHP zjednodušuje mnohé úkony vztahující se k databázím. API knihovny ADODB je založeno na Microsoft ADO, což je knihovna pro přístup k datům, která se používá v jazyce Visual Basic a jiných produktech společnosti Microsoft.

Výhody použití této knihovny jsou především – zjednodušení a zpřehlednění přístupu k databázi, kdy lze volit z mnoha notací. Knihovna nabízí mnoho funkcí, které usnadňují vývoj i práci s databází. Jedná se o prostředky k výpisu databázových dotazů při ladění aplikace, ke zpracování chyb, automatické generování jednoduchých výpisů tabulek databáze apod. Dále jde např. eventuálně pomocí jednoho atributu změnit typ databáze, ke které má ADODB přistupovat.

V první fázi práce s touto knihovnou bylo využito její rozšíření Active Records, které nabízelo možnosti přístupu k databázi ve stylu ORM⁸, nicméně později se ukázalo, že klasický přístup bude přehlednější a jednodušší pro správu a pozdější rozšiřování aplikace. Není totiž třeba se učit nové a poměrně komplikované zápisy a syntaxe pro pokládání i jednoduchých databázových dotazů.

⁷ RTF - Rich Text Format – jazyk, který nabízí široké formátovací schopnosti při popisu textových dokumentů.

⁸ ORM - Object Relation Mapping – programovací technika používaná v prostředí OOP jazyků, která vytváří databázovou snadno použitelnou abstraktní vrstvu mapováním relační databáze na objekty.

6.4.2 Šablonovací systém

Ve větších projektech je z hlediska vyšší přehlednosti lepší udržovatelnosti a rozšiřitelnosti aplikace velmi vhodné oddělit prezentační vrstvu od vrstvy aplikační. Stejně tak tomu je i tomto vyvíjeném systému.

Oddělením PHP (vlastní kód a logika aplikace) od HTML (designu), je možné snadno měnit vzhled aplikace bez jakéhokoli zásahu do programu a tudíž i bez znalosti PHP a struktury aplikace. Nabízí se i možnost vytvořit několik různých šablon a mít vlastně k jedné aplikaci několik designů případně několik jazykových verzí.

Pro PHP jazyk existuje velké množství různých šablonovacích systémů, které se liší především svou syntaxí a různými možnostmi. Na základě konzultace a uvážení různých kritérií a návrhů bylo nakonec přistoupeno ke tomu, že se nebude využívat žádný již existující komplexní šablonovací systém, ale že se vytvoří jednoduchý vlastní. Ten pokryje veškerou potřebnou funkcionalitu⁹.

Vytvořený šablonovací systém pracuje s vytvořenými šablonami dokumentů, které jsou napsány v jazyce XHTML. Tyto šablony obsahují zástupce a vlastní tělo dokumentu. Zástupcem se rozumí ohraničený řetězec, který je během zpracování skriptu nahrazen skutečnou hodnotou.

Každá stránka aplikace je generována pomocí tohoto mechanismu šablon. Před jejím zobrazením je vytvořena instance objektu Page (popsáno v kap 6.3.1.1), které se předávají jako parametry všechny fragmenty stránky, kterými budou nahrazeny zástupci v šabloně. V případě, že nebudou všechny zástupci nahrazeny (např. neuvedením parametru a neexistenci jeho implicitní hodnoty), ve výsledném dokumentu XHTML se to neprojeví, neboť prázdné hodnoty jsou zobrazeny jako poznámky. Po nahrazení zástupců je vygenerována finální stránka aplikace.

6.4.3 Navigace

Vhodně zvolený systém navigace je jedním z hlavních pilířů, které musí správný informační systém obsahovat. Jelikož systém obsahuje velké množství různých souborů, bylo zvažováno jak, vyřešit jednoduchý a přehledný systém navigace. Nejprve byla implementována vlastní metoda uchování informací o adresářové struktuře aplikace a příslušných sekcích systému, ale toto řešení se ukázalo jako velmi nepřehledné a z hlediska dlouhodobějšího rozšiřování neefektivní. Proto bylo přistoupeno ke kroku poohlédnout se po nějakém jiném a lepším způsobu. Po zvážení alternativ nakonec bylo rozhodnuto o využití knihovny HTML_Menu z balíku PEAR.

6.4.3.1 PEAR: HTML_Menu

Knihovna HTML_Menu je součástí rozšíření repositáře PEAR a je speciálně navržena ke generování nabídek menu. Pomocí této třídy je možné snadno vytvořit a udržovat navigační strukturu, která je

⁹ Pozn.: pro správnou funkcionalitu systému generování menu bylo nutné použití šablonovacího systému HTML_Template_Sigma z balíku PEAR.

konfigurovaná na základě vícerozměrných hešovaných struktur. Lze definovat různé režimy výstupů do formátu XHTML.

Využití této knihovny při implementaci se nakonec ukázalo jako velmi užitečné, neboť po počátečních problémech s její správnou konfigurací, pokryla všechny požadavky, které byly na systém generování nabídky menu vyžadovány. Třída tohoto rozšíření generuje strukturu menu ze zadaného vícerozměrného pole, které je uloženo v konfiguračním souboru a je pro každou roli uživatele v systému odlišné. Toto pole je velmi přehledné a vhodné pro jednoduchou modifikaci.

Pro vhodné naformátování vygenerované nabídky menu se muselo přistoupit ke kroku využití šablonovacího systému `HTML_Sigma`, který je obsažen v repozitáři PEAR. Bohužel jinými způsoby nešlo docílit potřebného efektu zobrazení a uspořádání, nicméně i přes nutnost podstoupení tohoto opatření je přínos použité knihovny významný.

6.4.4 HTML formuláře

Pravděpodobně nejdůležitější částí vyvíjené aplikace je zpracování formulářových dat. Prvky a součásti jazyka XHTML pro práci s formuláři sice poskytují nástroje k pokrytí všech požadavků na jejich zpracování, nicméně neposkytují žádné zautomatizované činnosti a prostředky na zjednodušení a sjednocení práce s nimi. Proto bylo rozhodnuto, že se využijí možnosti knihovny `HTML_QuickForm`, která je součástí balíku PEAR.

6.4.4.1 PEAR: HTML_QuickForm

Tato knihovna usnadňuje práci s formuláři HTML. Namísto nutnosti tisku prvků formuláře jednoho po druhém stačí volat metody modulu definující strukturu formuláře. Po dokončení definování elementů je možné celý formulář zobrazit naráz. `HTML_QuickForm` automaticky zachovává výchozí hodnoty při odesílání prvků formuláře, zobrazuje chybové zprávy, zajistí jednotný vzhled formulářů apod.

Využití této knihovny při implementaci bylo více než účelné, neboť velmi zjednodušilo a zpřehlednilo tvorbu všech formulářů, používaných v aplikaci. Největším přínosem používání tohoto modulem bylo to, že obsahuje automatické mechanismy k validaci formulářových polí. Stačí proto zadat typ elementu formuláře či jeho případná pravidla a požadavky na vstupní hodnoty a metody knihovny `HTML_QuickForm` se samy postarají o jejich validaci. Tuto kontrolu je možné provádět už na straně klienta, kdy je automaticky vygenerován potřebný kód v jazyce Javascript, který kontrolu zajistí. Na druhou stranu bylo třeba proniknout do logiky práce s tímto modulem a leckdy i jednoduché věci vyžadovaly sofistikovaný přístup. Během implementace se rovněž projevil některé nedokonalosti a bugy této knihovny a musely se náležitě vyřešit či obejít.

6.4.5 Výstupy aplikace

Každá aplikace by nějakým způsobem měla umožnit zobrazit zpracovaná data a výsledky v nějaké srozumitelné a přehledné formě. Na každý systém jsou v tomto ohledu kladeny různé požadavky v závislosti na jeho účelu a nejinak je to u této aplikace.

Pro potřeby uživatelů vyvíjeného systému je u většiny jeho částí plně dostačující výstup v podobě XHTML stránek. Existují ovšem i případy, kdy je nutné nějaký výstup vytisknout či uložit. Pro tyto potřeby jsou navrženy tyto způsoby výstupu.

6.4.5.1 Formát XHTML

Standardně je v této aplikaci využito systému šablon (více kap 6.4.2) XHTML, které jsou formátovány použitím kaskádových stylů CSS. V nich jsou nastavena formátovací pravidla jak pro výstup na obrazovku (či jiné zobrazovací zařízení) tak pro výstup, který je určen k tisku. V druhém případě jsou upraveny především použité fonty, je vynechána nepotřebná grafika a nabídky menu. Tyto vlastnosti jsou nastavené u všech stránek aplikace.

Pro speciální výstup, kterým je dokument ověření je použito odlišné šablony. Je to z důvodů zjednodušení zápisu obecného CSS a zjednodušení editace tohoto dokumentu. U vygenerovaného ověření (resp. jeho šablony) musel zůstat zachován předepsaný vzhled na přání zadavatele. V případě potřeby je možné šablonu kdykoli zeditovat v jakémkoli textovém editoru či vytvořit novou.

6.4.5.2 Formát RTF

Během implementace projektu byl vznesen požadavek na to, ponechat možnost editace některých šablon i nějakým jednoduchým způsobem použitelným i bez nutné znalosti HTML a CSS. Nejprve bylo otestováno použití volně šiřitelné WYSIWIG¹⁰ aplikace FCK editor¹¹. Tato varianta dávala příliš velký stupeň benevolence při modifikování šablon a přinášela spíše negativní vlastnosti a zmatek (nevalidní HTML, nevhodně použité CSS). Proto bylo nakonec toto řešení zavrhnuto a přišlo se s jiným návrhem. Z důvodu, že stávající šablony dokumentů byly uloženy ve formátu DOC produktu Microsoft Word, který žádné volně dostupné knihovny PHP neumějí zpracovat, našla se podobná varianta. Ta spočívala v tom, že se dané šablony převedou do formátu RTF, který je již možné zpracovat za použití standardních technik.

Řešení za použití stávajících šablon pouze jejich konverzí do univerzálního textového formátu RTF je poměrně jednoduché. Do dokumentu stačilo pouze doplnit zástupce, stejně jako tomu je u používání šablon (popsáno v kap. 6.4.2). Tito zástupci jsou při zpracování skriptem nahrazeni adekvátními hodnotami. Pro implementaci této metody stačilo pouze mírně modifikovat šablonovací třídu (*Page*) a zajistit uložení výsledného souboru na server.

¹⁰ Akronym spojení „What you see is what you get“, kterým se označují editory, ve kterých se pracuje s daty v podobě, která je velmi podobná jejich následnému výstupu.

¹¹ Textový editor napsaný v JavaScriptu více na <http://www.fckeditor.net>.

6.4.6 Export dat

Požadavky na možnosti sdílení dat v některém univerzálním formátu jsou v dnešní době kladeny na většinu systémů. U této aplikace bylo zhodnoceno, že bude plně dostačující provést export databázových tabulek do dvou formátů – XML a CSV. Uživatel vybere jeden z formátů a určí, které tabulky databáze se mají exportovat. Poté systém výstup zapíše do souboru na serveru, k němuž uživatel dostane odkaz ke stažení.

6.4.6.1 XML

Export do souboru XML je proveden s využitím rozšíření knihovny PEAR: XML_Serialize. Toto rozšíření poskytuje funkce, které nabízejí sofistikovaný způsob k vygenerování XML souboru. Stačí nastavit pouze několik parametrů výsledného souboru a export se provede automaticky z předaného pole databázových údajů.

6.4.6.2 CSV

Pro export vybraných tabulek dat do výstupního formátu CSV bylo využito funkcí z knihovny ADODB, která byla stručně popsána v kapitole 6.4.1. Výstupní textový soubor obsahuje uložené hodnoty tabulek oddělených čárkou.

6.5 Uživatelské rozhraní

Zároveň s kvalitou poskytovaných služeb jsou na informační systémy kladeny je neméně důležitá forma s jakou daná aplikace s uživateli komunikuje. Je velmi důležité, aby program, který je kvalitně zpracován a poskytuje požadované funkce, neztrácel na svých hodnotách kvůli špatně navrženému nebo ne zcela dobře vyhovujícím rozhraní, pomocí kterého se svými uživateli komunikuje. Nejinak je tomu i u vyvíjeném systému. K důležitosti potřeby kvalitně zpracovaného grafického rozhraní přispívá i ten fakt, že systém bude přístupný pro zákaznickou sféru, tedy veřejnost.

Grafické uživatelské rozhraní bylo implementováno za využití nástrojů kaskádových stylů CSS, který v kombinaci s XHTML přináší široké nejen formátovací možnosti.

6.5.1 Layout a design

Nejdůležitějším aspektem uživatelského rozhraní je správné rozvržení struktury stránky. Dle požadavků a konzultací bylo vybráno dvousloupcové rozvržení stránky s hlavičkou a patičkou na začátku resp. konci stránky. Toto osvědčené schéma přináší řadu výhod a je uživatelsky přístupné a ve webových sférách velmi rozšířené. Velká výhoda tohoto rozvržení také spočívá v tom, že umožňuje bezproblémové, nenásilné ale účelné zakomponování navigačních prvků do stránky, aniž

by se snížila jeho přehlednost či variabilita. Neobsahuje rovněž žádné zbytečné prvky, které by působily chaoticky a rušily tak jasně definovanou strukturu stránky.

Grafický design je dalším důležitým měřítkem uživatelského rozhraní aplikace. Zde byla ponechána volná ruka čistě na řešiteli projektu a ze strany zadavatele nebyly zadány žádné požadavky. Snahou bylo docílit harmonického a decentního vzhledu, který ale nebude na druhou stranu působit nijak staticky až usedle. Proto bylo zvoleno i použití střídavého barevného schématu, který sice obsahuje některé moderní prvky, ale žádné neobvyklé výstřednosti či kombinace. V tomto ohledu je volba designu velmi subjektivní záležitost a je jasné, že ne každému bude vyhovovat. Nicméně cílem bylo pokud možno jít zlatou střední cestou a vynasnažit se najít rozumný kompromis mezi použitelností a atraktivitou.

Jelikož aplikace obsahuje dvě hlavní části – pro interní zaměstnance AMS a pro jejich klienty, je využito použití dvou mírně odlišných verzí CSS souborů. Bylo tak učiněno z důvodů potřeby vyšší přehlednosti a jednoduchosti designu uživatelského rozhraní sekce pro uživatele, V druhé sekci jsou nároky orientovány spíše na zvýšení míry „atraktivity“ pro klienty.

Náhledy obrazovek systému ze sekce pro zaměstnance a pro klienty jsou umístěny v příloze D.

6.5.2 Navigace

Jak již bylo zmíněno v návrhu, je nedílnou součástí kvalitního softwarového produktu vhodně zvolený systém navigace. Pro vyvíjenou aplikaci bylo vybrána forma dvou úrovní nabídek menu.

6.5.2.1 Horizontální menu

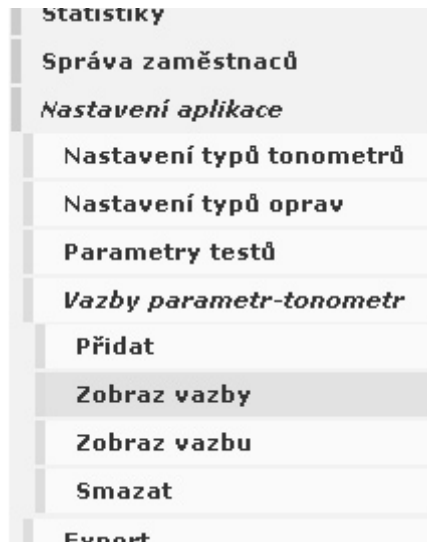
V záhlaví každé stránky se nachází globální stále viditelné menu, které zobrazuje hlavní sekce systému.. Je zobrazeno formou v dnešní době velmi populárních záložek. Tato interaktivní forma navigace dává uživateli přehled o tom, ve které části aplikace se právě nachází a dovoluje mu se kdykoli přemístit jinde.



Obrázek 6-5 Horizontální menu (záložky)

6.5.2.2 Vertikální menu

Druhou formou navigace, která je v této aplikaci použita, je hierarchické menu. Je umístěno v levé části stránek a dynamicky se mění v závislosti kontextu právě zobrazené stránky. Aktuální stránka je zde barevně odlišena, čímž je přesně znázorněno, kde se v systému uživatel nachází. Tato forma navigace znázorňuje i různé stupně zanořování, přičemž vyšší sekce aktuálního zanoření jsou odlišeny pro lepší orientaci.



Obrázek 6-6 Vertikální menu

6.6 Zpracování dat laboratoře

Jedním z nejdůležitějších úkolů, které bylo při tvorbě tohoto informačního systému nutné realizovat, bylo zpracování dokumentů laboratoře AMS. Zadavatel projektu, poskytl velké množství formulářů z různých etap revize (popsány v kap. 4.2.2), které ovšem bylo nutné sjednotit a zpracovat. Největším problémem bylo to, že ne každý z formulářů obsahoval správné položky a údaje a tato nekonzistence musela být eliminována. Bylo zapotřebí najít společné rysy dokumentů a na jejich základě navrhnout jejich vhodné zpracování.

Nakonec bylo přistoupeno k následujícímu řešení. Podle analýzy a návrhu, kdy byly tonometry rozděleny do dvou základních skupin, bylo stejně přistoupeno i k formulářům laboratoře. I když i v těchto dvou kategoriích dochází k dalším odchylkám mezi parametry tonometrů, daly se zde již vyzorovat mnohé společné vzory. Jednotlivé prvky formuláře byly tedy rozděleny na několik základních skupin pro každý ze dvou hlavních skupin tonometrů. Jednalo např. se o parametry s poznámkou, zaškrťovací položka, položka pro vyplnění hodnoty apod. V rámci těchto skupin byly rozlišeny jednotlivé elementy konkrétních položek a každé z nich byla přiřazena jednoznačná identifikace. Tato položka byla potom přiřazena zvlášť ke každému typu tonometru. To nese své výhody v tom, že je možné tyto vazby později editovat a či přidávat nové v případě, že přibude nějaký nový model tonometru apod. Na vazbu mezi typem tonometru a konkrétním prvkem formuláře je potom navázána konkrétní uložená hodnota revize

Tímto přístupem bylo dosaženo toho, že je v aplikaci zachován jistý stupeň možnosti konfigurace jednotlivých položek formulářů k daným typům tonometrů. Tyto změny se sice k krátkodobého hlediska nepředpokládají, nicméně se zda tato možnost nachází.

6.7 Zabezpečení aplikace

Ačkoliv žádné konkrétní požadavky na zabezpečení aplikace nezaznamenaly ani nebyly detailněji diskutovány, pro zachování potřebných vlastností systému zde bylo nutné některé mechanismy zavést.

6.7.1 Přihlašování

Hlavním zabezpečením aplikace je nutnost autentizace pro přístup k jejím službám. Přístup je zabezpečen pomocí přihlašovacího jména a hesla, které má přidělené každý uživatel systému. V případě zaměstnanců jsou tyto údaje poskytnuty administrátorem, klienti tyto informace získají poté, co budou zaregistrováni do systému buď elektronickou poštou, anebo jinou alternativní cestou.

Po zadání přihlašovacího jména a hesla uživatelem jsou tyto informace odeslány na server a zde porovnány s údaji uloženými v databázi. V případě, že se ověření uživatele nepodařilo, je uživatel požádán o opětné zadání přihlašovacích údajů.

Pokud se podařilo úspěšně autentizovat uživatele, dojde k jeho autorizaci. Podle zadaných přihlašovacích údajů je vytvořen objekt User, který se uloží do session pro pozdější využití. Rovněž je do session uložena informace o tom, že byl uživatel autentizován a dále časový limit, kdy přihlášení vyprší v případě delší doby nečinnosti. Při každém načtení další stránky aplikace, kdy tedy uživatel vykáže činnost, se tento časový údaj aktualizuje, aby nedošlo k jeho automatickému odpojení.

6.7.2 Uložení hesel

Hesla jsou v databázi uložena v podobě 128bitového otisku získaného pomocí funkce MD5¹². Použitím hashovací funkce je zabezpečeno případné vyzrazení hesla a nemožnost heslo z daného vygenerovaného řetězce zpětně získat. Tato metoda nese i nevýhody v tom, že např. případné zapomenuté heslo nelze dešifrovat a musí být nahrazeno jiným. Heslo jinak není nikde v aplikaci ani v session uchováváno, aby nemohlo dojít k jeho odcizení (ani jeho zahašované podoby).

6.7.3 Zabezpečení stránek

Kvůli potřebě rozlišení několika uživatelských rolí v systému musel být zaveden nějaký mechanismus práv. Princip implementovaného systému zabezpečení vychází z objektu User, který byl uložen do session během přihlášení. Každá stránka aplikace obsahuje informaci o tom, jakou úroveň zabezpečení vyžaduje. Tato informace je porovnána s hodnotou uloženou v session a na základě toho je povolen, popřípadě zamítnut přístup ke stránce.

¹² MD5 (Message-Digest algorithm 5) – hešovací funkce s otiskem (nebo-li kontrolním součtem) o velikosti 128 bitů. Popsáno v internetovém standardu RFC 1321.

Z důvodů zabezpečení některých PHP souborů, které nejsou přímo skripty aplikace a mohly by být volány zjištěním jejich názvů je využit jednoduchý mechanismus. V každém skriptu, který je volán je proveden test, zda je definovaná určitá konstanta. Ta je definována v PHP souborech, které přímo zajišťují běh aplikace. Pokud tedy dojde k požadavku na soubor z jiného místa, než jsou dané skripty, nedojde k jejich načtení.

6.7.4 Ošetření vstupů

Další z bezpečnostních aspektů, který je vhodné vyřešit je zpracování uživatelem zadaných dat. Nelze totiž předpokládat, že každý do vstupních polí formulářů zadá korektní hodnoty a je nutné na případný problém pružně reagovat. Velkým pomocníkem v této validaci zadaných hodnot jsou možnosti knihovny HTML:QuickForm, které je využita ke generování formulářů aplikace (popsáno v kap. 6.4.4.1). Její mechanismy umožňují hlídat formát vstupních dat již na straně klienta pomocí vygenerovaného kódu v jazyku JavaScript. Je nutné zamyslet se nad tím, co se stane v případě, že má uživatel ve svém webovém prohlížeči spouštění JavaScriptu zakázáno. V tom případě zmiňovaná knihovna rovněž pomůže, neboť automaticky kontroluje stejná pravidla na straně serveru a na případné chyby upozorní.

V případě zabezpečení uživatelských vstupů jsou velkým pomocníkem některé zabudované mechanismy PHP. V první řadě je to funkce „*addslashes*“, která pomocí před každé apostrofy, uvozovky nebo lomítka v požadavku POST, GET nebo cookie přidá zpětná lomítka, čímž eliminuje jejich potenciální nebezpečnost. Absencí ošetření vstupu takovýmto způsobem zneužívají útoky typu „*script injection*“, „*sql injection*“ a dalších. Druhou funkcí, která pomáhá při ošetřování uživatelem zadaných hodnot je „*htmlspecialchars*“. Ta převádí případně zadané HTML elementy na jejich textové entity a ošetří tak jejich případnou (a v případě této aplikace nežádanou) interpretaci. Tohoto nedostatku se využívá při útocích, kdy je stránce podvržen HTML kód, případně Javascript.

6.7.5 Další zabezpečení

Z hlediska bezpečnosti aplikace je vhodné zmínit i další možnosti a mechanismy, na které byl brán zřetel při implementaci. Kvůli eliminaci jakéhokoli pokusu nastavení proměnných v systému ze strany uživatele, které může být povoleno nevhodnou konfigurací serveru¹³, nejsou využívány žádné proměnné, které jsou globálně definovány a v případě, že jsou některé předávány metodou, kdy by byla možná jejich modifikace (např. metoda GET), je otestována jejich hodnota. Mezi další zabezpečení patří ošetření SQL dotazů. Byl kladen důraz na to, aby nebyly používány jako argumenty dotazů jakékoliv neošetřené, uživatelem zadané hodnoty. Některé soubory, jako je například provedený export databáze bylo nutné rovněž zabezpečit proti neoprávněnému přechtení. K tomu by případnému útočníkovi stačilo pouze znát jméno daného souboru a jeho cesty v souborovém systému

¹³ Např. povolení globálních proměnných (*register_globals*) na serveru Apache v *php.ini* apod.

webu. Tento nedostatek nelze nějakým rozumným způsobem vyřešit nástroji PHP a řeší je proto mechanismy serveru Apache. Přístup je povolen resp. zamítnul pomocí nastavení práv v souboru *.htaccess* uvedeného v adresářích, u kterých je zabezpečení vyžadováno.

6.7.6 Zhodnocení zabezpečení

Výše bylo popsáno několik konkrétních zabezpečovacích technik a mechanismů, které jsou použity ve vyvíjeném informačním systému. Zaručují potřebný stupeň bezpečnosti, ochrany a spolehlivosti chodu aplikace bez žádných vysokých nároků na její používání a provoz. Tyto prostředky jsou u podobných aplikací v dnešní době víceméně standardem základní, nicméně plně postačující, úrovně zabezpečení.

V případě nutnosti je samozřejmě možné zabezpečení vyvíjeného systému v několika krocích vylepšit. Jako první způsob se nabízí použití zabezpečené komunikace pomocí protokolu SSL¹⁴. Tím lze zajistit šifrování přenášených dat a autentizaci serveru pomocí digitálních certifikátů. Pro použití SSL je ovšem třeba mít na straně serveru nainstalovanou podporu SSL a také jej musí podporovat uživatelův webový prohlížeč.

Další návrh na zvýšení stupně zabezpečení se týká rozšíření možnosti autentizace uživatele. V aplikaci je tato činnost provedena pouze na základě zadání přístupového jména a hesla. Bylo by možné dále hlídat zároveň ještě např. IP adresy nebo jiné údaje uživatele. Útoky typu ukradení identity (tzv. „session stealing“), kdy může dojít ke zcizení session, se dá předcházet zavedením tohoto mechanismu. Potom se v případě, kdy se v průběhu práce s aplikací změnila IP adresa klienta, odepře přístup. Takovéto praktiky můžou ovšem způsobit určitý diskomfort uživatelům přistupujícím přes dial-up nebo těm, kterým se skutečně změnila IP adresa v průběhu práce. Je to ovšem daň za zabezpečení aplikace.

¹⁴ SSL - Secure socket layer - metoda zabezpečeného přenosu dat na přechodu mezi aplikační a transportní vrstvou TCP/IP vyvinutá společností Netscape

7 Rozšiřitelnost aplikace

V této kapitole jsou stručně shrnuty vlastnosti vyvíjeného informačního systému a je zde diskutována možnost jeho dalšího rozšíření.

7.1 Požadavky

V průběhu implementace informačního systému byly neustále kladeny nové a upřesňovány stávající požadavky na jeho funkčnost. Do jisté míry se většina nových požadavků dala okamžitě jednoduše realizovat, neboť se týkala spíše detailů. Objevily se i další nápady a připomínky, které by bylo eventuálně možné do aplikace přidat, ale ty svou obsáhlostí překračovaly zadaný rozsah této práce.

7.2 Diskuze rozšíření

Aplikace a její struktura byla navržena tak, aby dovolovala co nejsnadnější způsob rozšíření jejích funkcí či modulů. Proto byl informační systém rozdělen do několika samostatných částí, které jsou do jisté míry na sobě nezávislé.

Již z počátku zadání tohoto projektu bylo rozhodnuto o tom, že na jeho implementaci budou pracovat dvě osoby. Po fázi návrhu a specifikace požadavků však druhý řešitel svou účast odřekl a bylo nutné tedy upustit od implementace jeho části. Týkalo se to ekonomického modulu, který měl zajišťovat a spravovat procesy fakturace laboratoře AMS společnosti HNS mechanix s.r.o. Tato skutečnost je sama o sobě prvkem rozšíření, který je možný do aplikace přidat.

Mezi další rozšíření, které by bylo možné prodiskutovat a eventuálně implementovat by bylo vytvoření nějakého expertního systému. Ten by mohl obsahovat všechna pravidla a normy měření, které by potom maximálně zautomatizovaly laboratorní procesy AMS.

K dalším úvahám o rozšíření by mohlo patřit doplnění systému o pokročilou správu objednávek, která by pomohla spolu s fakturačním modulem centralizovat všechny činnosti AMS.

8 Závěr

Cílem této diplomové práce bylo seznámit se s prostředky tvorby aplikací v prostředí sítě Internet, provést analýzu požadavků konkrétní firmy a na základě toho navrhnout a implementovat nový informační systém.

Veškeré detaily projektu byly konzultovány s externím zadavatelem - spolujednatel firmou HNS mechanix s.r.o. V první fázi bylo nutné specifikovat požadavky nejen na funkčnost a vlastnosti nového systému, ale také na použité nástroje a technologie. Zde byl kladen důraz především na to, aby bylo navrženo vhodné řešení při zachování minimálních nákladů.

Poté byly se zadavatelem podrobně diskutovány požadavky na nový systém a jeho funkcionalitu, aby mohla být provedena jeho analýza a posléze detailní návrh. Jako opěrný bod částečně posloužil stávající jednoduchý systém, ze kterého bylo v určitých ohledech možno vycházet.

Po dokončení fáze návrhu se mohlo přistoupit k vlastní implementaci navrženého informačního systému. Během této, časově nejnáročnější, části projektu bylo nutné vyřešit mnoho problémů, které se objevily především na straně použitých nástrojů a rozšíření. Etapu implementace komplikovala také skutečnost, že vyvíjená aplikace byla během svého vývoje průběžně konzultována se zadavatelem, což s sebou neslo časté změny a upřesňování požadavků.

Diplomová práce a produkt, který v jejím rámci vznikl, znamená velký přínos pro své uživatele. Realizovaný informační systém nahradil dříve používanou aplikaci a změnil tak stávající pracovní postupy i využití informačních technologií v laboratoři firmy HNS mechanix s.r.o.. Nové řešení přispělo ke zvýšení pracovní efektivity, přehlednosti a centralizaci dat v laboratoři, ale hlavně pomohlo vylepšit kvalitu a rozšířit služby poskytované zákazníkům.

Mně osobně tato práce umožnila nejen získat nové poznatky v oblasti programování a vývoje podobných aplikací, ale také mi dala příležitost vyzkoušet si komunikaci se zákazníkem, návrh a následné zpracování praktického úkolu. Díky tomu jsem získal cenné zkušenosti, které jsou nezbytné pro budoucí uplatnění v praxi.

Literatura

- [1] *Wikipedia* [online]. 2007 , 11.5.2007 [cit. 2007-05-16]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Webov%C3%BD_server>.
- [2] HLAVENKA, J., SEDLÁŘ, R. *Vytváříme WWW stránky*. Praha, Computer Press 2000. 368 s.
- [3] KOSEK, J. *PHP – tvorba interaktivních internetových aplikací*. Praha, Grada Publishing 1999. 492 s.
- [4] GUTMANS, A. *Mistrovství v PHP 5*. Brno, Computer Press 2005. 655 s.
- [5] MASLAKOWSKI, M. *Naučte se mySQL za 21 dní*. Praha, Computer Press 2001. 505 s.
- [6] POŠMURA, V. *Apache – příručka správce WWW serveru*. Praha, Computer Press 2002. 311 s.
- [7] *Wikipedia* [online]. 2007 , 8.4.2007 [cit. 2007-05-16]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Unified_Modeling_Language>.
- [8] *HNS mechanix s.r.o.* [online]. 2007 , 8.4.2007 [cit. 2007-05-16]. Dostupný z WWW: <<http://www.hnsmechanix.com/index.php>>.
- [9] KRÁL, J. *Informační systémy*. Brno, Science 1998. 358 s.
- [10] W3C. *XHTML 1.0 The Extensible HyperText Markup Language* [online]. 2007 , 26.1.2002 [cit. 2007-05-16]. Dostupný z WWW: <<http://www.w3.org/TR/xhtml1>>.
- [11] *PHP - PHP project* [online]. 2007 , 1.5.2007 [cit. 2007-05-16]. Dostupný z WWW: <<http://www.php.net>>.
- [12] SKLAR, D. *PHP - moduly, rozšíření a akcelerátory*. Brno, Computer Press 2005. 344 s.

Seznam použitých zkratek

ADODB	knihovna pro PHP a Python poskytující abstraktní databázová vrstvu
AMS	autorizované metrologické středisko spol. HNS mechanix s.r.o.
CSS	kaskádové styly (Cascading HTML Style Sheets)
HTML	jazyk pro popis webových stránek (HyperText Markup Language)
HTTP	protokol pro přenos hypertextu – základní přenosový protokol WWW (HyperText Transfer Protocol)
MySQL	relační databázový systém
PEAR	framework pro PHP obsahující mnoho knihoven rozšířeních
PHP	objektově orientovaný skriptovací programovací jazyk (Hypertext preprocesor)
SGML	standardizovaný obecný značkovací jazyk určený k formálnímu popisu struktury dokumentů (Standard Generalized Markup Language)
SQL	univerzální deklarativní jazyk pro přístup k datům (Structured Query Language)
W3C	nezisková webová standardizační organizace (World Wide Web Consortium)
WWW	celosvětová počítačová síť známá jako Internet (World Wide Web)
XHTML	rozšiřitelný jazyk pro popis webové stránky (eXtensible Hypertext Markup Language)
XML	rozšiřitelný značkovací jazyk pro tvorbu strukturovaných dat odvozený ze SGML (eXtended Markup Language)
Zend	společnost vyvíjející engine PHP, název jádra PHP od verze 4

Seznam příloh

Příloha 1. Formuláře AMS

Příloha 2. Uživatelská příručka

Příloha 3. Obsah přiloženého CD

Příloha 1 – formuláře AMS

HNS mechanix s.r.o. Autorizované metrologické středisko
Zkušebna Schiötzových a aplanačních Goldmannových očních tonometrů
Havlíčková 2788,767 01 Kroměříž

ZÁPIS O VSTUPNÍ TESTU SCHIÖTZOVA TONOMETRU

Typ měřidla	Výrobní číslo	Datum vstupního testu	Č.zakázky
TS - 3			/ 2007

Zadavatel:	
------------	--

VSTUPNÍ TEST PARAMETRŮ MĚŘIDLA /dle specifikace č.5/ Poznámka

1	/	Kontrola měřidla na vlastním kalibru	
2	/	Kontrola shodnosti výrobních čísel na základně, pelotě a zákl.závaží 5,5g	
3	/	Optická kontrola povrchu peloty a základny	
4	/	Optická kontrola povrchu hrotu peloty	
5	/	Kontrola radiální vůle peloty v základně	
6	/	Optická kontrola ukazatele	
7	/	Kontrola povrchů hrotů A a B raménka	
8	/	Kontrola povrchové úpravy měřidla a stupnice	
9	/	Kontrola povrchu délky peloty	
10	/	Kontrola kompletnosti měřidla /závaží7,5g,10g,kalibr,kartáček,vložka/	

Kontrola mikrometrických údajů na ověřovacím stavu/v mm/při splnění podmínek 1-10:

Stupnice tonometru	Stanovená hodnota	Tolerance TS-3	Tolerance (x)ostatní typy	Naměřená Hodnota	Teplota v AMS (°C)
a/ 0.dílku stupnice	0,05	±0,01	±0,01	/	24°C
b/ 5.dílku stupnice	0,30	±0,01	±0,01		
c/10.dílku stupnice	0,55	±0,01	±0,015		
d/15.dílku stupnice(x)	0,80	-	±0,03		
e/18.dílku stupnice(TS-3)	0,95	±0,05	-		

Pohyb ukazatele o jeden dílek odpovídá posuvu peloty o 0,05 mm pro všechny tonometry.

Rozsah stupnice pro typ TS-3 platí od 0.do 18.dílku stupnice.

(x)Pro ostatní typy ZEISS,OCULUS,WINTER,BIRO,RIESTER platí rozsah stupnice od 0.do 15.dílku stupnice.

Nejistota měření $U=0,003\text{mm}\cdot k=2$ dle dokumentu EA 4/02. Použité etalony dle PJ 1/001/2002,Příloha P09/2.

Výsledek vstupního testu parametrů měřidla:

(označení kroužkem)

Samostatné ověření	Oprava I	Oprava II	Oprava III
---------------------------	-----------------	------------------	-------------------

(označit „ANO“ jen v případě neprovedeného ověření u měřidla, které nesplňuje metrologické vlastnosti, v ostatních případech proškrtnout)

Návrh na vyřazení	-----
-------------------	-------

Poznámka:

vstupní test provedl	Šprňa Jan
----------------------	-----------

Vstupní test je součástí zakázky a archivuje se v AMS

Obrázek Příloha 1 - 1 Formuláře AMS – vstupní test

Zápis o zkoušce očního tonometru

ZEISS

Výrobní číslo:

Výtah ze specifikace č.5 Komitétu pro standartizaci tonometrů americké oftalmologické a otolaryngologické akademie z 13.10.1959

Kontrola	Předepsaná hodnota	Tolerance	Naměřená hodnota
Váhové hodnoty	g	g	g
1. Pelota + páka + ukazatel			
a) na 5.dítku stupnice	5,5	±0,15	5,53
b) na 10.dítku stupnice	5,5	±0,20	5,53
2. Tonometr bez držátka	16,5	±0,5	16,315
3. Přídavná závaží pro			
a) 7,5g	2,0	±0,1	1,982
b) 10,0g	4,5	±0,02	4,481
Rozměrové hodnoty	mm	mm	mm
4. Základna tonometru			
a) průměr	10,1	±0,2	10,052
b) poloměr zakř.dolní pl.	15,0	±0,25	15
c) průměr konkávní části	9,0	minim.	>9
d) průměr otv. v základně	3,7	-0,5	3,2
5. Pelota			
a) průměr	3,0	±0,03	3,001
b) poloměr zakř.dolní pl.	15,0	±0,75	15
c) poloměr zakř.dol.okraje	0,255	±0,015	0,255
d) možnost vysunutí peloty	3,0	maxim	<3
Mikrometrické hodnoty			
a) 0.dítku stupnice	0,05	±0,01	0,044
b) 5.dítku stupnice	0,3	±0,01	0,292
c) 10.dítku stupnice	0,55	±0,02	0,532
d) 15.dítku stupnice	0,8	±0,03	0,776
(pohyb ukazatele o jeden dílek odpovídá posuvu peloty o 0,05mm)	minimálně v rozmezí od 1. do 15.dítku stupnice		
6. Ukazatel			
a) výška bříty	0,4	-0,08	0,4
b) šířka bříty	0,2	±0,02	0,2
c) vzdálenost od stupnice	1,0	max	0
7. Testovací blok			
a) průměr	11,0	maxim.	11
b) poloměr zakřivení	15,0	±0,04	15
Hodnoty v dílcích stupnice	dílky	dílky	dílky
8. Údaje na testovacím bloku			
a) zkušební blok rk=15mm	-1,0	±0,15	-1
b) vlastní blok rk=15mm	-1,0	±0,15	-1
9. Tření mezi pelotou a základnou ve st.sklonu			18°

Povrch	Požadovaná kvalita	Vizuální kontrola
10. dolní plocha základny	hladký bez poškozených míst	/
11. dolní plocha peloty	hladký bez poškozených míst	/
12. hrot peloty	hladký bez poškozených míst	/
13. kalibrační tabulka	Friedenwald 1955	

Použité etalony : viz PJ č. 1/001/2002, PŘÍLOHA č. 14/1

Tonometr	VYHOVUJE	NEVYHOVUJE
----------	----------	-----------------------

Rozšířená nejistota měření: 0,002 g pro K=2; 0,003 pro K=2 dle dokumentu EA4/02

Podmínky při zkoušce :	teplota na pracovišti	24°C
------------------------	-----------------------	------

Datum provedení zkoušky:
16.6.2005

Podpis:

Obrázek Příloha 1 - 2 Formuláře AMS – zápis o zkoušce

HNS mechanix s.r.o. Autorizované metrologické středisko

Zkušebna Schiötzových a aplanačních Goldmannových očních tonometrů

Havlíčková 2788,767 01 Kroměříž

POTVRZENÍ O OVĚŘENÍ STANOVENÉHO MĚŘIDLA

Měřidlo: **APLANAČNÍ OČNÍ TONOMETR**

Výrobce:

Typ:

Výrobní číslo:

Poslední ověření provedeno:

Měřicí rozsah: 0 –mN /0 - kPa pro apl.Ø 3,06mm/

Zkouška: provedena podle zkušebního protokolu.....normy ISO DIN 8612
a metodiky PJ č.1/001/2002,Příloha č.41

Použité etalony a kalibrace: podle metodiky Příručky jakosti č.1/001/2002,Příloha č.09/2

Vlastník měřidla:.....

**Měřidlo vyhovělo výše uvedeným požadavkům pro Aplanační oční tonometr
v souladu s §9,odst.2 zákona o metrologii č.505/1990 Sb.ve znění zákona č.119/2000 Sb.
a §6 vyhlášky MPO č.262/2000 Sb.ve znění vyhlášky č. 344/2002 Sb.bylo opatřeno úřední
značkou a letopočtem K1 na místech určených v certifikátu o schválení typu.**

Zkoušel:.....

V Kroměříži dne:..... Kulaté razítko Vedoucí AMS:.....

Toto potvrzení se vydává jako doklad o úředním ověření stanoveného měřidla.

Upozornění:

**Platnost ověření tohoto stanoveného měřidla je vyhláškou Ministerstva průmyslu
a obchodu ČR(dále MPO) č.345/2002 Sb.pol.2.3.1a) přílohy ve znění vyhlášky č.65/2006
sb. stanovena na jeden rok.**

**Platnost ověření zaniká dále v případech uvedených v §7,odst.2,písm.b/ až e/ vyhlášky
MPO č.262/2000 Sb. ve znění vyhlášky č.344/2002 Sb.**

Platnost tohoto ověření zaniká, jestliže:

- uplynula doba platnosti tohoto ověření
- byly provedeny jakékoliv změny,či úpravy měřidla
- měřidlo bylo poškozeno
- byly znehodnocena nebo odstraněna úřední značka K1
- při ztrátě metrologických vlastností

Obrázek Příloha 1 - 3 Formuláře AMS – potvrzení o ověření

Příloha 2 – Uživatelská příručka

Příloha 3 obsahuje uživatelskou příručku aplikace, která se skládá z návodu k instalaci a popisu jednotlivých částí programu včetně ukázek uživatelského prostředí aplikace.

Instalace

V této části je uveden stručný návod k instalaci, podrobnější instrukce jsou na přiloženém CD-ROM v souboru „readme.txt“.

Požadavky na server

Aplikace byla implementována za použití následujících nástrojů. Jsou zde uvedené verze, na kterých byl její provoz odladěn. Pro správný chod aplikace je tedy doporučeno použít tyto nebo vyšší verze.

- Apache v. 2.0.58 (<http://httpd.apache.org/download.cgi>)
- PHP v. 5.1.4 (<http://www.php.net/downloads.php>)
- MySQL v. 5.0.22 (<http://dev.mysql.com/downloads/mysql/5.0.html>)

Postup při instalaci aplikace

- Zkopírovat všechny soubory (z adresáře „system“ umístěným na přiloženém disku CD-ROM) do WWW adresáře na serveru.
- Nastavit přístupová práva k zápisu i čtení u adresářů „data“ a „dokumenty“, které byly nakopírovány v předchozím kroku na server.
- Vytvořit databázi a potřebné tabulky spuštěním sql skriptu „database.sql“ (z adresáře „sql“ umístěného na přiloženém disku CD ROM).
- Naplnění tabulek databáze spuštěním skriptu „data.sql“ (z adresáře „sql“ umístěného na přiloženém disku CD ROM).
- Nastavit přístupové jméno a heslo k databázi v souboru „db.php“ umístěného v adresáři „config“ v složce aplikace.

Postup při instalaci knihoven

Ke správnému provozu aplikace jsou nutné následující knihovny a rozšíření:

- ADODB 5 – <http://adodb.sourceforge.net/#download>
- PEAR - <http://pear.php.net/package/PEAR/download>
 - Serializer v. 1.47 (vyžaduje XML_Util v.1.28)
 - HTML_Menu v .1.14
 - HTML_QuickForm 1.162

- HTML_ Sigma 1.16

Všechny uvedené knihovny jsou na přiloženém CD-ROM (adresář „include“) a je nutné je buď nakopírovat, nebo použít instalaci PEAR (<http://pear.php.net/manual/en/installation.php>).

Požadavky na klienta

Uživatel aplikace musí mít přístup k Internetu a webový prohlížeč, který podporuje použité technologie XHTML, CSS2 a JavaScript. Doporučené prohlížeče a jejich verze jsou Internet Explorer 6.0, Firefox 1.5, Opera 9 případně vyšší.

Popis aplikace

K aplikaci se uživatelé připojí pomocí webového prohlížeče zadáním přístupové adresy severu. Na vstupní stránce jsou zobrazeny základní informace o systému a odkaz k zobrazení formuláře, který slouží k autentizaci do systému.

Přihlášení uživatele

Po uživateli se vyžaduje, aby se pro přístup do informačního systému přihlásil. Do zobrazeného formuláře je nutné vyplnit přístupové jméno, které je přidělo každému uživateli a odpovídající heslo. V případě nezadání hodnot je uživatel informován varovnou hláškou. Pokud se zdařil proces autentizace na základě zadaných údajů, je uživatel přesměrován na úvodní stránku systému.

Po přihlášení je uživateli zobrazena hlavní nabídka aplikace, odkud může vstoupit do libovolné sekce systému. Podle jeho role jsou zobrazeny dostupné části systému a v jejich kontextu jsou zobrazeny i ostatní navigační prvky - vertikální a horizontální menu.

The image shows a web page for user login. At the top is a green banner with the text 'HNSmechanix s.r.o.'. Below this is a navigation bar with two items: 'hlavní strana' and 'přihlášení'. The main content area is titled 'PŘIHLÁŠENÍ DO SYSTÉMU'. It contains a login form with two input fields: '*login:' and '*heslo:'. Below the fields is a button labeled 'Vstup'. A note below the button states '* denotes required field'. At the bottom of the page, there is a footer with 'Home |' on the left and '© 2006 HNS mechanix s.r.o. | Design by: RP |' on the right.

Obrázek Příloha 2 - 1 Přihlášení uživatele

Aplikační rozhraní pro Techniky

Pokud je přihlášený uživatel v roli technika, má přístup k sekcím systému, které mu umožňují následující činnosti.

Spravovat klienty

V této části má uživatel možnost spravovat všechny klienty laboratoře AMS. Tato sekce je rozdělena na následující části:

- **Přidat klienta** – zobrazí se formulář, kde uživatel vyplní informace o novém klientovi. Může zde zadat kontaktní a přihlašovací údaje stejně jako informace o instituci, pod kterou klient spadá. Povinné položky jsou označeny pomocí symbolu „*“ a jejich vyplnění se kontroluje již na straně klienta. Validace zadaných dat je vyžadována např. pro položky jako je email, kde je testován správný formát řetězce. Po zadání údajů a je zobrazen formulář s vyplněnými hodnotami a je vyžadováno jeho potvrzení. V případě, že uživatel zadal některé hodnoty chybně nebo nezadal požadované hodnoty vůbec je zobrazeno upozornění a je vyžadováno opětovné zadání. Pokud chce uživatel vrátit změny zpět a kupříkladu opravit některé položky, učiní tak volbou tlačítka „zpět“. Pokud zvolí tlačítko „potvrdit“, jsou data zpracována a uložena do databáze.
- **Zobrazit klienty** – tato volba vypíše seznam všech klientů zaregistrovaných u AMS. V seznamu jsou zahrnuty jak údaje o klientovi, tak počty jeho tonometrů. Po vybrání klienta se zobrazí jeho detailní informace. Stejně jako je možné zobrazit podrobné informace po zvolení tonometru.
- **Zobrazit klienta** – tato položka zobrazí vybraného klienta a všechny jeho údaje.

> **Zobrazení klienta**

Informace o uživateli	
titul:	MuDr.
*jméno:	Petr
*příjmení:	Srček
login:	xsršek
*heslo:	heslo
Informace o firmě	
ičo:	32510504
dič:	85420138
zařízení:	Ordinace Ráj
Kontakt	
telefon:	850478564
fax:	850478565
mobil:	602113222
ulice:	Jánská 25
město:	Brno
psč:	62000
email:	srsek@volny.cz
web:	
poznámka:	
posílat upozornění:	<input checked="" type="checkbox"/>
<input type="button" value="Vstup"/>	
<small>* denotes required field</small>	

Obrázek Příloha 2 - 2 Zobrazení detailů klienta

Spravovat tonometry

Tato sekce má za úkol spravovat informace o tonometrech a má následující podčásti.

- **Přidat tonometr** – zobrazí formulář pro zadání údajů o novém tonometru. Povinné položky jsou vyznačeny jako u ostatních formulářů symbolem „*“. Ke každému nově zadávanému tonometru je nutné určit klienta, kterému daný přístroj patří. Pokud není klient v systému zaregistrován, může tak uživatel učinit pomocí volby „zadat“. Po odeslání formuláře je vyžadováno potvrzení změn a data jsou odeslána ke zpracování a uložení do databáze.
- **Zobrazit tonometry** – touto volbou uživatel zobrazí všechny tonometry, které jsou registrovány u AMS. U každého tonometru jsou zobrazeny také informace o jeho typu a vlastníkovvi. Po vybrání položky je možné zobrazit detailní informace.
- **Zobrazit tonometr** – tímto příkazem se zobrazí detailní informace o vybraném tonometru. Na stránce jsou rovněž zobrazeny informace o revizích, které přísluší danému tonometru. Po výběru revize se uživatel přepne do detailního zobrazení jejich informací.

Spravovat revize

V této sekci má uživatel možnost zadávat a prohlížet jednotlivé položky a kroky revize.

- **Přidat revizi** – zobrazí formulář, kde očekává zadání základních informací o revizi. Dále je nutné specifikovat tonometr, ke kterému se daná revize vztahuje. V případě, že není zadán

žádný tonometr, má uživatel možnost zadat nový. Dále je zde možné zvolit druh prováděné opravy a případně ji popsat.

- **Zobrazit revize** – tato volba zobrazí všechny provedené revize. U každé z nich vypíše i tonometr, který byl předmětem jejího měření. Uživatel může zvolit prohlížení detailu vybrané revize i tonometru.
- **Zobrazit revizi** – po vybrání revize tato volba zobrazí její detailní informace. Je zde možné přepnout se do zobrazení detailních informací o jejích částech – vstupním testu a záznamu o zkoušce.
- **Vstupní test** – v této podsekci má uživatel možnost spravovat informace o vstupním testu.
 - **Vytvořit nový** – touto volbou se zobrazí formulář pro vytvoření nového vstupního testu vybrané revize. Jsou zde uvedeny jak informace o revizi, tak o tonometru, jeho typu a klientovi, kterému daný přístroj náleží. Konkrétní položky formuláře pro vyplnění vstupního testu jsou závislé na vybraném tonometru, resp. jeho typu. Za zmínku stojí informace o tom, zda byl vstupní test dokončen – v případě potřeby je možné dokončit jeho zadání později. Po vyplnění formuláře je vyžadováno potvrzení a poté jsou vyplněné hodnoty zpracovány a uloženy do databáze.
 - **Zobrazit** – tato volba zobrazí detailní informace o zvoleném vstupním testu. V případě, že je příslušná revize vybraného testu nedokončena, je uživateli povolena editace tohoto vstupního testu. Po zadání resp. aktualizování položek a potvrzení změn jsou data uložena do databáze.
- **Záznam o zkoušce** – tato sekce umožňuje správu záznamů o zkoušce.
 - **Vytvořit nový** – vybráním této akce uživatel zadává nový záznam o zkoušce vybrané revize. V zobrazeném formuláři jsou uvedeny informace o dané revizi, tonometru, jeho typu a majiteli. Položky vlastního záznamu o zkoušce jsou zobrazeny v závislosti na typu tonometru. Po vyplnění potřebných údajů a před jejich odesláním ke zpracování je nutné změny potvrdit.
 - **Zobrazit** – poté, co uživatel vybere záznam o zkoušce, jsou zobrazeny detailní informace příslušného testu. Pokud je tento formulář nedokončen, je umožněna jeho editace. Pro zpracování dat a uložení do databáze je nutné změny potvrdit.

➤ Zobrazení vstupního testu

Informace o revizi	
id:	18
obj.č.:	58
datum přijetí:	2007-05-21
poznámka k opravě:	poškrábaná pelota
typ opravy:	malá
Informace o klientovi	
jméno:	Jana Pyšná MuDr.
instituce:	nemocnice v sadech
Informace o tonometru	
sériové č.:	sb1
typ:	Schiotz - Biro
Informace o vstupním testu	
datum:	2007-05-21
teplota:	20
poznámka:	
provedl:	Josef Dvořák
Dokončen vstupní test:	<input checked="" type="checkbox"/>
Naměřené hodnoty	
Kontrola měřidla na vlastním kalibru:	<input type="text"/> <input checked="" type="checkbox"/>
Kontrola shodnosti výrobních čísel na základně, pelotě a zákl. závaží 5,5g:	<input type="text"/> <input checked="" type="checkbox"/>
Optická kontrola povrchu peloty a základny:	<input type="text"/> <input checked="" type="checkbox"/>
Optická kontrola povrchu hrotu peloty:	<input type="text"/> <input checked="" type="checkbox"/>
Kontrola radiální vůle peloty v základně:	<input type="text"/> <input checked="" type="checkbox"/>
Optická kontrola ukazatele:	<input type="text"/> <input checked="" type="checkbox"/>
Kontrola povrchů hrotů A a B raménka:	<input type="text"/> <input checked="" type="checkbox"/>
Kontrola povrchové úpravy měřidla a stupnice:	<input type="text"/> <input checked="" type="checkbox"/>
Kontrola povrchu délky peloty:	<input type="text"/> <input checked="" type="checkbox"/>
Kontrola kompletnosti měřidla /závaží 7,5g, 10g, kalibr, kartáček, vložka/:	<input type="text"/> <input checked="" type="checkbox"/>
poznámka, otestováno	
<input type="text"/>	
naměřené hodnoty	
0.dítku stupnice pož.hodnota = 0.05 (tolerance = 0.01) :	<input type="text" value="0.05"/>
5.dítku stupnice pož.hodnota = 0.30 (tolerance = 0.01) :	<input type="text" value="0.31"/>
10.dítku stupnice pož.hodnota = 0.55 (tolerance = 0.01) :	<input type="text" value="0.55"/>
15.dítku stupnice pož.hodnota = 0.80 (tolerance = 0.03) :	<input type="text" value="0.81"/>
<input type="button" value="Vstup"/>	

Obrázek Příloha 2 - 3 Zobrazení vstupního testu

Spravovat ověření

Tato sekce slouží ke správě a generování ověření dané revize.

- **Přidat osvědčení / zamítnutí** – vybráním této položky uživatel vytvoří nové ověření dané revize. V zobrazeném formuláři jsou uvedeny informace o revizi, klientovi, tonometru a jeho typu. Zadání údajů o ověření je nutné potvrdit a poté jsou data uložena do databáze.
- **Zobrazit seznam ověření** – tato akce zobrazí všechna vydaná ověření. Jsou zde uvedeny informace o příslušné revizi, tonometru a klientovi, jejichž detaily je možné zobrazit.
- **Zobrazit ověření** – tato volba zobrazí detailní informace o vybraném ověření.
- **Vygenerovat ověření** – po vybrání této položky jsou zobrazeny informace o zvoleném ověření. Jsou zde uvedeny údaje o příslušné revizi, klientovi, tonometru a ověření. Na této stránce jsou volby k vygenerování ověření do dvou různých výstupních formátů. První

možností je zobrazení certifikátu ověření v formátu HTML, který je určen k tisku. Druhá možnost je vygenerování souboru RTF. Tato volba se hodí např. do načtení souboru a jeho případné editaci v některém textovém editoru.

Přehledy

V této sekci má uživatel možnost sledovat různé statistiky a souhrny.

- **Tonometry v laboratoři** – tato volba vygeneruje seznam všech revizí, které nejsou ukončeny. Zobrazeny jsou zde informace o stavu revize, tonometrech a jejich klientech. Po získání detailních informací je možné zvolit příslušné akce.
- **Poslední ověření v intervalu** – v této sekci má možnost uživatel sledovat údaje o revizích, které proběhly v určeném intervalu. Je možné určovat parametry jako např. výsledek revize.
- **Poslední ověření k datu** – tato volba zobrazí stránku, kde je možné sledovat tonometry, kterým k danému datu končí ověření.
- **Statistiky** – tato stránka zobrazí statistické souhrny o proběhlých revizích. Živitel má možnost zadat časové období a parametry souhrnů – jaké informace chce sledovat.

The screenshot shows the HNSmechanix web application interface. At the top, there is a navigation bar with tabs: Klienti, Revize, Tonometry, Ověření, Přehledy, Dokumenty, Zaměstnanci, and Nastavení aplikace. The user is logged in as 'admin'.

The main content area is titled 'Výpis posledních revizí k datu'. It contains a search form with the following fields:

- datum:** 24 / 05 / 2007
- tolerance [dny]:** 5
- výsledek revize:** neomezenec

Below the form is a button labeled 'Vstup' and a link 'vygenerovat upozornění'.

The main data is presented in a table with the following columns: akce, revize, stav, přijato, ověřeno, poznámka, tonometr, and klient.

akce	revize	stav	přijato	ověřeno	poznámka	tonometr	klient
1	30	2007-05-18	2004-10-10	1 (výr.č.1)	Petr Srček		
7	30	2003-05-18	2007-05-09	2 (výr.č.sb1)	Jana Pyšná		
8	30	2007-05-18	2007-05-18	3 (výr.č.o1)	Jana Pyšná		
2	30	2007-05-18	2004-05-03	6 (výr.č.kow1)	Jana Pyšná		
13	30	2007-05-20	2000-05-20	7 (výr.č.t)	Bohuslav Tonič		
14	30	2007-05-20	2001-05-20	8 (výr.č.e)	Josef Skála		

At the bottom of the page, there is a footer with 'Home | © 2006 HNS mechanix s.r.o. | Design by: RP |'.

Obrázek Příloha 2 - 3 Výpis přehledu o revizích

Aplikační rozhraní pro Ředitele

Pokud je přihlášený uživatel v roli ředitele má všechny možnosti, jako výše popsání technici a některé další přidává.

Přehledy

- **přehledy o zaměstnancích** – v této sekci může uživatel sledovat některé informace o zaměstnancích AMS. Jde např. o délky trvání, které trvají jednotlivé části revize, účast zaměstnanců na jednotlivých revizích apod.
- **poslední ověření k datu** – tato volba je pro roli ředitele navíc rozšířena o možnost generovat dopisy a emaily s upozorněním. V případě, že je zobrazen seznam posledních ověření, je uživateli nabídnuta volba k vygenerování upozornění. Poté je možné zvolit buď automatického rozeslání emailů klientům, jejichž revize v k danému datu vyprší, anebo vytvoření automatických dopisů. Ty jsou vygenerovány do formy vhodné k tisku (každý dopis na nové stránce apod).

Aplikační rozhraní pro Administrátora

V případě, že je přihlášený uživatel administrátor, má neomezené přístupy ke všem položkám a sekcím aplikace. Obsahuje tedy nejen všechny výše popsané volby, ale mnohé další, které se týkají hlavně editace a mazání údajů a nastavení aplikace.

Správa klientů

Tuto sekci rozšiřuje o možnosti editace. V části „zobraz klienty“ má administrátor možnost vybrat klienty a po potvrzení akce je spolu s příslušnými záznamy odstranit z databáze. V sekci „zobraz klienta“ je administrátorovi umožněno provádět editace všech údajů.

Správa tonometrů

V této sekci má uživatel možnost mazat tonometry přímo ze seznamu v části „zobraz tonometry“. Tato volba vyžaduje potvrzení, přičemž dojde k výpisu upozornění a poté se vybrané údaje odstraní z databáze. Sekce „zobraz tonometr“ umožňuje administrátorovi editovat jakékoli údaje o vybraném tonometru.

Správa revizí

Uživateli, který je přihlášen jako administrátor, je umožněno editovat všechny položky revize. Ať už jde o údaje o vlastní revizi, Vstupním testu nebo Záznamu o zkoušce, má administrátor možnost editovat všechny jejich údaje. Uživatel má možnost odstranit celou revizi a s ní všechny její součásti.

Správa ověření

V této části systému může uživatel provádět editace všech údajů v sekcích „přidat osvědčení / zamítnutí“ a „zobraz ověření“.

Správa zaměstnanců

- **Přidat zaměstnance** – v této sekci může uživatel přidat nového zaměstnance AMS. Po zobrazení formuláře je nutné vyplnit osobní data, přihlašovací údaje apod. Je zde možné uvést i kontaktní informace na danou osobu.
- **Zobrazit seznam zaměstnanců** – tato volba zobrazí seznam všech zaměstnanců. Po vybrání konkrétní položky je zobrazen její detail.
- **Zobraz zaměstnance** – po vybrání této sekce jsou zobrazeny detailní informace o vybraném zaměstnanci. Zobrazená data lze přímo ve formuláři editovat. K zapsání změn do databáze dojde po potvrzení volby.
- **Smazat zaměstnance** - tato stránku je volána v případě odstraňování informací o zaměstnanci z databáze. Jsou zobrazeny jeho detailní informace a je vyžadováno proces mazání před jeho vykonáním potvrdit.

Nastavení aplikace

V této sekci může uživatel provádět změny v nastavení aplikace

- **Nastavení typů tonometrů** – tato sekce spravuje typy tonometrů.
 - **Přidat typ** – zobrazí formulář k zadání nového typu tonometru.
 - **Zobrazit typy** – tato volba zobrazí seznam typů tonometrů a jejich detailů.
 - **Zobrazit typ** – vybráním této položky uživatel zobrazí detail vybraného typu tonometru. Může jej v zobrazeném formuláři i modifikovat.
 - **Smazat typ** – tato volba odstraní zadaný typ tonometru.
- **Nastavení typů oprav** – v této části systému se dají spravovat typy oprav tonometrů.
 - **Přidat typ** – po vybrání této položky se uživateli zobrazí formulář, ve kterém je možné zadat nový typ opravy.
 - **Zobrazit typy** – tato volba zobrazí seznam typů oprav. Vybrání lze dané opravy odstranit z databáze.
 - **Zobrazit typ** – zobrazí vybraný typ opravy a umožní její editaci.
 - **Smazat typ** – po potvrzení volby smaže vybraný typ opravy.
- **Parametry testů** – v této sekci je možné editovat, přidávat a mazat formulářové položky Vstupního testu a Záznamu o zkoušce.
 - **Přidat** – tato akce umožňuje uživateli přidat nový typ parametru formuláře. Musí se zadat typ a požadované atributy nové položky. Po potvrzení zadání se údaje uloží do databáze.

- **Zobraz parametry** – tato volba zobrazí všechny parametry formulářů Vstupní test a Záznam o zkoušce. Uživatel si může nechat zobrazit detailní zobrazení vybrané položky. Zobrazené parametry je možné i odstranit.
- **Zobrazit parametr** – tato akce zobrazí detaily vybraného parametru testů. Umožňuje editaci těchto údajů, ke které dojde po potvrzení změn.
- **Smazat** – tato položka slouží k odstranění vybraných parametrů testů
- **Vazby tonometr parametr** – tato část slouží ke správě vazeb mezi typy tonometrů a parametrů testů (tj. Vstupního testu a Záznamu o zkoušce)
 - **Přidat** – pomocí této volby může uživatel zadat novou vazbu mezi typem tonometru a parametrem. Tyto položky uživatel vybere ze zobrazeného seznamu a přiřadí jim typ. Po potvrzení formuláře se zadané údaje vloží do databáze.
 - **Zobrazit vazby** – tato volba zobrazí seznam všech vazeb mezi typy tonometrů a parametrů testů. Ze seznamu je možné vybrat zobrazení detailu o každé vazbě. Uživatel může vazbu i odstranit.
 - **Zobrazit vazbu** – vybráním této položky dojde k zobrazení detailních informací o zvolené vazbě mezi typem tonometru a parametrem. Uživatel má možnost editovat položky této vazby.
 - **Smazat** – touto volbou dojde k odstranění zvolené vazby mezi typem tonometru a parametrem testu v databázi. Je vyžadováno její potvrzení.

- **Export**

Vybráním této akce má uživatel možnost provést export tabulek databáze do souboru. Nejprve je zobrazen formulář, ve kterém je nutné zadat jméno souboru a formát výstupu. Na výběr jsou formáty XML a CSV. Dále musí uživatel označit tabulky databáze, které chce exportovat. Po odeslání formuláře a jeho potvrzení dojde k uložení dat z databáze do specifikovaného souboru na serveru.

Příloha 3 – obsah přiloženého CD

Příložený disk CD-ROM obsahuje následující adresáře:

- document – tento dokument ve formátu „pdf“.
- system – kompletní zdrojové soubory aplikace.
- sql – „sql“ skripty pro vytvoření databázových tabulek a naplnění ukázkovými daty.
- include – knihovny, které jsou nutné pro správný chod aplikace

Více informací je uvedeno v souboru „readme.txt“ v root adresáři CD-ROM.