

Elektronická ordinace

E-health in Physician Office

Bc. Petr Korčák

Diplomová práce
2009



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2008/2009

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Petr KORČÁK**
Studijní program: **N 3902 Inženýrská informatika**
Studijní obor: **Počítačové a komunikační systémy**

Téma práce: **Elektronická ordinace**

Zásady pro vypracování:

1. Charakteristika webové aplikace, výhody, nevýhody.
2. Vypracování a analýza požadavků na informační systém.
3. Návrh systému na základě analýzy požadavků.
4. Realizace prototypu systému.
5. Zpracování uživatelské dokumentace k prototypu.
6. Zhodnocení přínosu systému a jeho možný rozvoj do budoucna.

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. Elektronické zdravotnictví (eHealth) [online]. 2008 , 27.11.2008 [cit. 2008-12-15].
Dostupný z WWW:
http://ec.europa.eu/health-eu/care_for_me/e-health/index_cs.htm.
2. ASP.NET 2.0 a C-sharp : tvorba dynamických stránek profesionálně / . Brno : Zoner Press, 2006. 1376 s.
3. Systems engineering principles and practice. – New York : Wiley, 2004. 465 s.
4. Softwarové inženýrství. Praha : Academia, 1991. 324 s.
5. ŠILHAVÝ, Petr, ŠILHAVÝ, Radek. Web-based Patient-Physician Communication. In SNÁŠEL, Václav. WOFEX 2008. Ostrava : Faculty of Electrical Engineering and Computer Science, VŠB – Technical University of Ostrava, 2008.

Vedoucí diplomové práce:

Ing. Petr Šilhavý

Ústav aplikované informatiky

Datum zadání diplomové práce:

20. února 2009

Termín odevzdání diplomové práce:

27. května 2009

Ve Zlíně dne 13. února 2009



prof. Ing. Vladimír Vašek, CSc.
děkan



doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem práce je vytvořit funkční prototyp webového rozhraní pro elektronické zdravotnictví. Tato aplikace bude umožňovat provádět základní požadavky lékařů na běžný chod ordinace, jako je objednávání pacientů, upozornění na termín očkování atd.

Teoretická část popisuje základní charakteristiku webových aplikací, analýzu a vypracování požadavků na informační systémy. Praktická část navrhuje systém na základě již zmíněné analýzy, zabývá se realizací funkčního prototypu a zpracování uživatelské dokumentace.

Klíčová slova: webová aplikace, Framework .NET, ASP.NET.

ABSTRACT

The aim of this project is create a functional prototype web interface for e-health. This application will allow doctors to perform the basic requirements of the normal functioning of a medical office, such as ordering patients or alert to nearing deadline vaccination, etc.

The theoretical part describes the basic features Web applications, analysis and elaboration of requirements for information system. The practical part proposes a system based on the above mentioned analysis. Next this part realizes the functional prototype and the processing of user documentation.

Keywords: Web application, Framework .NET, ASP.NET.

Děkuji vedoucímu diplomové práce ing. Petru Šilhavému za obětavou pedagogickou a odbornou pomoc a další cenné rady v průběhu řešení diplomové práce. Dále bych rád poděkoval své rodině, přítelkyni a její rodině za trpělivost a poskytnutí prostoru k vypracování diplomové práce.

Prohlašuji, že

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

Prohlašuji,

že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval.
V případě publikace výsledků budu uveden jako spoluautor.

Ve Zlíně, 27.5.2009

.....
Podpis diplomanta

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	10
1 INFORMAČNÍ SYSTÉMY	11
1.1 ÚVODNÍ STUDIE A ROZBOR ZADÁNÍ	12
1.2 ANALYTICKÉ MODELOVÁNÍ.....	13
1.2.1 Diagram datových toků (Data Flow Diagram – DFD).....	14
1.3 SYSTÉMOVÝ A OBJEKTOVÝ DESIGN	15
1.3.1 Metoda BORM (Business and Object Relation Modeling)	15
1.4 IMPLEMENTACE.....	17
1.5 ZKUŠEBNÍ PROVOZ A NASAZENÍ.....	18
2 WEBOVÉ APLIKACE	20
3 VYUŽITÉ TECHNOLOGIE	22
3.1 MICROSOFT .NET FRAMEWORK.....	22
3.1.1 Společný jazykový běhový modul – CLR	22
3.1.2 Knihovna tříd rámce .NET – FCL.....	24
3.2 WEBOVÉ APLIKACE ASP.NET	25
3.3 VZHLED APLIKACE	27
3.3.1 HTML (Hyper Text Markup Language).....	27
3.3.2 CSS (Cascading Style Sheets).....	28
3.3.3 JavaScript	28
3.4 DATABÁZE	29
3.4.1 Microsoft SQL Server 2008	30
II PRAKTICKÁ ČÁST	32
4 NÁVRH SYSTÉMU	33
4.1 UŽIVATELÉ SYSTÉMU	33
4.1.1 Lékař	39
4.1.2 Pacient	40
4.1.3 Administrátor	40
4.2 KOMUNIKACE.....	41
4.2.1 Objednání	41
4.2.2 Upozornění na očkování pomocí SMS	43
5 REALIZACE SYSTÉMU	44
5.1 DESIGN.....	44
5.2 TECHNICKÉ ŘEŠENÍ JEDNOTLIVÝCH WEBOVÝCH STRÁNEK	45
5.2.1 Nový lékař	47

5.2.2	Lékaři	48
5.2.3	Nový pacient	49
5.2.4	Výběr pacienta	50
5.2.5	Základní informace	50
5.2.6	Přehled lékařů.....	53
5.2.7	Karta pacienta.....	53
5.2.8	Očkovací průkaz.....	55
5.2.9	Objednání	57
5.2.10	Vypsání receptu.....	59
5.2.11	Rozvrh	60
6	UŽIVATELSKÁ DOKUMENTACE	61
6.1	ADMINISTRÁTOR	61
6.2	LÉKAŘ	62
6.3	PACIENT	64
7	ROZŠÍŘENÍ SYSTÉMU DO BUDOUCNA	66
7.1	VZHLED A ROZVRŽENÍ STRÁNEK	66
7.2	DATABÁZE UŽIVATELŮ	66
7.3	FUNKCE SYSTÉMU	67
7.4	OSTATNÍ ROZŠÍŘENÍ	67
	ZÁVĚR	69
	FINISH OF ENGLISH LANGUAGE.....	70
	CITOVANÁ LITERATURA	71
	SEZNAM OBRÁZKŮ	72
	SEZNAM TABULEK.....	73

ÚVOD

Oblast zdravotnictví je v našem státě jedním z nejdiskutovanějších témat. Existuje hodně věcí, které by se mohly změnit k lepšímu. Jednou z takových věcí je komunikace mezi pacientem a lékařem. Lékaři mají již dnes k dispozici software, který podobnou komunikaci umožňuje. Jedná se o jakousi elektronickou zdravotní kartu. Nicméně tento software pacienti ani lékaři příliš často nevyužívají.

Evropská unie přijala v roce 2004 akční plán rozvoje využívání informačních a komunikačních technologií ve zdravotnictví. Členské státy na základě tohoto plánu vypracovaly strategie rozvoje elektronického zdravotnictví. Z těchto důvodů se zrodil nápad vytvořit projekt elektronické ordinace.

Hlavní myšlenkou práce je položit základ k převedení všech odvětví zdravotnictví (lékárny, ordinace, ministerstvo zdravotnictví, atd.) do elektronické podoby. Práce se tedy zaměřuje na potřeby konkrétního lékaře, který potřebuje hlídat agendu své ordinace a na pacienty, kterým se usnadní komunikace s lékařem. Nejdůležitější funkcí systému je možnost objednání pacientů přes mobilní telefon (SMS zprávou) a přes webový formulář nebo funkce hlídání očkovacího období. Do systému mohou vstoupit pouze ověřeni uživatelé, kteří mají přidělena uživatelská práva podle své uživatelské role.

Aplikace je napsána pro platformu .NET (čti dot net) s využitím její části ASP.NET a její funkční logika je napsána v jazyce C# (čti sí-šarp). Databáze, která je v aplikaci využita pro správu uživatelů a dalších informací, je uložena na samostatném databázovém serveru.

Aplikace nemá nahradit žádný ze stávajících systémů, ale klade základ pro zjednodušení a nové vnímání chodu klasické ordinace.

I. TEORETICKÁ ČÁST

1 INFORMAČNÍ SYSTÉMY

Informační systém je systém pro sběr, udržování, zpracování a poskytování informací a dat. Příkladem informačního systému může být kartotéka, telefonní seznam, účetnictví. Systém nemusí být nutně automatizovaný pomocí počítačů a může být i v papírové podobě. Informační systémy mají poskytnout správné informace pro správné lidi ve správný čas. (1)

S informačními systémy úzce souvisí pojem *informace*. První pokusy o kvantifikaci přenášené informace provedl v roce 1924 Karl Küpfmüller. Podstatný zvrat v chápání informace přinesla v roce 1928 publikace R. V. L. Hartleyho „Přenos informací“. V autorově chápání odesílatel disponuje soustavou symbolů, z kterých vytváří posloupnosti¹. Nicméně ve čtyřicátých letech minulého století vytvořil základy *teorie informace* americký matematik C. E. Shannon, který považuje za míru informace *entropii*. O entropii uvažuje jako o míře neurčitosti, zatímco informaci považuje za míru určitosti. Při růstu informace klesá entropie a naopak. Odstraněním neurčitosti se získá informace.

Shannon vyšel z následující myšlenky: Máme-li dvě možnosti a dozvíme-li se, že jedna z nich platí, získáme nejmenší množství informace. Toto nejmenší množství informace – volbu ze dvou možností nazval Shannon *bit*².

Další pojem souvisejícím s informačními systémy jsou *data*. Daty míníme jakékoli zaznamenané poznatky či fakta. Jako zvláštní pojem zde vystupuje také *znalost* představující zobecnění poznání určité části reality.

(2)

¹ $I = n \log s$; kde s počet symbolů se stejnou pravděpodobností a n je celkový počet symbolů ve zprávě.

² Binary digit – dvojkové číslo.

Návrh informačního systému má v dnešní době tyto fáze:

- Úvodní studie
- Rozbor zadání
- Analytické modelování
- Systémový design
- Objektový design
- Implementace
- Zkušební provoz
- Nasazení

(1)

1.1 Úvodní studie a rozbor zadání

Úvodní studie a rozbor zadání, lze nejlépe vysvětlit na přípravě informační strategie v podniku. Manažer podniku, kterému je tvorba informačního systému svěřena, nejprve musí formulovat cíle, strukturu, rozsah a eventuálně omezující podmínky informační strategie. Dalším jeho krokem je analýza základních koncepčních materiálů. Pod tímto termínem si můžeme představit podnikovou strategii, marketingové analýzy, formulace cílů informačního systému, diskusi a formulování tzv. kritických faktorů. Dále následuje analýza současného stavu a zejména určení rozhodujících problémů provozu a dalšího rozvoje informačního systému. Po této analýze přichází na řadu návrh celé architektury (jednotlivých modulů – jejich podstatných vazeb), vymezení základní funkční struktury a řešení rozhodujících ekonomických, personálních a legislativních aspektů. Na konec se provádějí definice jednotlivých projektů.

Pro informační systémy by v současné době měly být významné následující vlastnosti:

- Schopnost adekvátně podporovat rozhodující cíle podniku, a to podle definovaných priorit (splňovat všechny požadavky).
- Vysoká vnitřní integrace dat a funkcí pokrývající všechny oblasti řízení podle definovaných cílů informačních systémů.
- Jasně definovaná celková architektura umožňující otevřenost informačního systému na úrovni technického, základního a zejména aplikačního

programového vybavení (plány na vybudování informačního systému a zavedení informačních technologií v podniku)

- Uzavřený informační systém – funguje nad daným operačním systémem, na přesně nadefinované moduly a nelze jej rozšířit.
- Otevřený informační systém – umožňuje využít různé operační systémy, umožňuje doplňování modulů (základní programové vybavení = operační systém).
- Možnost integrovat projekty různorodého charakteru lišící se nejen použitými technologiemi, ale také projekčními a provozními principy. Důležité je integrovat projekty tak, aby si systémy předávaly data.
- Schopnost efektivně zpřístupňovat jak interní datové zdroje a služby (vlastního informačního systému), tak zdroje a služby externí – internet, veřejné databáze (řídící pracovník by měl data dostat již vyříděná).
- Efektivní využívání a vzájemná provázanost různých technologií práce s daty – relační databáze, hypertext, textové editory, tabulkové procesory.
- Schopnost realizovat on-line propojení na IS obchodních partnerů, finančních institucí a dalších organizací na principech elektronické pošty.

(3)

1.2 Analytické modelování

Při analýze systému je zapotřebí vytvořit několik modelů, které představují napodobeninu reálného systému. Tvorba modelů systému tvoří největší část práce systémového analytika. Tyto modely zachycují všechny důležité rysy a zakrývají nepodstatné rysy systému.

Mezi metody, používaných při tvorbě informačních systémů, patří například procesně orientované přístupy, datově orientované přístupy, kombinace obou metod. Všechny tyto metody jsou součástí strukturované analýzy.

- Funkčně orientované modely systému – na systém se pohlíží jako na množinu funkcí transformujících data. Patří sem například diagram datových toků (Data Flow Diagram – DFD), diagram datových toků s řízením (Control Data Flow Diagram – CDFD) a minispecifikace.
- Datově orientované modely systému – systém je chápán jako úložiště, ze kterého se zpětně získávají transponovaná data. Hlavními představiteli jsou

entitně relační diagram (Entity Relationship Diagram – ERD) a datový slovník.

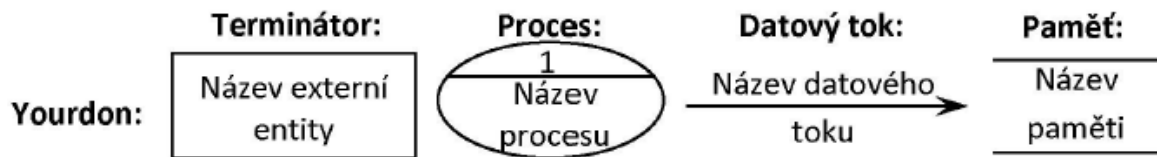
- Kombinace předchozích metod – zástupcem této kategorie je Yourdonova moderní strukturovaná analýza (Yourdon Modern Structured Analysis – YMSA).

(4)

1.2.1 Diagram datových toků (Data Flow Diagram – DFD)

Diagram datových toků je jedním z nejpoužívanějších modelovacích nástrojů strukturované analýzy, který poskytuje funkčně orientovaný pohled na systém. Slouží tedy k modelování funkcionality systému. DFD zobrazuje systém jako síť procesů. Tyto procesy plní určité funkce a pomocí datových toků si mezi sebou předávají data.

- **Terminátory** – reprezentují externí entity, které patří do okolí systému a se kterými systém komunikuje. Všechny informace, které systém přijímá nebo vysílá, jsou vysílány, respektive přijímány terminátory. Terminátory jsou nejčastěji osoby nebo skupiny osob, ale mohou to být i jiné systémy, se kterými náš systém komunikuje. Analytik, ani systém nemůže změnit obsah terminátorů nebo způsob jakým pracují.
- **Procesy** – jsou jediné části systému, které převádějí vstupy na výstupy. Každý proces by měl být jednoznačně identifikován a vhodně pojmenován, buďto jedním slovem, jednoduchou větou nebo frází. Jméno vyjadřuje, co daný proces dělá. Ke každému procesu na DFD existuje buď minispecifikace, nebo je dekomponován na nižší úrovni DFD, kde jsou znázorněny jeho subprocessy.
- **Datové toky** – popisují pohyb informačních paketů nebo fyzických materiálů mezi jednotlivými částmi systému. Datové toky jsou pojmenovány podle toho, jaká data přenášejí. Některé datové toky nemusí být pojmenovány, pokud je zřejmé jaká data přenášejí. Typicky to jsou datové toky směřující z nebo do paměti. Šipka znázorňuje směr toku dat. *Řídící toky*, které neobsahují žádná data, se zakreslují přerušovanou čarou.
- **Paměti** – jsou pasivní prvky systému, kde se data ukládají pro pozdější zpracování. Povolené operace jednoho datového toku nad paměti jsou buď nedestruktivní čtení, zápis, modifikace nebo mazání.



Obrázek 1. Digram datových toků

(4)

1.3 Systémový a objektový design

Pod objektivě orientovaným přístupem si většina odborníků v IT představí především jeho přínosy do oblasti implementace informačních systémů – tedy do oblasti analýzy a návrhu softwarových struktur a jejich následné implementace pomocí objektových programovacích jazyků a někdy také i objektových databází.

Z objektivě orientovaného procesního modelu lze dobře s aktivní pomocí zadavatelů najít funkce, strukturu, rozsah požadovaného systému a také role budoucích uživatelů vytvářeného systému.

(5)

1.3.1 Metoda BORM (Business and Object Relation Modeling)

Tato metoda je od počátku orientována na podporu tvorby objektivě orientovaných softwarových systémů založených na čistých objektivě orientovaných programovacích jazycích a vývojových prostředích, jakými jsou například prostředí Smalltalku a nerelační objektové databáze. BORM³ je možné využít nejen ve tvorbě softwaru, ale i k analýze požadavků na projektovaný systém a na modelování business procesů.

BORM lze charakterizovat pomocí následujících tří vlastností:

- 1) BORM je navržen jako metoda, která pokrývá všechny fáze vývoje softwaru. Velká pozornost je věnována úvodním fázím projektu a postupům, jak najít objekty v zadaném problému a zkontrolovat jejich správnost. Techniky z těchto fází lze používat samostatně pro modelování procesů i takových systémů, které nemají přímý vztah k tvorbě softwaru.

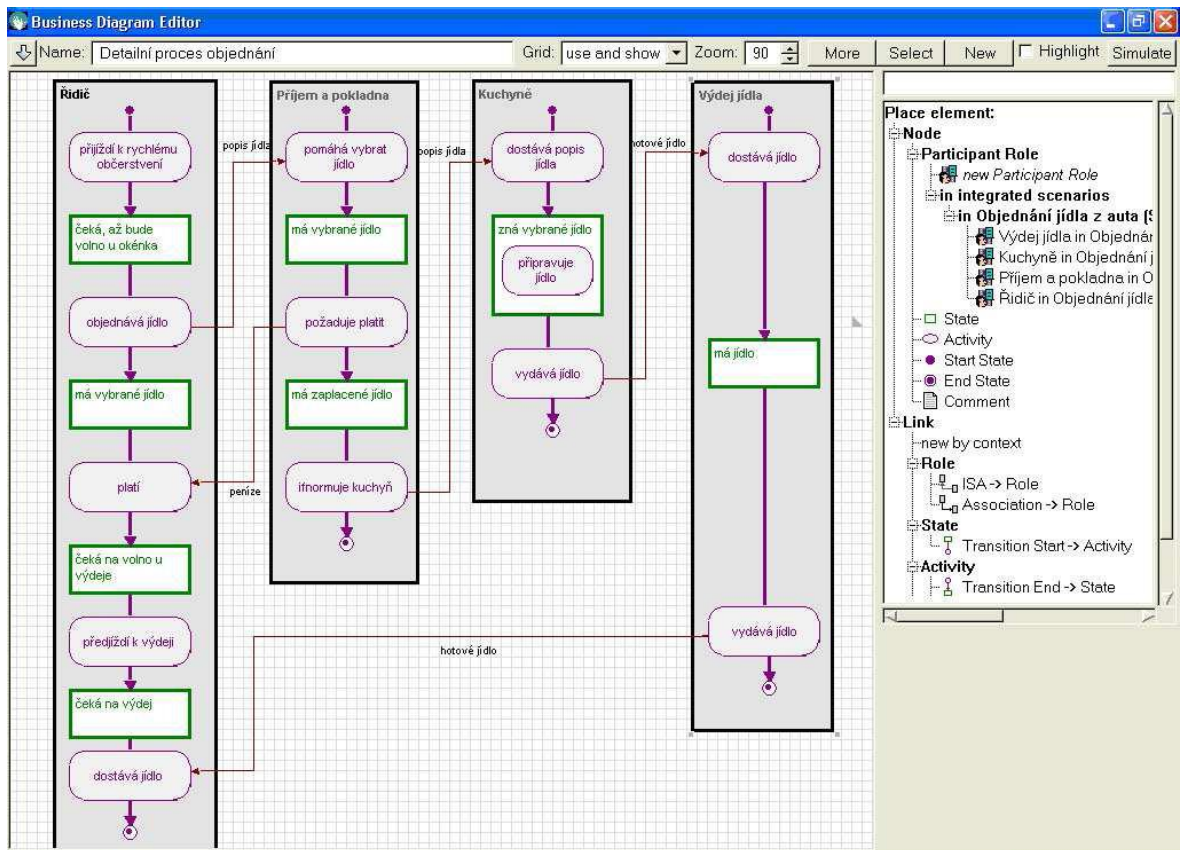
³ Více informací o této metodě v citované literatuře.

- 2) BORM pro každou jednotlivou fázi životního cyklu využívá v diagramech jen omezenou sadu pojmů. Předpokládá se totiž, že během projektování dochází k postupným přeměnám pojmů na jiné. Například ve fázi analýzy se nepoužívají pojmy jako agregace, jednoduchá či vícenásobná dědičnost, protože tyto pojmy jsou relevantní až pro implementaci. Naopak pojmy jako stav, přechod či asociace jsou používány během analýzy, ale ve fázi implementace, kdy se snažíme model přizpůsobit cílovému implementačnímu prostředí, se s nimi již nepracuje. Nejde jen o postupné zvyšování úrovně detailu ve vytvářeném modelu, ale skutečně o řadu transformací modelu v průběhu životního cyklu.
- 3) V BORMu je každý pojem reprezentován shodnými symboly bez ohledu na to, jestli se jedná např. o diagramy datové struktury nebo komunikací mezi objekty. BORM používá pro znázorňování potřebných pojmů většinu symbolů shodně s jazykem UML, ale dovoluje v jednom diagramu znázornit například posílání zpráv mezi metodami různých objektů v různých stavech. Tento přístup dovoluje vyjádřit konzistentním způsobem některé žádoucí detaily softwarové konstrukce, které lze výhodně aplikovat především při návrhu pro čistě objektově orientované programovací jazyky. Tento originální způsob nahrazuje tvorbu více od sebe oddělených třídních, stavových a kolaboračních diagramů a také dovoluje zobrazit větší množství spolu souvisejících informací. Samostatné stavové či interační diagramy jsou však samozřejmě také používány.

Pro využití předchozí metody v praxi je potřeba mít tzv. CASE nástroje. Některé nástroje však nepodporují všechny potřebné etapy (requirement engineering, analýza a návrh informačních systémů). Mezi nástroje, které tyto etapy podporují, patří například finský CASE nástroj MetaEdit⁴ a český CASE nástroj Craft.CASE⁵. Oba nástroje dovolují modelovat pomocí metody BORM.

⁴ Oficiální webové stránky [zde](#).

⁵ Oficiální webové stránky [zde](#).



Obrázek 2. Příklad využití nástroje Craft.CASE

(5)

1.4 Implementace

Většina systémů se implementuje jako tzv. Data Warehouses (DW). Jedná se o architekturu, která transformuje operativní data do jiné podoby, u které se bere ohled například na čas a rychlost následných dotazů. Tato data se nemění, mohou se transformovat z více zdrojů (např. od dodavatelů) a jsou aktualizována v časových intervalech. Nad nimi se dělají statistiky či analýza. To je poslední fáze, tzv. OLAP⁶.

Opakem DW jsou OLPT⁷ systémy, které jsou často přirovnávány k „výrobě“ podniku, DW pak ke „skladování“ výrobků, následně OLAP systémy jsou pak jakýmsi „prodejem“.

⁶ Online Analytical Processing

⁷ Online Transaction Processing Systems

Je zřejmé, že OLAP systémy jsou rozšířením OLTP systémů, také jejich návrh je složitější. Je zde použita tzv. multidimenzionální architektura. Další dimenzí je zde čas, oblast či obchodník. OLAP systémy jsou tak specifické, že se v nich může porušovat například normalizace a data jsou v těchto systémech velmi řídká.

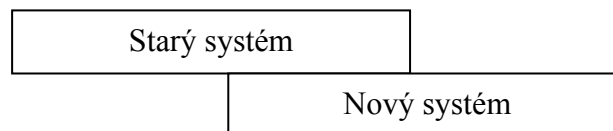
Systémy OLAP jsou implementovány buď nad relačními databázemi, nebo nad speciálními (zejména objektovými) OLAP databázemi. Z dnešních systémů jmenujme například Intersystem Caché nebo Oracle OLAP.

(1)

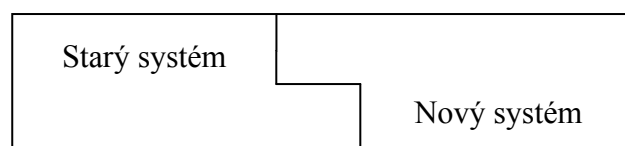
1.5 Zkušební provoz a nasazení

Existují čtyři strategie zavádění informačních systémů. Každá z těchto strategií má své klady a zápory.

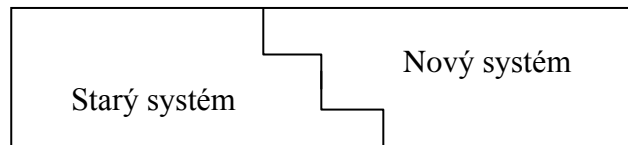
- 1) **Souběžná strategie** – informační systémy běží souběžně, většinou jeden výrobní cyklus (dekáda, měsíc). Výhodou této strategie je, že pokud nový systém není plně provozuschopný, je možné používat data ve starém systému. Nevýhodou je nutnost zajistit provoz dvou systémů (veškeré operace probíhají dvakrát).



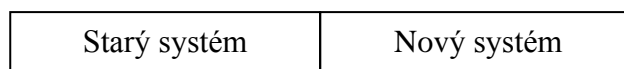
- 2) **Pilotní strategie** – tato strategie vybere jeden modul z požadavků na informační systém a na tom si vyzkouší nový systém, pokud vyhovuje, zavedeme informační systém do všech oblastí. Při aplikaci této strategie se odstraní problémy a zaškolí se pracovníci, což je výhoda. Je nutné si ale dát pozor na výběr modulu.



- 3) **Postupná strategie** – zde se zavádí vše postupně, jeden modul za druhým. Používá se u rozsáhlých úloh a problémů. Klad této strategie tkví v soustředění pouze na jeden modul a řešení jeho problémů. Jako zápornou vlastnost lze uvést zdlouhavý zaváděcí proces informačního systému.



- 4) **Nárazová strategie** – k nasazení nového systému dochází téměř ihned, většinou k jednomu dni. Výhodou jsou minimální nároky na souběžný provoz. Při zavádění systému může dojít k problémům, což je nevýhoda této strategie.



(3)

2 WEBOVÉ APLIKACE

V softwarovém inženýrství je aplikace poskytovaná uživatelům z webového serveru, přes počítačovou síť Internet nebo přes vnitropodnikovou síť Intranet, nazývána **Webovou aplikací**. Webové aplikace jsou používány pro implementaci mnoha podnikových i jiných informačních systémů, ale i freemailů, internetových obchodů, online aukcí, diskusních fór, atd.

V dřívějších typech aplikací typu klient-server bývalo časté, že každá aplikace měla svůj vlastní klientský program, který sloužil jako její uživatelské rozhraní a musel být instalován na osobním počítači každého uživatele. Aktualizace serverové části typicky vyžadovala i aktualizaci klientských programů na každé pracovní stanici, což zvyšovalo náklady na podporu a snižovalo efektivnost zaměstnanců. Oproti tomu webové aplikace generují dynamicky sérii webových stránek ve standardním formátu HTML/XHTML, který je podporován běžnými prohlížeči. Obecně je každá jednotlivá webová stránka dodána prohlížeči jako statický dokument, ale sled takových stránek může vyvolat pocit interaktivity, např. díky reakci na vstup uživatele do formulářových prvků vložených do kódu stránky. Pro přidání dynamických prvků do uživatelského rozhraní se používá skriptovací jazyk JavaScript. Webový prohlížeč v průběhu běhu aplikace interpretuje a zobrazuje stránky a funguje jako univerzální klient pro libovolnou webovou aplikaci.

Webové aplikace jsou obvykle strukturovány jako třívrstvé. V té nejběžnější formě je webový prohlížeč první vrstvou (prezentační), nástroje pro dynamické generování stránek (např. CGI, PHP, javové servlety nebo ASP) je vrstvou střední (logickou) a databáze je vrstvou třetí (datovou). Webový prohlížeč posílá požadavky střední vrstvě, která je obsluhuje prostřednictvím dotazů do databáze (resp. její aktualizací) a generováním uživatelského rozhraní. Webové aplikace mají spoustu výhod i nevýhod.

Mezi nevýhody lze zařadit například napadení serveru virem. Výpadek serveru, ať už z jakýchkoli příčin, způsobí, že webová aplikace je nedostupná a to má vždy velmi nepříjemný dopad na chod systému, který webová aplikace představuje. Webové rozhraní v některých směrech omezuje funkčnost a možnosti klienta. Metody známé z desktopových aplikací, jako je například vykreslování na obrazovku či obecné techniky jako „drag and drop“, nejsou standardními technologiemi prohlížečů podporovány. Tvůrci webů pro přidání funkčnosti často používají skriptování na straně klienta, hlavně pro vytvoření dojmu interaktivity bez nutnosti znovunačtení stránky, které jinak řadu uživatelů

zdržuje. V poslední době se začínají používat technologie, které umožňují spolupráci skriptů na klientské straně se serverovou částí aplikace. Jedním z příkladů těchto technologií je AJAX. AJAX je technika vývoje webu využívající kombinaci HTML, JavaScriptu a rozhraní XMLHttpRequest⁸. AJAX tedy umožňuje načítat klientským skriptům informace ze serveru, aniž by bylo třeba obnovovat celou stránku.

Mezi největší výhody zcela bezpochyby patří všudypřítomnost webového prohlížeče jako jejich klienta. Další takovou výhodou je schopnost aktualizovat a spravovat webové aplikace bez nutnosti šířit a instalovat software na potenciálně tisíce uživatelských počítačů. Co se týká technických aspektů, pak podstatnou výhodou vývoje webových aplikací, stavějících na standardních funkcích prohlížeče, je jejich schopnost pracovat podle určení bez ohledu na operační systém či jeho verzi instalovanou na daném klientském počítači. Místo psaní variant aplikace pro Windows, Linux, Mac OS X a další operační systémy stačí teoreticky aplikaci napsat jednou a nabídnout téměř kdekoliv. V praxi ale nekonzistentní implementace HTML, CSS a další specifikace jednotlivých prohlížečů způsobují problémy. Navíc mají uživatelé možnost nastavit způsob zobrazení ve svém prohlížeči (např. zvolit jiný řez či velikost písma, barvy či vypnout podporu skriptování), což může rušit jednotný vzhled aplikace.

(1)

⁸ Rozhraní umožňující webovým aplikacím komunikaci mezi serverem a klientem prostřednictvím protokolu HTTP.

3 VYUŽITÉ TECHNOLOGIE

3.1 Microsoft .NET Framework

Microsoft .NET Framework je platforma pro vytváření a provozování aplikací. Jejimi základními komponenty jsou společný jazykový běhový modul⁹ a knihovna tříd¹⁰ rámce .NET. CLR abstrahuje služby operačního systému a slouží jako vykonávací jádro pro *řízené aplikace*¹¹. FCL poskytuje objektově orientované rozhraní API, do něhož řízené aplikace zapisují. Při psaní aplikací v rámci .NET tedy není potřeba využívat různé technologie a nástroje, jako jsou API Windows, MFC, ATL, COM, ale stačí využívat pouze FCL. Samozřejmě lze při psaní aplikací v rámci .NET volat funkce API nebo objekty COM, ale pak je zapotřebí přechod z řízeného kódu na *neřízený kód*¹². Takové přechody omezují výkonnost a správce systému je dokonce může zakázat.

.NET Framework podporuje různé programovací modely. Mezi tyto modely patří zápis webových služeb, konzolové aplikace, aplikace GUI (formuláře Windows), webové aplikace a služby Windows. Celý rámec také pomáhá „konzumovat“ webové služby – tedy zapisovat klienty webových služeb.

(6) (7)

3.1.1 Společný jazykový běhový modul – CLR

Při vytváření aplikace programátor obvykle napíše nějaký kód ve vybraném programovacím jazyce, kompilátor jej zkompiluje do binárního formátu a nakonec se aplikace spustí a vykoná. Kdykoli chceme takovou aplikaci spustit na počítačích odlišného typu (např. PC a Macintosh), musí se znovu zkompilovat.

I v systému .NET musí programátor napsat kód, který se po sléze kompiluje. Cílem této kompilace není ale binární kód, ale zvláštní „mezilehlí“ jazyk, nazvaný Microsoft Intermediate Language (MSIL). Tento jazyk představuje jakousi zkrácenou reprezentaci veškerého zapsaného programového kódu. Při kompilaci do MSIL produkuje aplikace takzvaná **metadata**, což jsou popisné informace o aplikaci. Metadata tak říkají, co aplikace

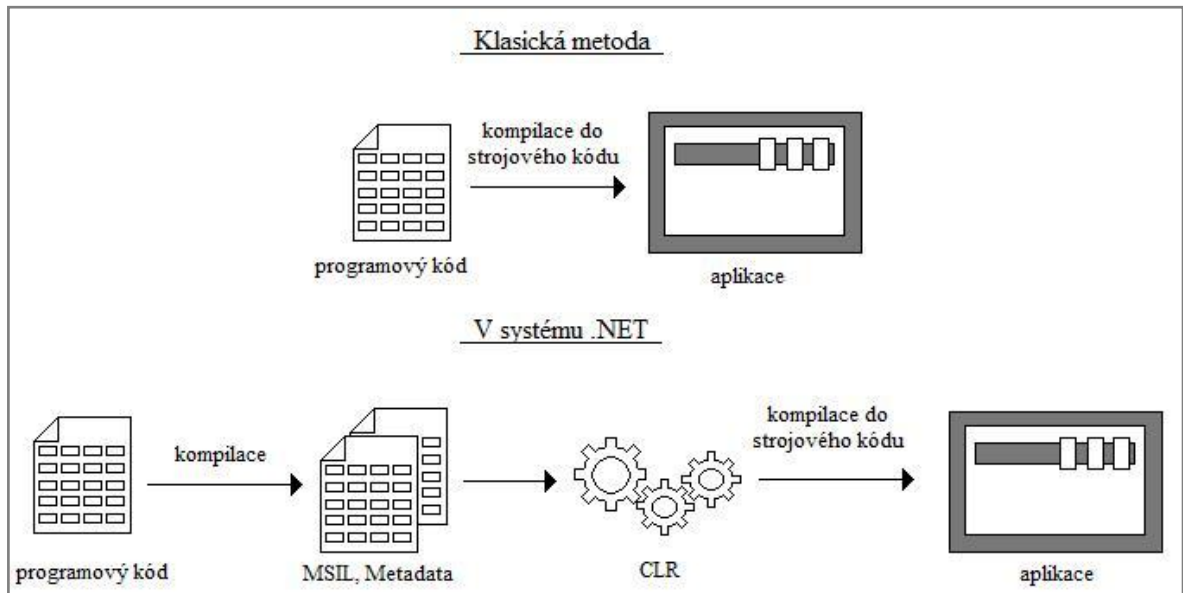
⁹ CLR – Common Language Runtime

¹⁰ FCL – Framework Class Library

¹¹ Aplikace, jejichž všechny akce podléhají schválení u CLR.

¹² Nativní strojový kód, který běží bez pomoci runtimeového modulu.

umí, kam patří a podobně. Jakmile je potřeba spustit program, tak tuto úlohu převezme CLR. Rozdíl mezi klasickým procesem a prací v systému .NET ukazuje následující obrázek.



Obrázek 3. Postup u klasické aplikace versus systém .NET

(6)

CLR se nachází nad operačním systémem a poskytuje virtuální prostředí pro hostování řízených aplikací. Když se otevře řízený vykonatelný soubor, CLR nahraje modul obsahující daný spustitelný soubor a vykoná v něm obsažený kód. Kód zacílený na CLR se označuje za řízený (managed) kód a skládá se z instrukcí zapsaných v pseudo-strojovém kódu označeném za *společný zprostředkovací jazyk*¹³. Instrukce CIL se na požádání¹⁴ zkompilují do nativního strojového kódu za běhu. Ve většině případů se určitá metoda zkompiluje technikou JIT jen jednou (při svém prvním volání) a následně se uloží do paměti, aby ji příště bylo možné vykonat bez zpoždění. Kód, který se nikdy nezavolá, se ani nezkompiluje. Třebaže kompilace JIT nepochybně snižuje výkonnost, její negativní efekty jsou minimalizovány skutečností, že daná metoda se kompiluje jen jednou během celého života aplikace.

Výhod provozování kódu v řízeném prostředí CLR je velmi mnoho. Jednou z takových výhod je, že když kompilátor JIT převádí instrukce CIL na nativní kód, používá

¹³ CIL – Common Intermediate Language

¹⁴ JIT – just-in-time

proces ověřování kódu zajišťující typovou bezpečnost daného kódu. Je téměř nemožné vykonat nějakou instrukci, která přistupuje k paměti, k níž nemá oprávnění. V řízené aplikaci nikdy nenastane problém se ztracenými ukazateli, protože CLR před použitím takového ukazatele vyvolá výjimku. Typ nelze převést na něco, čím není, protože taková operace není typově bezpečná. Nemůže se ani zavolat metoda se špatně zformovaným rámcem zásobníku, protože CLR to prostě nedovolí.

Další významná výhoda je, že prostředky alokované řízeným kódem se samočinně uvolňují z paměti. Programátor si sice paměť alokuje, ale už ji neuvolňuje, za něho to dělá systém. CLR zahrnuje inteligentní nástroj uvolňování paměti, který sleduje odkazy na objekty vytvářené programátorem kódem a tyto objekty ničí, když je jimi obsazená paměť zapotřebí jinde. Tento nástroj se nazývá Garbage Collection.

Díky tomuto nástroji nezpůsobují aplikace, jež se skládají výhradně z řízeného kódu, paměťové úniky. Uvolňování paměti dokonce zvyšuje výkonnost, protože algoritmus alokování paměti používaný v CLR je mnohem rychlejší, než odpovídající rutiny alokování paměti v runtimeovém modulu jazyka C. Nevýhodou je, že když dochází k uvolňování paměti, vše ostatní v daném procesu se na okamžik zastaví. Naštěstí však k uvolňování paměti dochází relativně zřídka, čímž se výrazně snižuje jeho vliv na výkonnost.

(7)

3.1.2 Knihovna tříd rámce .NET – FCL

Knihovna tříd rámce .NET je úplně novým rozhraním API s více než 7000 typy – třídami, strukturami, rozhraními, výčty a delegáty¹⁵, které jsou integrální součástí .NET Frameworku. Některé třídy FCL obsahují více než 100 metod, vlastností a dalších členů.

Aby bylo používání FCL zvládnutelnější, byla knihovna tříd rozdělena do hierarchických jmenných prostorů (jejich počet se pohybuje okolo 100). Následující tabulka uvádí několik jmenných prostorů FCL a krátce popisuje jejich obsah. Termín „et al“ představuje následníky jmenných prostorů. Například *System.Data* et al znamená *System.Data*, *System.Data.Common*, *System.Data.OleDb*, *System.Data.SqlClient* a *System.Data.SqlTypes*.

¹⁵ Typově bezpečná obálka kolem funkcí zpětného volání.

Obor názvů	Obsah
<i>System</i>	Základní datové typy a pomocné třídy
<i>System.Collections</i>	Hešovací tabulky, pole s proměnnou velikostí a další kontejnery
<i>System.Data</i> et al	Třídy ADO.NET přístupu k datům
<i>System.Drawing</i>	Třídy pro generování grafického výstupu (GUI+)
<i>System.IO</i>	Třídy pro vykonávání operací I/O s proudy a soubory
<i>System.Net</i>	Třídy obalující síťové protokoly jako HTTP
<i>System.Reflection</i> et al	Třídy pro čtení a zápis metadat
<i>System.Runtime.Remoting</i>	Třídy pro vytváření distribuovaných aplikací
<i>System.ServiceProcess</i>	Třída pro vytváření služeb Windows
<i>System.Threading</i>	Třídy pro vytváření a spravování vláken
<i>System.Web</i>	Třídy podpory HTTP
<i>System.Web.Services</i>	Třídy pro vytváření webových služeb
<i>System.Web.Services.Protocols</i>	Třídy pro vytváření klientů webových služeb
<i>System.Web.UI</i>	Základní třídy používané v ASP.NET
<i>System.Web.UI.WebControls</i>	Serverové ovládací prvky ASP.NET
<i>System.Windows.Forms</i>	Třídy pro aplikace GUI
<i>System.XML</i> et al	Třídy pro čtení a zápis dat XML

Tabulka 1. Některé ze jmenných prostorů FCL

Fyzicky se FCL nachází v sadě knihoven DLL v adresáři `\\%SystemRoot%\Microsoft.NET\Framework\v.1.0.nnnn`. Každá knihovna DLL je kompletem, který lze na požadavek CLR nahrát.

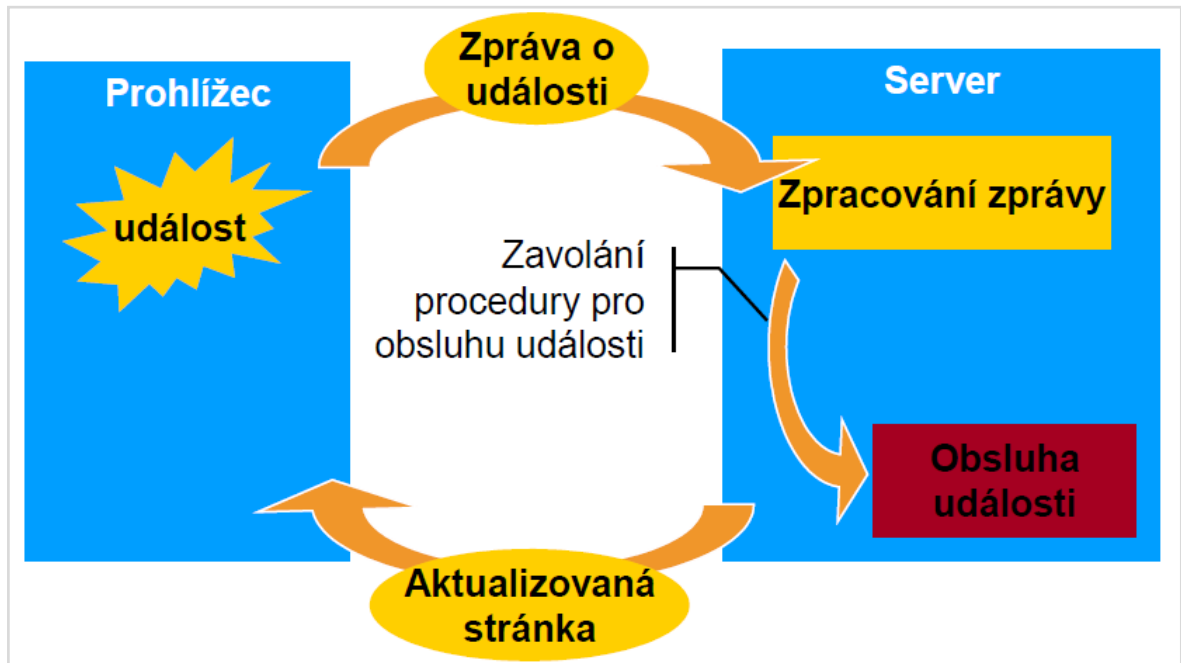
(7)

3.2 Webové aplikace ASP.NET

Internet pracuje v modelu **klient/server**. Při provádění určitého úkolu tak spolupracují dva počítače (většinou server a klientský počítač), které si vzájemně zasílají různé potřebné informace. Tato výměna informací implementována modelem **žádost/odpověď** (request/response).

Pro komunikaci mezi servery a klienty existuje však ještě jeden model, známý jako **událostmi řízený model**. Server při něm čeká na vznik určité události u klienta. Po vzniku události převezme server aktivitu do svých rukou a vykoná určité funkce. Webový server

tedy reaguje na operace v klientském počítači, jako je například kliknutí na obrázek, zapsání textu do formuláře webové stránky. ASP.NET pracuje přesně takovým způsobem – detekuje operace (akce) a posléze na ně reaguje.



Obrázek 4. Vztah mezi klientem a serverem s ASP.NET stránkou

Vzhledem k tomu, že klient může se serverem komunikovat pouze metodou žádost/odpověď, nastává problém, jak se server dozví o určité události na straně klienta. Tento problém je vyřešen pomocí skriptů.

Pokud se na straně klienta děje nějaká akce, pak takzvaný „klientský skript“ tuto akci sleduje a po jejím ukončení zašle zprávu na server, kde nastane reakce na tuto akci. Jakmile se na severu provede zpracování zprávy a obsluha příslušné události, pak server zašle aktualizovanou stránku zpět klientskému prohlížeči. Kompilátor ASP.NET tedy zdrojový kód stránky obsluhující událost přeložil do MSIL, které posléze CLR přeloží do strojového kódu, který je v tomto případě čistý kód HTML. Pro psaní funkčnosti stránek v ASP.NET tedy můžeme využít jakýkoliv podporovaný programovací jazyk (C#, J#, Visual Basic .NET, atd.), a kompilátor takový kód přeloží do HTML (přesněji skriptové) podoby.

(7)

3.3 Vzhled aplikace

3.3.1 HTML (Hyper Text Markup Language)

HTML je značkovací jazyk pro hypertext. Patří mezi jazyky pro vytváření webových stránek. Pro jeho přenos po počítačové síti byl navržen protokol HTTP¹⁶. HTML existuje ve více verzích a je charakterizován množinou značek (tagů) a jejich atributů definovaných pro danou verzi. Mezi tagy se uzavírají části textu dokumentu a tím se určuje význam obsaženého textu. Názvy jednotlivých tagů se uzavírá mezi úhlové závorky (< a >). Tagy existují párové a nepárové (ve specifikaci XHTML jsou párové všechny). Část dokumentu tvořená otevírací značkou, obsahem a odpovídajícím ukončovacím tagem se nazývá element. Atributy jsou doplňující informace, které upřesňují vlastnosti elementu. (1)

```
<body>
<div id="obal">
<form name="formular">
<table border="0">
  <tr>
    <td><p class="text">Jméno:</p></td>
    <td><INPUT name="jmeno" type="text" onblur="fce_jmeno()"></td>
  </tr>

  <tr>
    <td><p class="text">Příjmení:</p></td>
    <td><INPUT name="prijmeni" type="text" onblur="fce_prijmeni()"></td>
  </tr>

  <tr>
    <td><p class="text">e@mail:</p></td>
    <td><INPUT name="email" type="text" onblur="fce_email()"></td>
  </tr>

  <tr>
    <td><p class="text">Tel. číslo:</p></td>
    <td><span class="text" id="tel">+420</span>
    <td><INPUT name="telefon" type="text" onblur="fce_telefon()"></td>
  </tr>

  <tr>
    <td><p class="text">Rodné číslo:</p></td>
    <td><INPUT name="rodc" type="text" onblur="fce_rodc()"></td>
  </tr>

  <tr>
    <td><p class="text">Ulice:</p></td>
    <td><INPUT name="ulice" type="text" onblur="fce_ulice()"></td>
  </tr>

  <tr>
    <td><p class="text">Číslo domu:</p></td>
    <td><INPUT name="cdomu" type="text" onblur="fce_cdomu()"></td>
  </tr>
</table>
</div>
</body>
```

Obrázek 5. Ukázka HTML kódu

¹⁶ HyperText Transfer Protocol – přenosový protokol hypertextu

3.3.2 CSS (Cascading Style Sheets)

CSS, česky nazvané kaskádové styly, kaskádové proto, že na sebe mohou vrstvit definice stylu. Je kolekce metod pro grafickou úpravu webových stránek. Použití CSS má spoustu možností, například nastavit libovolnou a přesnou velikost písma, nastavit odsazení prvního řádku odstavce, zvýrazňovat odkazy po přejetí myši, umístit nějaký objekt kamkoliv do stránky.

Jsou tři možnosti, jak použít CSS. První způsob je **přímým zápisem**, kdy se v určitém elementu použije atribut *style*. Změna, provedená tímto způsobem, se projeví pouze u daného elementu. Druhý způsob, jak použít CSS styly, je využití takzvaného **stylopisu**. Do hlavičky daného HTML dokumentu se napíše stylopis uzavřený mezi tagy `<style></style>`. Takto napsaný stylopis se projeví v celém dokumentu na všechny dané tagy, ale je možné pomocí tříd a identifikátorů zařídit, aby se změny neprojeví všude. Poslední možností je vytvořit CSS styly v externím souboru. Vytvoří se soubor s příponou `.css` a do něho se zapíšou požadované styly. Pak do hlavičky HTML dokumentu, který chceme stylem ovlivnit, se zapíše odkaz na tento soubor pomocí tagu *link*.

(8)

3.3.3 JavaScript

JavaScript je programovací jazyk, který je interpretován na straně klienta. To znamená, že spuštění javascriptového kódu dochází na počítači klienta, kde jsou webové stránky načteny a ne na severu, kde jsou stránky uloženy.

Jedno z hlavních omezení JavaScriptu souvisí z prací se soubory. JavaScript nedokáže uchovávat, měnit nebo vymazávat soubory nacházející se na webovém serveru. Z pochopitelných bezpečnostních důvodů tvůrci JavaScriptu nezpracovaly funkce, které by mohli manipulovat se soubory na klientském počítači i když by to teoreticky zvládnout mohl. Díky JavaScriptu lze do webové stránky zabudovat grafické efekty, prvky, které zjednodušují navigaci mezi stránkami. Dokáže vytvářet stránky, které řeší složité matematické úkoly.

Skripty se na webu nikdy nevyskytují samostatně, jejich funkce je vždy vázaná na určitou stránku HTML a to má vliv na jeho umístění. Skripty se dají umístit buď do vnitřku HTML stránek, nebo do externích souborů s příponou `.js` s tím, že se v příslušných stránkách HTML na ně odvoláme. Do skriptu umístěného externě pak používáme syntaxi

stejnou jako u skriptu umístěného vnitřně. Umístění skriptů hraje v některých případech důležitou roli.

```
<html>
<head>
  <title>Titulek stránky</title>
</head>
<body>
  <script type="text/javascript" language="javascript">
    <!--
      document.write("Text vypsaný JavaScriptem");
    // -->
  </script>
</body>
</html>
```

Obrázek 6. Skript umístěného uvnitř HTML

```
<html>
<head>
  <title>Titulek stránky</title>
</head>
<body>
  <script src="vypis.js" type="text/javascript" language="javascript">
  </script>
</body>
</html>
```

Obrázek 7. Skript umístěný externě

(9)

3.4 Databáze

Databáze je určitá uspořádaná množina informací (dat), která slouží pro popis reálného světa a je uložena na paměťovém médiu. Toto médium může mít papírovou formu (kartotéka) nebo počítačovou formu (databázový soubor). V širším smyslu jsou součástí databáze i softwarové prostředky, které umožňují manipulaci s uloženými daty a přístup k nim. Takový software se v české odborné literatuře nazývá **systém řízení báze dat** (SŘBD). (1)

Na osobních počítačích se dnes provozují převážně SŘBD s relační architekturou. Mezi takové systémy patří například MS SQL Server, Oracle, Informix. Relační model dat má dva principy. První princip je, že data mají pravidelnou strukturu a ukládají se v tabulkách. Druhý princip je, že vztahy mezi daty se realizují pomocí operací relační algebry.

Návrh struktury databáze se provádí pomocí normalizace, což je proces dekompozice dat na jednotlivé tabulky a určení vztahu mezi nimi. Principy a pravidla vedoucí k dobře navrženému datovému modelu se nazývají Normální formy, které mají čtyři úrovně (1NF – 4NF)¹⁷.

Následující pravidla určují, jak správně navrhnout databázový model:

- 1) **Flexibilita** – je základním stavebním kamenem tvorby databáze. Je nutné vytvořit takovou databázi, která bude vyžadovat při nepatrné změně požadavků na funkci systému i nepatrné změny vlastní realizace.
- 2) **Stabilita systému z hlediska vývoje systému** – malý zásah do nestabilního systému si vynutí dodatečné velké úpravy (změní-li se jedna tabulka, pak je nutné předělat desítky dalších dotazů), aby se systém vrátil do funkčního stavu. Tomuto problému se dá předejít objektovým přístupem.
- 3) **Každé pole v tabulce by mělo mít svůj jednoznačný význam** – neměl by se měnit v závislosti na jiných hodnotách (například hodnota v předchozím sloupci).
- 4) **Nemít všechny informace v jedné tabulce** – je dobré vytvořit více tabulek, které budou provázány přes ID (jedinečné identifikační číslo). V této fázi návrhu je nutné dbát na flexibilitu navrhované databáze.

(10)

3.4.1 Microsoft SQL Server 2008

Zatím poslední verzí databázového serveru od Microsoftu je SQL Server 2008. Oproti předchozím verzím má SQL Server 2008 několik vylepšení¹⁸. Následující seznam uvádí přehled některých změn:

- 1) O šifrování a dešifrování dat se stará engine. Ten udržuje nešifrovanou podobu dat v paměti a data na pevném disku šifruje. Již není třeba změn v aplikaci pro použití šifrovacích funkcí.
- 2) Lze vytvářet šifrované zálohy databází, aby nedošlo k nežádoucí manipulaci. Možnost nastavení počtu povoleného obnovení z dané zálohy.

¹⁷ Více informací v citované literatuře.

¹⁸ Bližší informace o produktu naleznete [zde](#).

- 3) Správu šifrovacích klíčů lze provádět nově i externě nástroji třetích stran, ukládání klíče na speciální HW.
- 4) Komprese databáze, komprese zálohy databáze.
- 5) Nástroj Resource Governor – spravuje prostředky serveru (CPU, timeout, execution time), které přiděluje jednotlivým uživatelským skupinám a procesům v SQL serveru. V případě překročení limitu lze spustit automatizovanou akci.
- 6) Možnost přidání nového procesoru za běhu systému a využití jeho kapacity bez restartu.
- 7) Již nepoužívané zdroje SQL Serveru jsou uvolňovány rychleji, což zvýšilo výkon dotazovacích operací.
- 8) Implementován auditovací systém pro sledování velkého množství akcí (příkaz AUDIT).
- 9) Vylepšen systém instalace, aplikace záplat a servicepacků.
- 10) Zjednodušení procesů zprávy databází (již nejsou nutné hluboké znalosti technologie, vysoký podíl automatizace, škálovatelnost mnoha serverů pod jednotnou správou).
- 11) Uživatelem definované datové typy již nejsou velikostně omezeny (doposud 8kb na stránku).
- 12) Nové datové typy pro datum a čas – Date, Time, DateTime Offset (údaj s časovou zónou), DateTime2 (vyšší rozsah hodnot).
- 13) Datový typ File Stream, umožňuje uložit nestrukturovaná data přímo ze souborového systému. Možnost uložení velkého objemu binárních dat, která lze dotazovat a modifikovat jazykem SQL.
- 14) Nové datové typy (Location a Geometry) obsahují data se vztahem ke geografické lokaci (územní souřadnice) pro systémy na bázi GPS.

(11)

II. PRAKTICKÁ ČÁST

4 NÁVRH SYSTÉMU

Primárním cílem je navrhnout jádro systému, které tvoří komunikace mezi pacientem a lékařem prostřednictvím systému elektronické ordinace. Celý systém je rozdělen ze dvou hledisek a to z pohledu uživatele, respektive jeho uživatelských práv a z hlediska jejich komunikace.

4.1 Uživatelé systému

Systém obsahuje jednu databázi. V databázi jsou uloženy potřebné informace o uživateli systému. Databáze je uložena na samostatném SQL serveru. Databázi tvoří dvanáct tabulek, které jsou vzájemně propojeny. Níže uvedené tabulky představují strukturu a návrh tabulek v databázi. Není-li napsané jinak, musí být sloupce tabulek vyplněny.

Uživatelé		
<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
User_ID	int	Primární klíč, slouží k jednoznačné identifikaci uživatele.
Jmeno	varchar(30)	Jméno uživatele. (i s titulem před jménem)
Prijmeni	varchar(50)	Příjmení uživatele. (i s titulem za jménem)
PrihlasUdaje	int	Sekundární klíč, ukazatel do tabulky PrihlasovaciUdaje.
InfoPacient	int	Sekundární klíč, ukazatel do tabulky InfoPacient, nemusí být vyplněno.
InfoLekar	int	Sekundární klíč, ukazatel do tabulky InfoLekar, nemusí být vyplněno.

Tabulka 2. Uživatelé

PrihlasovacíUdaje

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
PrihlasovacíUdaje_ID	int	Primární klíč, slouží k jednoznačné identifikaci přihlašovacích údajů.
ID	varchar(9)	Jednoznačný identifikační uživatelský údaj. (číslo)
Heslo	varchar(15)	Uživatelské heslo. (maximálně 15 znaků)
UzivRole	int	Sekundární klíč, ukazatel do tabulky UzivRole.

Tabulka 3. Přihlašovací údaje

UzivRole

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
Role_ID	int	Primární klíč, slouží k jednoznačné identifikaci uživatelské role.
TypUzivatele	varchar(10)	Název Uživatelské role. (Admin, Lekar, Pacient)

Tabulka 4. Uživatelské role

InfoPacient

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
InfoPacient_ID	int	Primární klíč, slouží k jednoznačné identifikaci uložených informací o pacientovi.
RodneCislo	varchar(11)	Rodné číslo.
DatumNarozeni	varchar(20)	Datum narození.
Telefon	varchar(15)	Telefonní číslo. (např.: +420608001001)
Email	varchar(30)	Email, nemusí být zadáno.
KrevniSkupina	varchar(3)	Krevní skupina. (např.: AB+)
PraktickyLekar	int	Sekundární klíč, ukazatel do tabulky Uzivatele.
Adresa	int	Sekundární klíč, ukazatel do tabulky Adresa.
Pojistovna	int	Sekundární klíč, ukazatel do tabulky Pojistovna.

Tabulka 5. Informace o Pacientovi

Adresa

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
Adresa_ID	int	Primární klíč, slouží k jednoznačné identifikaci uložených adres.
Stat	varchar(50)	Stát.
Město	varchar(50)	Město.
Ulice	varchar(50)	Ulice.
CisloPopisne	varchar(10)	Číslo popisné.
SmerovaciCislo	varchar(10)	Směrovací číslo.

Tabulka 6. Adresa

Pojistovna

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
Kod	int	Primární klíč, slouží k jednoznačné identifikaci pojišťovny.
Nazev	varchar(50)	Název pojišťovny.

Tabulka 7. Pojišťovna

InfoLekar

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
InfoLekar_ID	int	Primární klíč, slouží k jednoznačné identifikaci uložených informací o lékaři.
RodneCislo	varchar(11)	Rodné číslo.
DatumNarozeni	varchar(20)	Datum narození
Odbornost	varchar(50)	Slouží k identifikaci odbornosti lékaře.
Nemocnice	varchar(50)	Název nemocnice.
AdresaNemocnice	int	Sekundární klíč, ukazatel do tabulky Adresa.

Tabulka 8. Informace o lékaři

ZdravotniKarta

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
Datum	varchar(20)	Primární klíč, slouží k jednoznačné identifikaci zdravotního záznamu.
Zaznam	text	Zde lékař vloží svůj lékařský popis diagnózy.
Pacient	int	Sekundární klíč, ukazatel do tabulky Uzivatele.
LekarPodpis	int	Sekundární klíč, ukazatel do tabulky Uzivatele.

Tabulka 9. Zdravotní karta

OckovaciKarta

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
Datum	varchar(20)	Primární klíč, slouží k jednoznačné identifikaci očkovacího záznamu.
Status	varchar(10)	Slouží k identifikaci plánovaných očkování.
Pacient	int	Sekundární klíč, ukazatel do tabulky Uzivatele.
LekarPodpis	int	Sekundární klíč, ukazatel do tabulky Uzivatele.
TypOckovani	int	Sekundární klíč, ukazatel do tabulky Ockovani.

Tabulka 10. Očkovací karta

Ockovani

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
Ockovani_ID	int	Primární klíč, slouží k jednoznačné identifikaci typu očkování.
Typ	varchar(30)	Název typu očkování.

Tabulka 11. Typ očkování

Rozvrh

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
Datum	varchar(20)	Primární klíč, slouží k jednoznačné identifikaci pojišťovny.
Status	varchar(10)	Slouží k identifikaci. (je-li možné se na termín přihlásit)
TypVysetreni	int	Nemusí být vyplněn. Sekundární klíč, ukazatel do tabulky TypVysetreni.
Pacient	int	Nemusí být vyplněn. Sekundární klíč, ukazatel do tabulky Uzivatele.
Lekar	int	Sekundární klíč, ukazatel do tabulky Uzivatele.

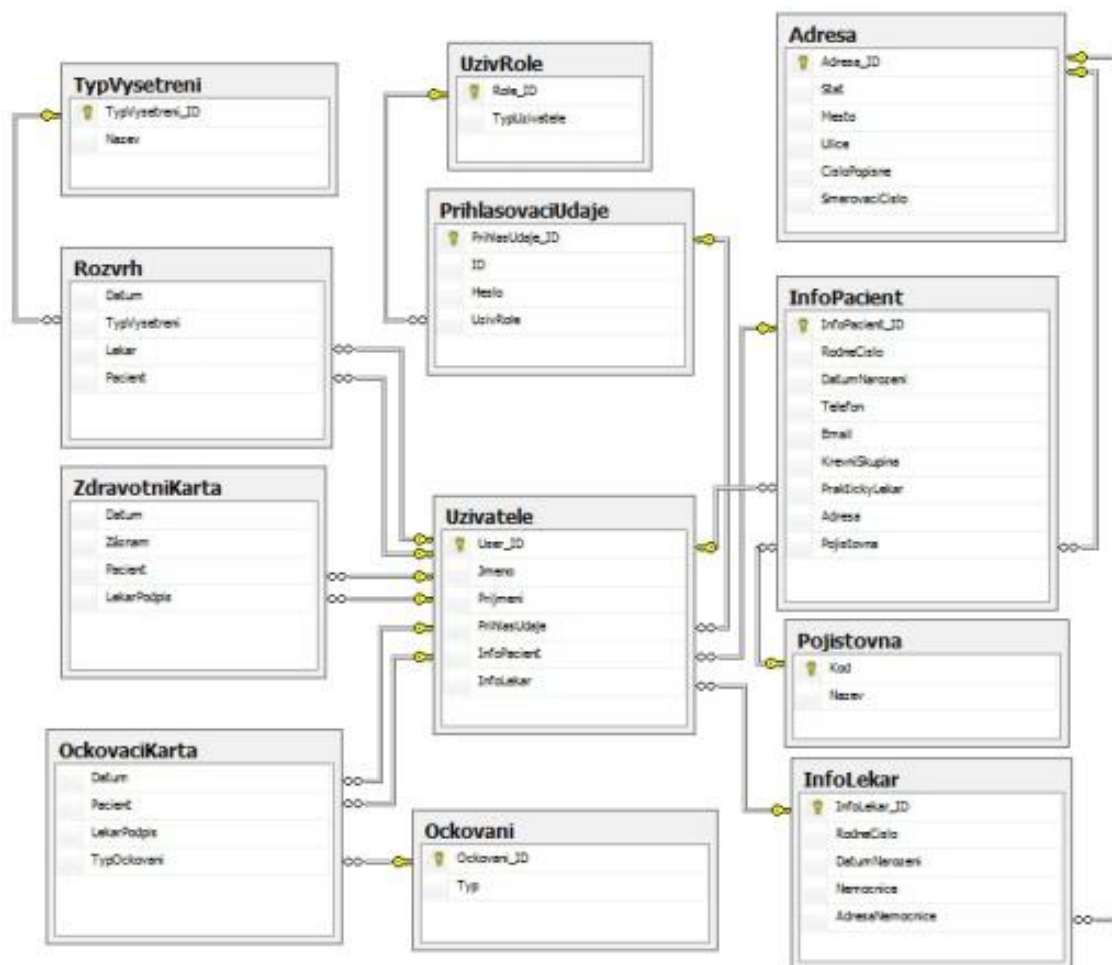
Tabulka 12. Rozvrh

TypVysetreni

<i>Sloupec</i>	<i>Datový typ</i>	<i>Popis</i>
TypVysetreni_ID	int	Primární klíč, slouží k jednoznačné identifikaci typu vyšetření.
Nazev	varchar(50)	Název typu vyšetření. (např.: očkování, preventivní prohlídka, ...)

Tabulka 13. Typ vyšetření

Relace mezi tabulkami zobrazuje následující obrázek.



Obrázek 8. Relační diagram

Každému uživateli, který se přihlásí do systému, se zobrazí pouze relevantní informace. Přihlášení do systému proběhne přes úvodní stránku celého systému.

Vzhledem k tomu, že systém by měl být co nejobecnější a měl by sloužit pro všechny lidi v každém věku, pak je nutné vyřešit problém s jejich identifikací. Dospělé a dospívající lidi lze například identifikovat pomocí občanského průkazu. Děti však žádné doklady mít nemusí. Předpokládejme tedy, že se každý člověk bude identifikovat pomocí biometrických prvků (otisk prstu, dlaně, oční sítnice, atd.). Informace o těchto biometrických prvcích budou uloženy v centrální databázi a dospělým, či dospívajícím lidem budou vystaveny doklady, obsahující dané informace. Každý záznam v databázi bude mít jednoznačný kód, a ten bude sloužit také pro identifikaci v systému elektronické ordinace. Není vhodné zde spekulovat o podobě identifikačního kódu, ale předpokládejme, že ponecháme stávající identifikační systém u občanských průkazů, tedy devítimístný, číselný kód.

Uživatel se tedy přihlašuje do systému pomocí devítimístného číselného kódu a hesla. Je zde i možnost zapamatování uživatele na aktuálním počítači. Design aplikace má klasické rozvržení. Aplikace se skládá ze čtyř částí, panel pro nadpis, levý panel pro navigační menu, pravý panel pro zobrazení kalendáře či eventuelních doprovodných informací a nakonec prostřední panel, ve kterém se zobrazuje jádro stránky. Navigační panel zobrazuje odkazy na stránky v závislosti na přihlášeném uživateli.

4.1.1 Lékař

Lékař, který se přihlásí do systému, má největší možnosti. Zobrazí se mu následující záložky:

- **Nový pacient** – jako jediný uživatel, může lékař přidat nového pacienta. Kromě osobních údajů, bude muset doplnit i lékařskou dokumentaci. To se provádí na stránce „Karta pacienta“.
- **Výběr pacienta** – karta, kde se vybere pacient, který přišel do ordinace. Po výběru pacienta se zobrazí jeho identifikační údaje jako adresa, datum narození, atd. v záložce „Základní informace“.
- **Karta pacienta** – zde se zobrazí rolovací seznam, který bude obsahovat všechny pacientovy lékařské záznamy, které jsou identifikovány datem záznamu a lékařem, který tento záznam vypsál. Po vybrání záznamu se zobrazí informace v něm uložené. Je zde možnost editace a přidání nového záznamu.
- **Očkovací průkaz** – zde jsou informace o předchozích očkovacích termínech pacienta. Je zde i možnost přidání nového záznamu, což bude také sloužit jako objednání k lékaři na očkování.
- **Objednání¹⁹** – zde může lékař objednat pacienta jakémukoli lékaři, například na vyšetření nebo na kontrolu.
- **Vypsání receptu** – karta, která obsahuje formulář pro objednání léku v lékárně. Po odeslání receptu si bude pacient moci vyzvednout léky v lékárně. Lékař smí recept vytisknout pro případ, že by lékárna měla potíže s připojením k systému.

¹⁹ Více informací o způsobu objednání viz kapitola 4.2.

- **Základní informace** – zde se zobrazí informace o vybraném pacientovi. Pokud není pacient vybrán, pak zde žádné informace nebudou zobrazeny. Lékař má možnost editace.

4.1.2 Pacient

Po přihlášení pacienta se opět zobrazí jen relevantní záložky. Některé jsou podobné jako u lékaře, ale na rozdíl od lékaře má pacient omezený přístup k některým funkcím.

- **Základní informace** – karta s obecnými informacemi přihlášeného pacienta. Pacient má možnost některé z těchto informací editovat.
- **Karta pacienta** – pacient si zde může prohlédnout své lékařské záznamy samozřejmě bez možnosti editace.
- **Očkovací průkaz** – podobně jako karta pacienta. I zde si pacient může prohlédnout záznamy bez možnosti editace.
- **Objednání** – v této kartě má pacient podobné možnosti, jako lékař. Může se tedy k lékaři objednat sám. Nelze se ovšem objednat na očkování.
- **Přehled lékařů** – karta sloužící pouze k prohlédnutí některých informací o lékařích uložených v systému. Hlavním důvodem zařazení této karty je, aby si pacient mohl zjistit identifikační číslo lékaře pro případné objednání mobilním telefonem.
- **Rozvrh** – zde si pacient může prohlédnout svůj rozvrh dne, který je vybrán pomocí kalendáře. Stránka má pouze informativní charakter.

4.1.3 Administrátor

Osoba, která se přihlásí pod tímto účtem, má jen dvě možnosti. Přístup k tomuto účtu by měl mít člověk, který spravuje zařízení, kde lékaři slouží. Je patrné, že by to mohl být ředitel nemocnice, primář nebo konkrétní lékař v soukromé ordinaci.

- **Přidat lékaře** – zde je formulář pro přidání lékaře do databáze lékařů. K dispozici má administrátor vyplnění osobních údajů o lékaři, odborné způsobilosti atd.
- **Lékaři** – karta, ve které se po vybrání lékaře zobrazí informace o lékaři včetně identifikačního čísla a hesla. Je zde i možnost editace.

4.2 Komunikace

Objednávání a upozornění na očkování pomocí SMS je hlavní myšlenkou komunikace mezi lékařem a pacientem. Předpokládejme, že každý člověk má přístup k internetu a k mobilnímu telefonu.

4.2.1 Objednání

Objednávat lze dvěma způsoby. První způsob je přes webový formulář a druhý způsob je přes SMS. Ať už se pacient objedná jakýmkoli způsobem, tak systém automaticky kontroluje objednané pacienty a zasílá jim informativní SMS o blížícím se termínu návštěvy u lékaře. Týden před datem plánované návštěvy se pacientovi pošle SMS informující o tom, že se blíží termín návštěvy a že se lze z tohoto termínu přeobjednat na jiný. Druhá, a také poslední upomínka se pacientovi pošle den před návštěvou. Ta informuje pacienta o datu i času plánované návštěvy a upozorňuje, že se již nelze odhlásit.

Objednání pomocí webového formuláře

Tento typ objednání mohou využít pacienti i lékaři. Po vybrání karty *objednání* se zobrazí seznam lékařů. Jakmile je zvolen lékař, tak se zobrazí jeho rozvrh na aktuální den. Po výběru data v kalendáři se rozvrh načte na vybraný den. Rozvrh bude ve formě tabulky zobrazovat volné a obsazené termíny. Na volný termín se může objednat pacient pomocí tlačítka *objednat*. Pro zrušení objednání slouží tlačítka *zrušit objednání*. Pacient smí objednávku zrušit nejpozději dva dny předem. Poté bude tato možnost nedostupná a pacient se nebude moci objednat na žádný jiný termín. Lékař smí objednání zrušit kdykoli. Pokud lékař zruší objednávku na aktuální či následující den, pak se pacientovi automaticky pošle informativní SMS o zrušení objednání.

Každý lékař má všechny práva upravovat svůj rozvrh. Má tedy možnost smazat záznam a přidat záznam jakéhokoliv pacienta v kterýkoliv den. V jeho vlastním rozvrhu se zobrazují jména objednaných pacientů. Pokud lékař objednává pacienta k jinému lékaři, pak se mu zobrazí cizí rozvrh. V tomto rozvrhu se zobrazí pouze jméno aktuálně vybraného pacienta, pokud je už objednán a u ostatních termínů se zobrazí pouze status *obsazeno* či *volno*. Lékař má tedy možnost pouze objednat a zrušit objednání aktuálně vybraného pacienta. Lékař smí ve svém rozvrhu zablokovat možnost objednání na jakoukoli hodinu. To znamená, že ostatní uživatelé nebudou mít možnost na tento termín nikoho objednat ani zrušit jeho objednání.

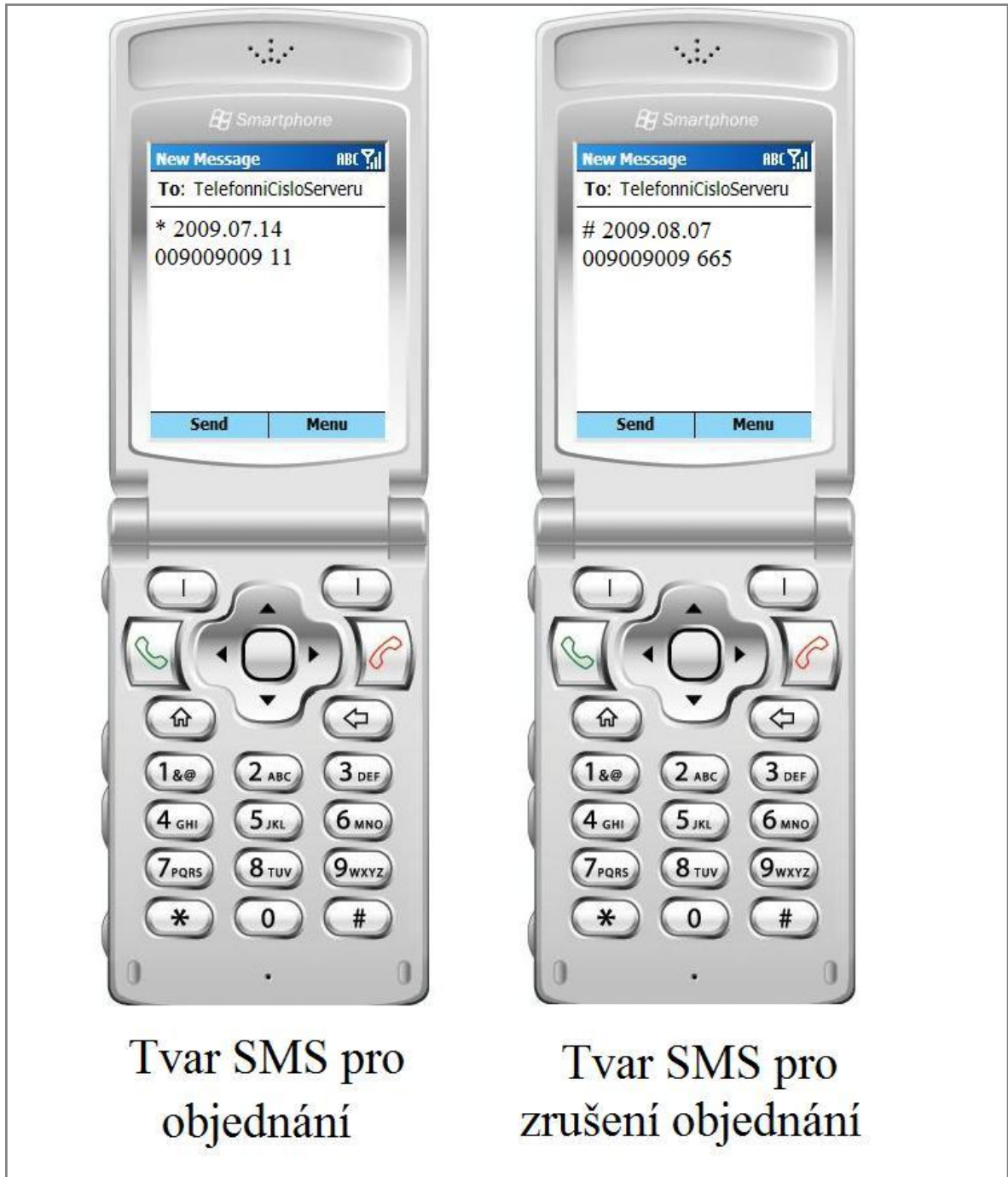
Podobné možnosti jako lékař má i pacient, který se chce objednat k lékaři. Ten si opět může vybrat lékaře, u kterého chce objednání uplatnit. Po zobrazení rozvrhu lékaře má pacient k dispozici volné termíny pro objednání. U ostatních termínů je vidět pouze status *obsazeno*.

Objednání pomocí SMS

Tento typ objednání slouží jen pro pacienty. Jeho základní omezení spočívá v tom, že by tento typ objednání neměl být drahý. Proto stačí potencionálnímu pacientovi poslat jednu SMS pro objednání a systém ho rovnou objedná. Pacient tedy pošle SMS v předepsaném formátu a systém sám objedná pacienta na nejbližší možný termín. Systém pošle SMS s datem a časem, na který je pacient objednán. Pro zrušení objednání lze opět zaslat SMS v předepsaném formátu. Zrušit objednávku lze nejpozději dva dny předem. Poté ji může odstranit pouze lékař.

Předepsaný formát je následující, všechny části jsou odděleny mezerami:

- **Klíčový znak** – může být jen * nebo #. Hvězdička značí symbol pro objednání a křížek značí symbol pro zrušení objednání.
- **Datum** – určuje požadované datum návštěvy. Píše se ve formátu „rok měsíc den“, kdy dané části jsou odděleny tečkami.
- **Kdo** – určuje pacienta, který se chce objednat. Uživatel zde napíše své identifikační číslo.
- **Ke komu** – určuje lékaře, ke kterému se chce pacient objednat. Pacient zde napíše identifikační číslo lékaře. Toto číslo **není** přihlašovací kód lékaře, ale určuje pozici lékaře v databázi. Lze ho zjistit v kartě *Přehled lékařů*.



Obrázek 9. Příklad tvarů pro SMS

4.2.2 Upozornění na očkování pomocí SMS

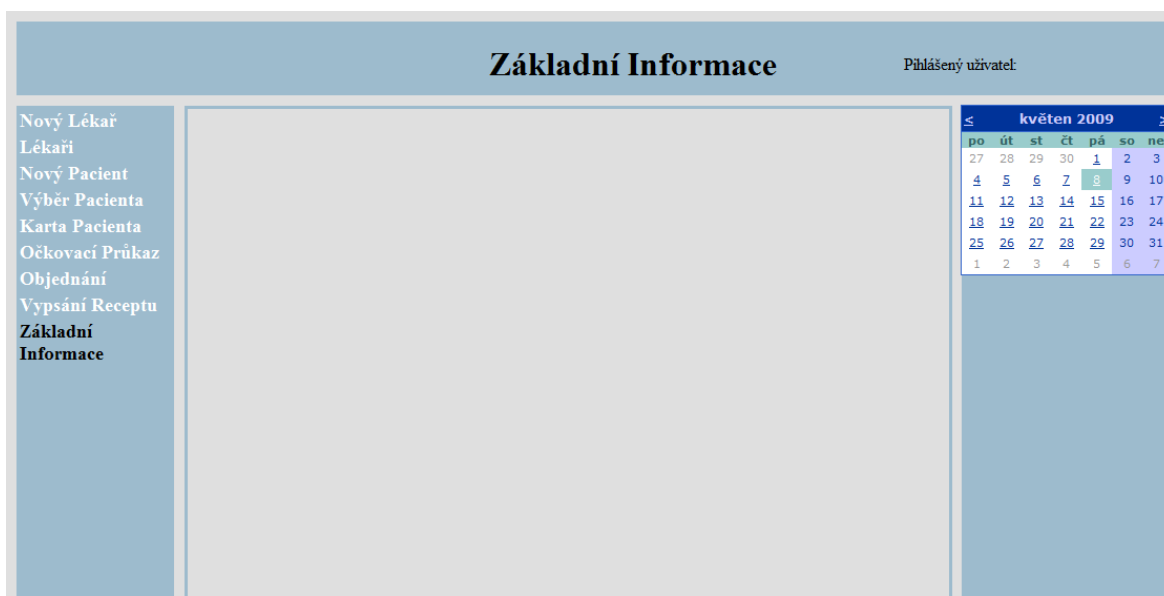
O termínu očkování rozhoduje lékař. Zadá do systému na kdy, a koho chce objednat. Po potvrzení objednávky na očkování se automaticky do databáze uloží upozornění, které se pacientovi pošle týden a den před návštěvou ve formě SMS.

5 REALIZACE SYSTÉMU

Realizace systému je rozdělena do dvou fází. První fáze je design webové aplikace, který je tvořen pro jednoduchý vzhled aplikace. Druhou fází je technické řešení jednotlivých webových stránek, kde jsou stručně popsány nejdůležitější funkce konkrétních stránek.

5.1 Design

Okno webové aplikace je rozděleno do čtyř částí. Všechny stránky mají jednotný design, a proto její základní rozvržení je naprogramováno ve stránce MasterPage.aspx, na kterou se odkazují všechny ostatní stránky. Pro rozvržení panelů jsou použity CSS styly s absolutním pozicováním.



Obrázek 10. Design a logické rozdělení aplikace

V levém panelu se nachází navigační menu. Menu tvoří hypertextové odkazy na požadované stránky, které jsou dynamicky načítány z XML souboru. Jeho strukturu tvoří dvě základní položky. V první položce je uložen název stránky, na kterou bude směřovat odkaz, a který se zobrazí v navigačním menu. Druhou položkou je samotný odkaz na požadovanou stránku. Odkazy v navigačním menu mají bílou barvu, při přejetí myši se barva odkazu změní na černou.

Navigační panel je programově vytvořená uživatelská komponenta a funguje na jednoduchém principu. Nejdříve načte XML soubor a pročítá ho do konce po jednotlivých položkách. Po načtení položky se provede kontrola totožnosti přihlášeného uživatele a na

základě jeho uživatelské role se pro každou položku se vytvoří řádek v tabulce. Do každého řádku takto vytvořené tabulky se vloží hypertextový odkaz na adresu požadované webové stránky.

Kalendář, který je umístěn v pravém panelu, vytváří při vybrání libovolného data prvek cookie s názvem „*VybraneDatum*“. Je v něm uložena jedna hodnota, která identifikuje právě vybrané datum (příklad: Datum=2009.05.17). S tímto datem se dále pracuje v některých stránkách. Je možné vybrat pouze pracovní dny v týdnu.

5.2 Technické řešení jednotlivých webových stránek

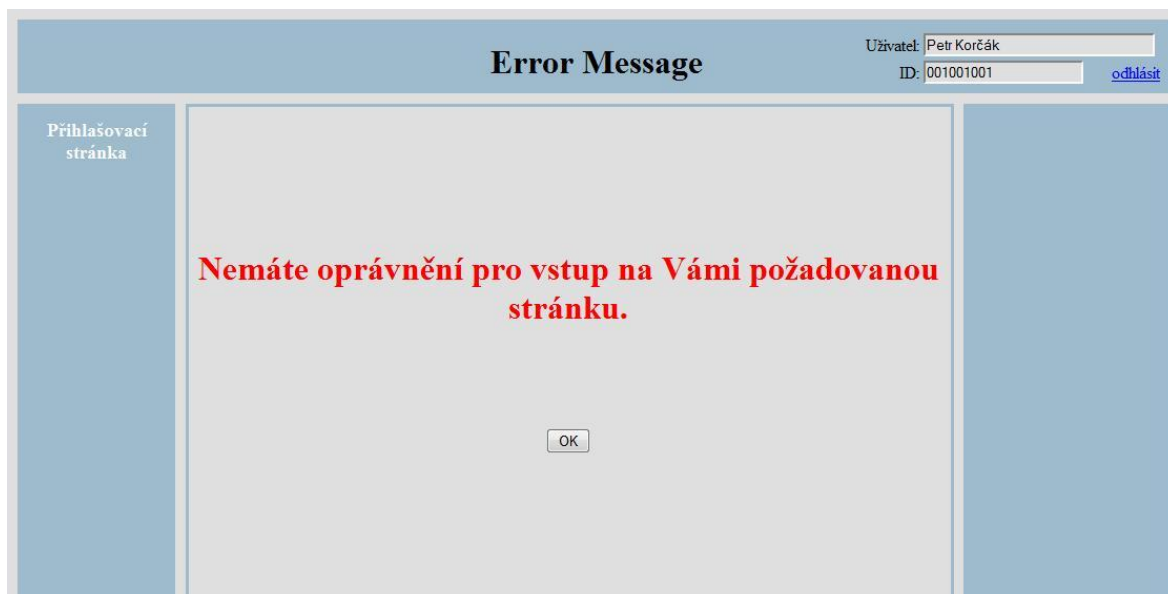
Přihlášení do systému se provádí přes přihlašovací stránku „*Default.aspx*“. Není použit žádný předdefinovaný poskytovatel či komponenta. To má vliv na vlastní autorizaci a autentizaci v systému.

The image shows a login form titled "Přihlášení" centered on a dark blue background. The form has a light gray background and contains the following elements: a label "ID:" followed by a text input field; a label "Heslo:" followed by a text input field; a checkbox labeled "Pamatovat si na tomto počítači?"; and a button labeled "Přihlásit" at the bottom right.

Obrázek 11. Přihlašovací stránka

Po úspěšném přihlášení do systému se vytvoří prvek cookie s názvem „*PrihlasovaciUdaje*“. Data v prvku cookie tvoří přihlašovací identifikační číslo a typ přihlášeného uživatele (příklad: ID=009009009&Role=Lekar). Pro přenos prvku cookie je použit zabezpečený protokol HTTPS. Po vytvoření prvku cookie je uživatel přesměrován na defaultní webovou stránku jeho uživatelské role.

Na každé stránce je tedy vytvořen mechanismus, který kontroluje roly přihlášeného uživatele a určuje, zdali má přihlášený uživatel práva pro zobrazení požadované stránky. Jestliže se uživatel snaží přejít na stránku, pro kterou nemá přiřazena uživatelská práva, pak ho aplikace přesměruje na chybovou stránku „*ErrorMessage.aspx*“.



Obrázek 12. Chybová stránka

V pravém rohu chybové stránky jsou identifikační údaje přihlášeného uživatele. V navigačním menu je jediný odkaz, který přesměruje uživatele na přihlašovací stránku a dává tak přihlášenému uživateli možnost, přihlásit se pod jiným účtem. Po přesměrování na přihlašovací stránku se prvek cookie nezničí, proto je možné přejít na jakoukoli autorizovanou stránku zadáním její adresy do adresní lišty prohlížeče. Aby nebyl tento způsob příliš komplikovaný, je pod chybovou hláškou umístěno tlačítko „OK“. Kliknutím na tlačítko aplikace přesměruje přihlášeného uživatele na defaultní stránku jeho uživatelské role.

Všechny informace na stránkách aplikace jsou úzce propojené s daty v databázi, a proto hlavní náplní téměř všech obslužných rutin tvoří databázové dotazy. Přesněji řečeno, většinou je potřeba pro správnou funkci stránek několik sekvencí databázových dotazů. Sekvence dotazů pro každou potřebnou akci jsou uloženy na databázovém serveru ve formě procedur. Ty jsou aplikací v potřebnou chvíli volány a jejich výsledky zpracovány podle potřeby. Jednoduché databázové dotazy jsou aplikací volány přímo.

Stránky, které požadují uživatelské vstupy, jsou důkladně kontrolovány na správnost zadaných dat. Typickým příkladem je zadání rodného čísla. Jeho formát je přesně daný a jeho část před lomítkem musí souhlasit s datem narození. O podobě rodného čísla rozhoduje také pohlaví jeho majitele. Ženy a muži mohou mít tedy různý formát části rodného čísla před lomítkem a i s takovým detailem aplikace počítá.

5.2.1 Nový lékař

Role: Administrátor Uživatel: Petr Korčák
ID: 001001001 [odhlásit](#)

Nový Lékař

Nesouhlasí rodné číslo a datum narození.
Nemocnice s touto adresou již v databázi existuje a bude použita k uložení.

Jméno: MUDr. Jana Jméno nemocnice: Bařova Nemocnice

Příjmení: Maliková **Adresa nemocnice**

ID: 100100100 Ze seznamu

Heslo: aaa Zobrazit znaky Stát: Česká Republika

Rodné číslo: 675604/312 Město: Zlín

Datum narození: 1967 02 01 Ulice: tř. Tomáš Bati

Rok Měsíc Den Číslo popisné: 8

Odbornost: Praktický lékař Směrovací číslo: 76501

	Stát	Město	Ulice	Číslo popisné	Směrovací číslo
<input type="button" value="Výběr"/>	Česká Republika	Zlín	tř. Tomáš Bati	8	76501
<input type="button" value="Výběr"/>	Česká Republika	Otrokovice	Nádražní	12	76502
<input type="button" value="Výběr"/>	Česká Republika	Uherské Hradiště	Husova	13	76503

Obrázek 13. Stránka „Nový Lékař“

Všechny uživatelské vstupy musí být vyplněny. Pokud se uživatel pokusí formulář uložit, pak se vedle nevyplněného textového políčka ukáže červená hvězdička. Tuto kontrolu zajišťuje komponenta nazvaná *RequiredFieldValidator*, která je vestavěná ve Visual Studiu.

Textové pole pro identifikační číslo navíc kontroluje komponenta *RegularExpressionValidator*, která zjišťuje, zdali je zadáno devítimístné číslo. Posledně zmíněná komponenta kontroluje také textové pole pro zadání rodného čísla. Regulární výraz pro rodné číslo očekává šest čísel, lomítko a na konec tři nebo čtyři číslice. *RegularExpressionValidator* ještě kontroluje emailovou adresu, telefonní číslo a směrovací číslo.

Poslední použitou komponentou, pro kontrolu uživatelského vstupu je validátor s názvem *CustomValidator*. Tato komponenta je nastavena na kontrolu textového pole pro zadání rodného čísla a hlídá programátorem definovanou úlohu. V tomto případě je kontrolováno, zdali je shodná část rodného čísla před lomítkem se zadaným datem narození.

Pro vybrání data narození jsou použity rolovací seznamy. Jakmile uživatel vybere rok a měsíc data narození, pak se zpřístupní možnost pro vybrání dne data narození. Pro

inicializaci rolovacího seznamu dne je naprogramována metoda, která zjišťuje počet dnů v měsíci daného roku. Tato metoda je volána pokaždé, kdy uživatel změní datum. Nemůže se tak stát, že by měsíc únor měl třicet jedna dní.

Zaškrťovací políčko „Ze seznamu“ dává uživateli možnost vybrat adresu již uloženou v databázi. Po zaškrtnutí políčka se zavolá databázová procedura a ve spodní části stránky se zobrazí tabulka se všemi adresami nemocnic. Pro zobrazení tabulky s adresami nemocnic je použita komponenta „GridView“. Po načtení dat z databáze se vypíší získané výsledky do příslušných textových polí.

Jakmile jsou všechny požadované uživatelské vstupy všechny vyplněny a odpovídají požadavkům validátorů, pak může uživatel kliknout na tlačítko uložit. V metodě, obsluhující akci uložení, se dále kontrolují některé informace určené k uložení a pro tuto kontrolu jsou využity databázové procedury. Jako první se kontroluje, zdali vyplněné identifikační číslo nemá již nějaký uživatel přiřazené. Přihlašovací identifikační číslo musí být jedinečné. To samé platí i pro rodné číslo. Další kontrola zjišťuje, jestli zadaná adresa již v databázi existuje a zdali nepatří nějakému pacientovy. Není možné, aby pacient měl bydliště v nemocnici nebo aby nemocnice měla adresu pacienta. Tato kontrola se provádí pouze tehdy, není-li zaškrtnuto políčko „Ze seznamu“, protože tabulka adres vykresluje jen adresy nemocnic.

5.2.2 Lékaři

Lékaři																																																																			
Role: Administrátor		Uživatel: Petr Korčák																																																																	
		ID: 001001001 odhlásit																																																																	
Nový Lékař Lékaři	<input type="button" value="Upravit"/> <input type="button" value="Odstranit"/>	MUDr. Jan	Novák, CSc	002002002	aaa	530325/3214	1953.03.25	Praktický lékař	Baťova Nemocnice	Česl Repub	<table border="1"> <thead> <tr> <th colspan="7">květen 2009</th> </tr> <tr> <th>po</th> <th>út</th> <th>st</th> <th>čt</th> <th>pá</th> <th>so</th> <th>ne</th> </tr> </thead> <tbody> <tr> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> </tr> <tr> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15</td> <td>16</td> <td>17</td> </tr> <tr> <td>18</td> <td>19</td> <td>20</td> <td>21</td> <td>22</td> <td>23</td> <td>24</td> </tr> <tr> <td>25</td> <td>26</td> <td>27</td> <td>28</td> <td>29</td> <td>30</td> <td>31</td> </tr> <tr> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> </tr> </tbody> </table>	květen 2009							po	út	st	čt	pá	so	ne	27	28	29	30	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	1	2	3	4	5	6	7
	květen 2009																																																																		
	po	út	st	čt	pá	so	ne																																																												
	27	28	29	30	1	2	3																																																												
	4	5	6	7	8	9	10																																																												
11	12	13	14	15	16	17																																																													
18	19	20	21	22	23	24																																																													
25	26	27	28	29	30	31																																																													
1	2	3	4	5	6	7																																																													
<input type="button" value="Upravit"/> <input type="button" value="Odstranit"/>	MUDr. Marie	Koubková, PhD	006006006	aaa	655622/1212	1965.06.22	Alergolog	Poliklinika Otrokovice	Česl Repub																																																										
<input type="button" value="Upravit"/> <input type="button" value="Odstranit"/>	MUDr. Martin	Drahoš	007007007	aaa	770606/3452	1977.06.06	Praktický lékař	Poliklinika Uherské Hradiště	Česl Repub																																																										
<input type="button" value="Upravit"/> <input type="button" value="Odstranit"/>	MUDr. Marie	Šafářová	011011011	aaa	755228/321	1975.02.28	Praktický lékař	Baťova Nemocnice	Česl Repub																																																										

Obrázek 14. Stránka „Lékaři“

Na této stránce se pomocí komponenty „GridView“ zobrazuje seznam lékařů. Vytvoření komponenty je jednoduché, nicméně složitost tohoto prvku spočívá v SQL dotazech pro úpravu a vymazání dat z databáze. Tyto dvě akce spouští tlačítka v prvním sloupci tabulky. Kontrola uživatelských vstupů je provedena podobným způsobem, jako na stránce „Nový Lékař“. Záznamy v tabulce se dají seřadit a to podle jakéhokoliv sloupce.

5.2.3 Nový pacient

Role: Lékař

Nový Pacient

Uživatel: MUDr. Jan Novák, CSc.
ID: 002002002 [odhlásit](#)

Neplatný formát rodného čísla.

Jméno: Roman Praktický lékař: MUDr. Marie Šafářová

Příjmení: Mach Pojišťovna: 205 - Hutnická Pojišťovna

ID: 020202020

Heslo: aaa Zobrazit znaky

Rodné číslo: 870704 Ze seznamu

Datum Narození: 1987 07 04
Rok Měsíc Den

Stát: Česká Republika

Telefon: +420 778090909 Město: Otrokovice

Email: mach.roman@yahoo.com Ulice: Jiráskova

Krevní skupina: B+ Číslo popisné: 1332

Směrovací číslo: 76502

květen 2009						
po	út	st	čt	pá	so	ne
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Obrázek 15. Stránka „Nový pacient“

Funkčnost stránky je téměř podobná funkčnosti stránky „Nový Lékař“. Jsou zde pouze přidány další formulářové prvky, jako krevní skupina, email, telefon a pojišťovna. Data, načtená do položek rolovacích seznamů pro telefonní předvolbu, krevní skupinu a státy, jsou načítána z XML souborů. Pokud je tedy třeba přidat například telefonní předvolbu, pak stačí vložit do příslušného XML souboru příslušný element a aplikace jej automaticky přidá do rolovacího seznamu.

Data, pro rolovací seznamy „Praktický lékař“ a „Pojišťovna“, jsou načítána při inicializaci přímo z databáze. Je-li potřeba přidat položku do těchto rolovacích seznamů, musí se editovat databáze.

5.2.4 Výběr pacienta

Role: Lékař

Uživatel: MUDr. Jan Novák, CSc.
ID: 002002002 [odhlásit](#)

Výběr Pacienta

	ID	Jméno	Příjmení	Datum narození
Výběr	003003003	František	Bouda	1961.02.18
Výběr	005005005	Jaroslav	Hunr	1982.11.03
Výběr	008008008	Roman	Hutka	1987.08.25
Výběr	009009009	Anna	Franková	1982.07.20
Výběr	010010010	Jan	Chromý	1971.05.18

Pacient byl vybrán.

květen 2009

po	út	st	čt	pá	so	ne
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Obrázek 16. Stránka „Výběr Pacienta“

Stránka má jediný úkol. Tím úkolem je přenést do celého systému informaci o identifikaci vybraného pacienta. Po kliknutí na tlačítko „Výběr“ se zavolá metoda, která přeneše do systému informaci o pozici vybraného pacienta v databázi. Metoda vytvoří prvek cookie s názvem „VyberPacienta“. Tento prvek nese jedinou informaci (příklad: UserID=324). Informace tedy určuje pozici vybraného uživatele v databázi a tím ho jednoznačně identifikuje. O vybrání pacienta informuje hláška.

5.2.5 Základní informace

Stránka „Základní informace“ má obsáhlý zdrojový kód. To je dáno tím, že si tuto stránku mohou prohlídnout jak lékaři, tak i pacienti. Platí to i pro jiné stránky systému. Vzhledem k tomu, že lékaři i pacienti mají různá oprávnění pro zobrazování stejných stránek, tak je i kód téměř jakékoli metody rozdělen do dvou částí. Jedna část spravuje práva lékaře a druhá část práva pacienta.

Role:
Základní Informace
Uživatel:
ID: [odhlásit](#)

Nový Pacient

Výběr Pacienta

Základní Informace

Karta Pacienta

Očkovací Průkaz

Objednání

Vypsání Receptu

Jméno:

Příjmení:

ID:

Heslo:

Rodné číslo:

Datum Narození:

Rok Měsíc Den

Telefon:

Email:

Krevní skupina:

Praktický lékař:

Pojišťovna:

Adresa

Ze seznamu

Stát:

Město:

Ulice:

Číslo popisné:

Směrovací číslo:

Upravit

květen 2009						
po	út	st	čt	pá	so	ne
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Obrázek 17. Stránka „Základní Informace“ z pohledu lékaře

Je-li uživatel přihlášen jako lékař, pak se při načtení této stránky zjišťuje, jestli existuje prvek cookie s názvem „VyberPacienta“. Neexistuje-li tento prvek, pak se na stránce nezobrazí žádné informace o pacientovi a vypíše se pouze chybová hláška informující o skutečnosti, že nebyl vybrán žádný pacient.

Existuje-li prvek cookie, pak se na základě v něm uložené informace zavolá procedura uložená na databázovém serveru. Metoda, která tuto akci vykonává, pak informace převzaté z databáze vypíše do příslušných polí formuláře.

Lékař má možnost vypsané informace editovat po zaškrtnutí políčka „Upravit“. Lékař smí editovat všechna pole, které jsou opět kontrolovány na správnost zadání.

Role:
Základní Informace
Uživatel:
ID: [odhlásit](#)

Základní Informace

Přehled lékařů

Karta Pacienta

Očkovací Průkaz

Objednání

Rozvrh

Jméno:

Příjmení:

ID:

Heslo: Zobrazit znaky

Rodné číslo:

Datum Narození:

Rok Měsíc Den

Telefon:

Email:

Krevní skupina:

Praktický lékař:

Pojišťovna:

Adresa

Ze seznamu

Stát:

Město:

Ulice:

Číslo popisné:

Směrovací číslo:

Upravit

< květen 2009 >

po	út	st	čt	pá	so	ne
27	28	29	30	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31
1	2	3	4	5	6	7

Obrázek 18. Stránka „Základní Informace“ z pohledu pacienta

Pokud se přihlásí uživatel pod účtem pacienta, tak se existence prvku cookie „VyberPacienta“ nekontroluje, ale aplikace načítá informace z databáze přímo o přihlášeném uživateli. Jakmile jsou data načtena, tak se vykreslí do příslušných polí.

Pacient má také možnost editovat svůj záznam, nicméně nemá oprávnění upravovat všechna data, jako například krevní skupinu, praktického lékaře atd. Pokud je potřeba změnit praktického lékaře, smí to udělat pouze lékař.

Po kliknutí na tlačítko uložit se v obou případech vykoná databázová procedura pro update záznamu. Všechny vstupní informace jsou kontrolovány validátory.

5.2.6 Přehled lékařů

Základní informace		Přehled Lékařů					květen 2009							
Přehled lékařů		Identifikační pozice	Jméno	Příjmení	Datum narození	Odbornost	Nemocnice	po	út	st	čt	pá	so	ne
Karta Pacienta		2	MUDr. Jan	Novák, CSc	1953.03.25	Praktický lékař	Bařova Nemocnice	27	28	29	30	1	2	3
Očkovací Průkaz		6	MUDr. Marie	Koubková, PhD	1965.06.22	Alergolog	Poliklinika Otrokovice	4	5	6	7	8	9	10
Objednání		11	MUDr. Martin	Drahoš	1977.06.06	Praktický lékař	Poliklinika Uherské Hradiště	11	12	13	14	15	16	17
Rozvrh		23	MUDr. Marie	Šařárová	1975.02.28	Praktický lékař	Bařova Nemocnice	18	19	20	21	22	23	24
								25	26	27	28	29	30	31
								1	2	3	4	5	6	7

Obrázek 19. Stránka „Přehled Lékařů“

S využitím komponenty „GridView“ a jednoduchého databázového dotazu jsou zobrazeny základní informace identifikující lékaře. Sloupec „Identifikační pozice“ je nejdůležitějším údajem v tomto přehledu, neboť je použit pacientem při telefonickém objednání. Záznamy se dají seřadit podle jakéhokoli sloupce.

5.2.7 Karta pacienta

Ke kartě pacienta mají přístup lékaři i pacienti. Funkčnost stránky je opět rozdělena podle uživatelských práv. Rozdíl mezi stránkou načítající se pacientovi nebo lékaři je, že lékaři se navíc zobrazí funkční tlačítka „Nový záznam“, „Uložit“ a zaškrťovací políčko „Upravit záznam“.

Je-li přihlášen lékař, tak se při načtení stránky zjišťuje, existuje-li prvek cookie „VyberPacienta“ v němž je informace, která se pak využívá v databázovém dotazu. Neexistuje-li prvek cookie, tak se nezobrazí žádný záznam a vypíše se chybová hláška. Pokud aplikace našla prvek cookie, pak do pole záznamu načte posledně uložený záznam v kartě pacienta, který napsal přihlášený lékař a do rolovacího seznamu pro datum nastaví příslušné datum vytvoření záznamu. Do pole pro čas se nastaví hodnota času vytvoření záznamu. V rolovacím seznamu pro praktického lékaře se nastaví jméno přihlášeného lékaře. Vybere-li lékař z tohoto seznamu jiného doktora, tak se v rolovacím seznamu pro datum nastaví pouze data, která jsou spojena se záznamy pacienta, které vytvořil vybraný lékař.

Zaškrtnutím políčka „Upravit záznam“ se samotný záznam stane upravitelný. Upravit záznam smí pouze lékař, který záznam vytvořil. Do textových polí pro čas se načte aktuální čas a do rolovacího seznamu se přidá aktuální datum. Tyto textové pole pro čas se začnou chovat jako hodiny. Tím je zajištěno, že při úpravě záznamu se uloží aktuální čas. Do rolovacího seznamu pro praktického lékaře se nastaví jméno přihlášeného lékaře. Pokud se záznam neuloží, pak po odškrtnutí zaškrtačacího políčka se záznam vrátí do původního stavu. Po kliknutí na tlačítko „Uložit“ se záznam přepíše.

Tlačítko pro nový záznam vyvolá akci, která nastaví aktuální datum, jméno přihlášeného lékaře, aktuální čas (opět se příslušná pole tváří, jako hodiny) a v poli záznamu vytvoří volné místo pro nový záznam. Po kliknutí na tlačítko „Uložit“ se formulář ukládá jako nový záznam v databázi.

Obrázek 20. Stránka „Karta Pacienta“

Zobrazí-li tuto stránku pacient, pak má práva pouze prohlížet své záznamy. Při načtení této stránky se nehledá prvek cookie „VyberPacienta“. Funkce rolovacích seznamů je stejná, jako při otevření stránky lékařem.

5.2.8 Očkovací průkaz

Obrázek 21. Stránka „Očkovací Průkaz“

Při načítání stránky se hledá prvek cookie s názvem „VyberPacienta“. Pokud prvek cookie není nalezen, tak se vypíše hláška, že nebyl vybrán žádný pacient. Pakli-že prvek cookie existuje, tak se vypíše tabulka se záznamy očkování vybraného pacienta. Na konec tabulky se vygeneruje přidávací řádek, který umožňuje lékaři objednat pacienta na očkování.

Aplikace se pokusí načíst prvek cookie s názvem „VybraneDatum“. Je-li datum v tomto prvku nastavené na minulost, systém na tuto skutečnost upozorní a formulář je vyhodnocen jako prosté přidání záznamu (při uložení se do sloupce „Status“ v databázi se uloží hodnota „Potvrzeno“), je-li datum nastavené, pak je záznam vyhodnocen jako objednáni (při uložení se do sloupce „Status“ v databázi se uloží hodnota „Odstranit“). Není-li tomu tak, pak se datum vyplní automaticky na den, který je nalezen v prvku cookie. Vybráním data v kalendáři lze tedy toto datum nastavovat. Jméno lékaře, který záznam pořizuje je vyplněno také automaticky jménem přihlášeného lékaře. Rolovací seznam s typem očkování je načítán dynamicky z databáze. Po kliknutí na tlačítko „Uložit“ se záznam uloží s příslušným statusem do očkovací karty pacienta a tabulka se rozšíří o daný záznam.

Jakmile je záznam uložen, tak se také vytváří dva záznamy ve frontě pro posílání automaticky generovaných SMS. Jestli-že je záznam vyhodnocen jako pouhé přidání záznamu, pak se do fronty nepřidává žádný záznam pro upozornění.

Ve frontě má jeden záznam tři položky, kterými jsou informace o datu upozornění, text, který se má odeslat a telefonní číslo, na které se má text odeslat. V položce text jsou uloženy informace o typu upozornění (v tomto případě očkování), datum návštěvy a jméno lékaře, ke kterému se má pacient dostavit.

Data upozornění jsou nastavena na sedm a dva dny před plánovanou návštěvou. Ve frontě se tedy vytvoří dva záznamy. Pokud vypočítané datum upozornění je již v minulosti, pak se do fronty nepřidává a tím pádem se nepošle upozornění na plánovanou návštěvu. Na tuto skutečnost je lékař také upozorněn.

Lékař smí mazat pouze záznamy z očkovací karty pacienta, které sám vytvořil. To se provádí pomocí tlačítka „Odstranit“, které je zobrazeno v závislosti na statusu daného záznamu a na faktu, zdali daný záznam vytvořil přihlášený lékař. Jakmile je záznam odstraněn lékařem, tak se z fronty pro odesílání informativních SMS odstraní eventuelní záznamy pro upozornění a pak se do fronty vloží jeden záznam, který má stejnou podobu, jako upozornění na objednání. Rozdíl je v datu upozornění, které je nastaveno na nejbližší datum prohlížení fronty a v textu budoucí zprávy je informace o zrušení objednání pacienta.

Tlačítko „OK“ slouží k potvrzení záznamu (ve sloupci „*Status*“ se přepíše hodnota na status „*Potvrzeno*“). Toto tlačítko je dostupné pouze tehdy, je-li datum očkování nastavené na minulost. Není-li do požadovaného data potvrzen záznam, pak se automaticky smaže, aby se neukládala nevykonaná očkování.

5.2.9 Objednání

Stránka „Objednání“ patří mezi nejsložitější stránky celého systému. Je zde potřeba mnoho databázových dotazů (procedur), propojení s kalendářem a spousta úprav formuláře za běhu.

Role: Lékař

Objednání

Uživatel: MUDr. Jan Novák, CSc
ID: 002002002 [odhlásit](#)

Pacient
Anna Franková

Praktický lékař
MUDr. Jan Novák, CSc

Datum
2009.06.19

Čas

Hodina	Minuta	Status	Povoleno	Objednat	Zrušit
07	00	ZAKÁZANO	<input type="checkbox"/>	Objednat	Zrušit
	15	ZAKÁZANO	<input type="checkbox"/>	Objednat	Zrušit
	30	ZAKÁZANO	<input type="checkbox"/>	Objednat	Zrušit
	45	ZAKÁZANO	<input type="checkbox"/>	Objednat	Zrušit
08	00	Martina Tichá	<input checked="" type="checkbox"/>	Objednat	Zrušit
	15	Roman Plachý	<input checked="" type="checkbox"/>	Objednat	Zrušit
	30	VOLNO	<input checked="" type="checkbox"/>	Objednat	Zrušit
	45	Marcel Sup	<input checked="" type="checkbox"/>	Objednat	Zrušit
09	00	Marcel Sup	<input checked="" type="checkbox"/>	Objednat	Zrušit
	15	VOLNO	<input checked="" type="checkbox"/>	Objednat	Zrušit
	30	ZAKÁZANO	<input type="checkbox"/>	Objednat	Zrušit

Calendar: červen 2009

Obrázek 22. Stránka „Objednání“ z pohledu lékaře

Je-li při inicializaci stránky přihlášeným uživatelem lékař, tak tato stránka slouží i jako jeho rozvrh. Při inicializaci stránky se do rolovacího seznamu pro praktického lékaře nastaví jméno přihlášeného lékaře. Do pole pro datum se nastaví aktuální den (ve skutečnosti se datum nastaví pomocí prvku cookie „VybraneDatum“ podle vybraného data v kalendáři) a vypíše se do tabulky jeho rozvrh na tento den. Do polí, kde není objednan žádný pacient, se vypíše hláška „VOLNO“. Odškrtnutím políčka „Povoleno“ se do příslušného pole vypíše hláška „ZAKÁZANO“, tlačítka „Objednat“ a „Zrušit“ se zablokují. Zobrazuje-li se rozvrh přihlášeného lékaře, pak se do polí, která jsou již obsazena, vypíše jméno pacienta, který má termín zablokovaný. Je-li vybrán jiný lékař, pak se tyto jména nahradí hláškou „OBSAZENO“.

Jestli-že chce lékař objednat pacienta, musí existovat prvek cookie „VyberPacienta“. Jestliže neexistuje, pak se zablokují všechny tlačítka „Objednat“ a „Zrušit“ a vypíše se hláška informující o této skutečnosti. Po kliknutí na tlačítko „Objednat“ se vytvoří příslušný záznam v databázi. Existuje-li daný záznam v databázi, pak se provede jeho

úprava. Hlášky „VOLNO“ a „ZAKÁZÁNO“ se ukládají do databáze (sloupec „Status“), aby bylo možné nastavovat tlačítka při dalším zobrazování denního rozvrhu. Tlačítko „Zrušit“ vymaže záznam z databáze a nastaví příslušný status na „VOLNO“.

Role: Pacient

Objednání

Uživatel: Anna Franková
ID: 009009009 [odhlásit](#)

Základní Informace
Přehled lékařů
Karta Pacienta
Očkovací Průkaz
Objednání
Rozvrh

Pacient
Anna Franková

Praktický lékař
MUDr. Jan Novák, CSc.

Datum
2009.06.19

Čas

Hodina	Minuta	Status	Objednat	Zrušit
07	00	ZAKÁZÁNO	Objednat	Zrušit
	15	ZAKÁZÁNO	Objednat	Zrušit
	30	ZAKÁZÁNO	Objednat	Zrušit
	45	ZAKÁZÁNO	Objednat	Zrušit
08	00	OBSAZENO	Objednat	Zrušit
	15	OBSAZENO	Objednat	Zrušit
	30	VOLNO	Objednat	Zrušit
	45	OBSAZENO	Objednat	Zrušit
09	00	OBSAZENO	Objednat	Zrušit
	15	VOLNO	Objednat	Zrušit
	30	ZAKÁZÁNO	Objednat	Zrušit

červen 2009

po	út	st	čt	pá	so	ne
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Obrázek 23. Stránka „Objednání“ z pohledu pacienta

Přístupuje-li ke stránce pacient, pak inicializace stránky je stejná, jako když je přihlášen lékař. Z pochopitelných důvodů se pacientovy nezobrazují zaškrťovací políčka „Povolit“. To je funkce určená pouze lékařům. Pro pacienta je důležité, že se zobrazí pouze volné termíny, na které se může zapsat. Jakmile se pacient запиše na nějaký termín, pak už se nemůže zapsat na jiný termín. Řešením je zrušit objednávku a přihlásit se na jiný termín.

Jakmile lékař objedná pacienta nebo se pacient objedná sám, tak nejen že se uloží záznam do databáze, ale zároveň se také vloží opět dva záznamy ve frontě pro posílání informativních zpráv SMS. Princip a formát vytvoření záznamů ve frontě je stejný, jako u objednání na očkování. Rozdíl je pouze v tom, že se změní typ návštěvy lékaře. Při zrušení objednání lékařem se vymaže příslušný záznam v databázi (popřípadě upozorňující záznamy ve frontě) a do fronty informativních SMS se vloží jeden záznam, který je vytvořen stejně, jako při zrušení objednání očkování.

Samotné generování SMS probíhá tak, že se v určitou dobu prochází fronta a hledají se záznamy s datem upozornění, které odpovídají aktuálnímu datu. Všechny vyhovující záznamy, respektive její položka „text“, se odešlou na telefonní číslo uložené v záznamu.

Objednání pomocí SMS

Jakmile systém obdrží SMS zprávu od pacienta, tak aplikace validuje text zprávy. Zjišťuje, zdali existuje pacientovo ID a lékař, ke kterému se chce pacient objednat. Jestliže textová zpráva nevyhovuje předepsanému formátu, zašle se SMS zpráva (na mobilní telefon, ze kterého požadavek přišel), informující pacienta o neplatném požadavku.

Projde-li textová zpráva kontrolou, tak se z ní extrahují jednotlivé řetězcové hodnoty. Poté se databázovým dotazem zjistí, jestli v požadovaném datu má lékař volno. Jestli se vrátí pozitivní výsledek, tak se pacient запиše na první volnou hodinu. Vráti-li se negativní výsledek, tak se v cyklu prohledávají následující dny, a pacient se запиše do nejbližšího volného termínu.

Ať už se uložení podaří nebo nepodaří, je vygenerována SMS odpověď a vzápětí poslána na telefonní číslo, ze kterého požadavek pochází.

5.2.10 Vypsání receptu

The screenshot shows the 'Vypsání Receptu' (Prescription) interface. At the top, the role is set to 'Lékař' (Doctor) and the user is identified as 'MUDr. Jan Novák, CSc.' with ID '002002002'. The main area is titled 'Vybraný pacient' (Selected patient) and shows 'Anna Franková'. Below this, there is a text area containing the prescription: '2x Paralen - 3x denně jedna tableta' and '1x Stopangin - večer 30 kapek'. At the bottom of the text area are buttons for 'Odeslat' (Send) and 'Tisk' (Print). On the right side, there is a calendar for May 2009, with the 17th highlighted in green.

Obrázek 24. Stránka „Vypsání Receptu“

Jakmile se načte stránka, proběhne kontrola na přítomnost prvku cookie s názvem „VyberPacienta“ a jestliže prvek neexistuje, pak se vypíše hláška, že nebyl vybrán žádný pacient. Odeslat recept lze jen tehdy, když je textové pole vyplněno. Po kliknutí na tlačítko „Odeslat“ se vyplněný formulář odešle do lékárny. K receptu se automaticky přidávají

informace jako aktuální datum, jméno lékaře, který recept vypsál a jméno pacienta, pro kterého je recept určen. Tyto informace lze také vytisknout kliknutím na tlačítko „Tisk“.

5.2.11 Rozvrh

The screenshot shows a web application interface for a schedule. At the top, there is a header with the title 'Rozvrh' and user information: 'Uživatel: Roman Hutka' and 'ID: 008008008'. Below the header, there is a navigation menu on the left with options like 'Základní Informace', 'Přehled lékařů', 'Karta Pacienta', 'Očkovací Průkaz', 'Objednání', and 'Rozvrh'. The main content area contains a table with the following data:

Čas	Akce	Poznámka
08:30	Pacient je objednan k lékaři: MUDr. Marie Šafářová.	
11:30	Pacient je objednan k lékaři: MUDr. Jan Kovadina	Očkování

On the right side, there is a calendar for June 2009. The calendar shows the days of the week (po, út, st, čt, pá, so, ne) and the dates from 29 to 3. The date 31 is highlighted in green.

Obrázek 25. Stránka „Rozvrh“

Po vybrání data z kalendáře se při načtení stránky hledá prvek cookie s názvem „VybraneDatum“. Z něj se načte hodnota vybraného data a následně je poslán dotaz na SQL server, zdali existuje záznam daného data a pacienta. Pokud ano, pak jsou vrácené výsledky vypsány do tabulky. Pokud záznam neexistuje, pak není vypsán žádný záznam.

6 UŽIVATELSKÁ DOKUMENTACE

Uživatelské prostředí elektronické ordinace je intuitivní a lehce ovladatelné. Systém nedovolí udělat uživateli žádnou nepovolenou akci. Přístup ke stránkám webové aplikace mají pouze ověření uživatelé. Na základě jejich uživatelské role jsou jim přiřazena práva.

6.1 Administrátor

Administrátorský účet slouží k správě lékařů. Uživatel, přihlášený pod tímto účtem, smí lékaře do systému přidávat, upravovat jejich záznam a mazat je ze systému. K vytvoření účtu administrátora je nutné editovat přímo databázi.

Přidání záznamu nového lékaře

Na stránce „Nový Lékař“ administrátor vyplní všechny požadované položky formuláře. Při vyplňování textového pole pro jméno nejdříve administrátor vypíše všechny tituly lékaře před jménem a poté vyplní jeho jméno. Do textového pole pro příjmení vyplní administrátor nejdříve příjmení lékaře, a poté všechny tituly za jménem. Předešlé věty jsou spíše doporučením, než nutností proto, aby se přímo v databázi dali rozeznat od pacientů.

Do pole „ID“ se smí zadat pouze číslice. Počet číslic je devět, víc ani méně jich být nesmí. Heslo má maximální velikost patnáct znaků. Do textového pole pro rodné číslo se zapíše nejdříve šest číslic, lomítko a na konec tři nebo čtyři číslice. Datum narození musí souhlasit s částí rodného čísla před lomítkem. Rolovací seznam pro den se zobrazí až po vybrání roku a měsíce data narození. Textové pole s názvem nemocnice a část formuláře pro zadání adresy jdou propojeny. Nelze přiřadit nemocnici adresu pacienta. Pokud administrátor využije možnosti vybrání adresy ze seznamu, pak po jejím vybrání se doplní název nemocnice automaticky.

Jakmile jsou všechny požadované informace ověřeny příslušnými validátory, pak je možné nového lékaře uložit do databáze.

Přehled lékařů

Stránka „Lékaři“ umožňuje kompletní přehled lékařů, uložených v databázi. V prvním sloupci tabulky jsou dvě tlačítka. Kliknutím na tlačítko „Odstranit“ se vybraný záznam lékaře z databáze odstraní. Tlačítko „Upravit“ složí k editaci záznamu lékaře. Po kliknutí na tlačítko se do prvního sloupce vloží tlačítka „Aktualizovat“ a „Storno“. Příslušný řádek se změní na formu editování záznamu a tlačítkem aktualizovat se záznam

v databázi přepíše. Tlačítko storno zruší editační mód. Vstupy jsou hlídány validátory, takže nelze aktualizovat data neplatným uživatelským vstupem.

6.2 Lékař

Lékař provádí administraci záznamů o pacientech a při vyšetření zaznamenává svá zjištění do systému.

Nový pacient

Zde platí stejná pravidla, jako při vytváření nového lékaře. Jsou pouze přidány jiné informace. Email je jediná položka, která nemusí být vyplněna. Pokud ji lékař vyplní, pak aplikace očekává správný tvar emailové adresy. Emailová adresa se skládá ze tří částí. První a druhá část je oddělena zavináčem, druhá a třetí část je oddělena tečkou. Telefonní číslo je devíti místně bez předvolby. Nelze přiřadit pacientovy adresu, která je již přiřazena k nějaké nemocnici. Lze však pacientům s jinými příjmeními přiřadit stejnou adresu. Jakmile jsou všechny požadované informace vyplněny správně, lze nového pacienta uložit.

Výběr pacienta

Jedinou povolenou akcí je kliknutí na tlačítko, které vybere pacienta. Po vybrání pacienta se zpřístupní další funkce v systému. Dokud tedy není vybrán pacient, nelze provádět žádnou akci související s existujícím pacientem na následujících stránkách aplikace.

Základní informace

Na této stránce se lékaři zobrazí informace o vybraném pacientovi. Lékař smí všechny tyto informace upravovat. Pokud lékař zvolí možnost upravit záznam, pak je postup totožný s postupem zadávání nového pacienta do systému.

Karta pacienta

Stránka umožňuje lékaři prohlížet zdravotní kartu pacienta. Má i možnost editovat záznamy v kartě pacienta, ale pouze ty, které sám vytvořil. Tato opatření jsou zavedena z důvodu bezpečnosti. Nemělo by se stát, aby lékař napsal zprávu o zdravotním stavu pacienta a někdo jiný ji změnil. Prohlédnutí záznamu v kartě se provádí tak, že se nejdříve vybere lékař, poté datum záznamu.

Pokud lékař chce zadat nový záznam do karty pacienta, stačí kliknout na příslušné tlačítko a začít rovnou psát záznam o vyšetření. Po kliknutí na tlačítko uložit se

automaticky uloží i aktuální čas i datum. Lékař se o tuto drobnost nemusí starat ani při vytváření nového záznamu, ani při editaci existujícího záznamu.

Očkovací průkaz

Na této stránce si může lékař prohlédnout očkovací záznamy vybraného pacienta. Má možnost objednat pacienta na očkování vyplněním posledního řádku vypsané tabulky. V kalendáři vybere datum, na které chce pacienta objednat a toto datum se vloží do příslušného sloupce tabulky. Je-li datum v minulosti, pak je záznam považován jako doplnění očkovacího průkazu, je-li datum situované do budoucnosti, pak se záznam bere jako objednání na očkování. Jméno lékaře se také vkládá automaticky a je vloženo jméno lékaře, který je aktuálně přihlášen. Lékař musí pouze vybrat typ očkování. Poté je možné záznam uložit.

Lékař smí mazat pouze záznamy, který sám vytvořil. Smazat lze ale jen záznamy, které jsou svým datem určeny do budoucnosti nebo týden do minulosti. Toto opatření je přijato proto, že se pacient na očkování nemusí dostavit, a bude tak potřeba vytvořený záznam smazat. Pokud pacient na očkování dorazí, lékař daný záznam potvrdí tlačítkem „OK“ a záznam se definitivně uloží do očkovacího průkazu pacienta. Pokud lékař nepotvrdí ani neodstraní vytvořený záznam pro objednání a uběhne týden od data objednání, pak bude očkování označeno za nevykonané a záznam se automaticky smaže.

Objednání

Lékař objednává pacienta pouze pomocí formuláře. Lékař vybere v rolovacím seznamu jméno lékaře (při načtení stránky je defaultně nastaveno jméno přihlášeného lékaře), ke kterému chce pacienta objednat a v kalendáři vybere datum na, které chce pacienta objednat. Pak se zobrazí v tabulce rozvrh na daný den pro vybraného lékaře.

Pro objednání pacienta se předpokládá, že lékař na stránce „Vyber pacienta“ zvolil pacienta, kterého chce objednat, jinak nebude přístupné žádné tlačítko „Objednat“ a lékař na to bude upozorněn hláškou. V položce tabulky, kde je zeleně vypsáno „VOLNO“, smí lékař kliknutím na příslušné tlačítko pacienta objednat. Jakmile je pacient objednan, nelze ho objednat na daný den v jinou dobu.

Zrušit objednání kteréhokoli pacienta smí lékař kdykoli příslušným tlačítkem. Zaškrťovací políčko, které je v každém řádku tabulky, slouží k naplánování přestávek či dovolené. Je-li políčko neoznačené, pak se na tento čas ve vybraném datu nemůže nikdo zapsat.

Pokud lékař objednává pacienta k jinému lékaři, pak má lékař prakticky stejné možnosti objednání, jako když se objednává pacient sám.

Vypsání receptu

Před vypsáním receptu, je nutné mít vybraného pacienta. Vypsání receptu je jednoduché. Lékař musí pouze vyplnit textové pole, určené pro samotný recept. Poté ho stačí jen odeslat nebo vytisknout.

6.3 Pacient

Pacientovi slouží webová aplikace k prohlížení svých záznamů a pro rychlé objednání k lékaři.

Základní informace

Informace zobrazené na této stránce se vztahují k přihlášenému uživateli a při prvním načtení si je lze pouze prohlédnout. Rozhodne-li se pacient upravit nějaké údaje o svojí osobě, stačí označit zaškrťovací políčko. Po jeho zaškrtnutí se mohou editovat jen některé informace, jako například telefon, email či pojišťovna. Nelze editovat například rodné číslo a datum narození, protože existuje oprávněný předpoklad, že se tyto informace nemění.

Pro vyplnění povolených údajů lze tyto informace uložit příslušným tlačítkem. Opět je nutné splnit požadavky na správnost zadání vstupních polí.

Přehled lékařů

Na této stránce nejsou vyžadovány žádné uživatelské vstupy. Stránka slouží pouze pro získání identifikačního čísla lékaře, které se dá využít při objednání pomocí SMS.

Karta pacienta

Pro pacienta má tato karta jen informativní charakter. Nelze zde nic upravovat. Pacient může pouze uložené záznamy prohlížet.

Objednání

Objednání k lékaři má dvě podoby. První je formou vyplnění formuláře a druhá je pomocí SMS.

Formulářové objednání je z pohledu pacienta jednoduché. Stačí vybrat lékaře, ke kterému se chce pacient objednat a poté v kalendáři vybrat datum, které pacientovi

vyhovuje. Pacienta zajímají pouze zeleně vyplněné pole, ve kterých je text „VOLNO“. V každém řádku tabulky se zelenou hláškou je tlačítko „Objednat“. Po kliknutí na tlačítko je pacient objednan k lékaři. Jakmile je pacient objednan, nelze se již přihlásit na **žádný** jiný termín.

Zrušit objednávku provede pacient kliknutím na příslušné tlačítko. Zrušit objednání může pacient pouze do dvou dnů před plánovaným termínem. Poté tuto akci smí provést pouze lékař.

Objednání pomocí SMS je také jednoduché, stačí pouze v mobilním telefonu napsat SMS zprávu v již zmíněném předepsaném tvaru a aplikace již pacienta objedná. O výsledku objednání je pacient obratem informován.

Rozvrh

Na této stránce nejsou vyžadovány žádné uživatelské vstupy. Pacient si zde může prohlédnout rozvrh na vybraný den v kalendáři. Jsou zde informace, jako například že je přihlášený pacient objednan na desátou hodinu k lékaři a na třináctou hodinu na očkování k jinému lékaři.

7 ROZŠÍŘENÍ SYSTÉMU DO BUDOUCNA

System elektronické ordinace je pouze základní stavební kámen možného komplexního systému pro zdravotnictví a má pouze nejzákladnější možnosti. Tyto možnosti lze dále rozšiřovat.

7.1 Vzhled a rozvržení stránek

Nedílnou součástí všech webových stránek je grafický vzhled a rozvržení stránek. Jednou z věcí, které by se tedy mohli do budoucna změnit, je právě jednoduchý vzhled elektronické ordinace. Je důležité, aby se uživatel navštěvující webové stránky elektronické ordinace cítil dobře.

Grafika by měla využít uklidňujících barev, zajímavého designu a nových technologií. Velmi zajímavé možnosti nabízí technologie WPF, respektive její webová podoba Silverlight, od Microsoftu.

Rozvržení stránek má klasickou podobu, ale při využití zmíněné technologie je pravděpodobné, že se rozvržení stránek bude dynamicky měnit.

7.2 Databáze uživatelů

Databáze je navrhnutá pouze k základnímu chodu systému. Jsou v ní, mimo jiné, uloženy informace potřebné k přihlášení a identifikaci uživatele.

Jedním z prvních kroků rozšíření by mělo být doplnění databáze o všechny relevantní a potřebné data. Hlavně rozšíření databáze z hlediska lékařů, například uvedení oddělení lékaře, přidání tabulky s typem diagnóz, přidání tabulky se seznamem odbornosti lékaře a další upřesňující informace. K tomuto rozšíření je potřeba dlouhodobá spolupráce z co největším spektrem lékařů, kteří by svými praktickými zkušenostmi mohli zefektivnit celou logiku databáze. Dále by se databáze mohla rozšířit o multimediální data, jako jsou záznamy EKG, EEG, ECHO, rentgeny, CT, atd. To by pomohlo ke zlepšení komunikace mezi odděleními nemocnice a ke zmenšení čekací doby na výsledky vyšetření.

Pro zjednodušení jsou přihlašovací údaje ukládány v nešifrované formě, což by bylo při nasazení systému do praxe nežádoucí, a proto by se tyto údaje měly vhodnou formou šifrovat.

7.3 Funkce systému

V dnešní době se k přihlášení do zabezpečené webové aplikace nejčastěji používá formulářové ověření. I systém elektronické ordinace používá tento typ autentizace. Nicméně existují i jiné druhy ověření totožnosti, jako je například ověření pomocí biometrických prvků. Takový typ ověření by mohl ulehčit práci například lékařů a tím snížit časovou náročnost vyšetření.

Představme si následující scénář. Lékař přichází do své ordinace a pro otevření dveří přiloží ruku ke čtečce biometrických údajů. Čtečka přečte požadovaná data a vpustí ověřeného lékaře do ordinace. Data, které čtečka přečetla, budou uloženy v šifrované podobě na disku serveru a budou svázaná s IP adresou lékařova počítače. Lékař zapne svůj počítač, spustí prohlížeč s URL adresou elektronické ordinace, která je uložena jako domovská stránka. Při požadavku na URL adresu elektronické ordinace se server pokusí načíst data uložená čtečkou. Pokud data najde, pak automaticky lékaře přihlásí do systému, pokud je nenajde, pak se přistoupí k formulářovému ověřování.

Jakmile začne lékař ordinovat a přijde první pacient, pak musí lékař v základní verzi elektronické ordinace vybírat pacienta ručně ze seznamu pacientů. Po zavedení biometrických prvků by systém mohl automaticky načítat přicházejícího pacienta a lékaři by se tak opět ušetřil čas.

Dalším užitečným rozšířením by mohla být funkce, která by měla na starost vytváření nemocenských a omluvenek pro školáky. Automaticky by vkládala do formuláře potřebné informace, například jméno pacienta, datum, důvod návštěvy, razítko lékaře a možnost vytisknout formulář.

7.4 Ostatní rozšíření

Každou aplikaci je potřeba po určitém čase testování upravit. Již při tvorbě webové aplikace existují drobná vylepšení, která nemají vliv na funkci systému a měli by být zavedeny do webové aplikace.

Například by mohl lékař mít možnost naplánovat svou dovolenou a tím zakázat zobrazení určitého data v kalendáři, aby se na požadované datum nemohl nikdo objednat. Teoreticky taková možnost v systému existuje, ale je poměrně komplikovaná, což není v souladu s hlavní myšlenkou aplikace, kterou lze popsat jedním slovem – jednoduchost.

Pokud by byl na požadované datum někdo objednan, pak by systém automaticky přeobjednal pacienta na volné datum.

Dalším z drobných rozšíření by mohlo být při výběru libovolného města automatické doplňování směrovacích čísel a podobné drobnosti. Neméně užitečným rozšířením by mohla být i drobná úprava uživatelských vstupů. Některé požadované informace, jako například odbornost lékaře, musí uživatel zapsat ručně do „*TextBoxu*“. To není nejlepší řešení, nicméně pro účely základní verze aplikace plně dostačující. Jakmile by existoval nějaký úplný seznam odborností lékaře, pak by se tento seznam mohl dynamicky načítat do rolovacího seznamu, aby nedocházelo ke špatnému zadání odbornosti lékaře.

Na konec by se dalo uvažovat o změně či přidání uživatelských práv některých uživatelů, hlavně uživatele přihlášeného pod administrátorským účtem.

ZÁVĚR

Úkolem práce bylo navrhnout a vytvořit prototyp webové aplikace, která by zpravovala administraci ordinace. Zadání má v dnešní době digitalizace dat velký význam.

Prvním krokem řešení bylo vybrání vhodných technologií. Technologie a softwarové prostředky, které byly použity, jsou ve všech případech komerční produkty. Pro správu databáze byl vybrán MS SQL Server 2008. Vývoj a ladění stránek webové aplikace byl prováděn v MS Visual Studiu 2008. Webová aplikace byla vytvořena pro platformu .NET, respektive ASP.NET. Pro programování funkční logiky byl použit jazyk C#.

Pro splnění zadání bylo využito všech poznatků, které se nacházejí v teoretické části práce. Nejdříve byla provedena úvodní studie a rozbor zadání. Potom byl pomocí systémové analýzy navrhnout teoretický model systému. Na základě metody BORM byl pomocí CASE nástroje vytvořen objektový model navrženého systému. Verze použitého nástroje Craft.CASE byla demoverze, která nedovolovala ukládat a spravovat víc, jak dvacet objektů. Proto bylo těchto teoretických poznatků využito pouze pro navrhnutí základního obecného modelu systému elektronické ordinace. Nasazení a zkušební provoz systému byl proveden na krátký čas z důvodu blížícího se termínu odevzdání práce.

Jakmile byl navrhnout systém, tak se přistoupilo k jeho realizaci. Nejdříve bylo nutné seznámit se s obecným principem běhu webové aplikace. Poté proběhlo vytváření designu jednotlivých webových stránek systému. Nakonec byla všem funkčním blokům naprogramována jejich funkce.

Pro bezproblémový chod webové aplikace eHealth je třeba mít správně nastavený webový prohlížeč tak, aby měl povolenou správu souborů cookies. Dále je třeba mít administrátorská práva pro přístup k databázi, která je vytvořena na samostatném SQL serveru. Design webové aplikace eHealth není vytvořen pro žádný konkrétní webový prohlížeč. Všechny bloky jsou pomocí CSS stylů pozicovány absolutně.

Byla vytvořena webová aplikace, která splňuje základní požadavky pro bezproblémový chod ordinace. Webová aplikace eHealth klade důraz na bezpečnost a jednoduchost uživatelských vstupů a to způsobuje rozsáhlost zdrojového kódu celé webové aplikace.

FINISH OF ENGLISH LANGUAGE

The general task of this work was suggested and created a prototype web application that would manage administration of medical office. This assignment has great importance at the present time digitalization of data.

The first step of solution was finding the efficient technology. Used technologies and software resources are all commercial version. MS SQL Server 2008 was used for control database and MS Visual Studio 2008 was used for development and debugging pages of web application. The web application was created for the platform .NET let us say ASP.NET. The language C# was used for programming function logic of the system e-health.

All knowledge of theoretical part this work was used for created web application. The start study and analysis project was suggested earliest. Then, the theoretical model of system e-health was suggested base on system analysis. The object model of proposed system was created base on method BORM by the help of tool Craft.CASE. Version of used tool Craft.CASE was demo, which did not allow to store and manage more as twenty objects. Hence the theoretical knowledge was used for proposed the general model of system e-health only. The testing operation of system e-health was performed for a short time.

Once the system was proposed that it was approached its implementation. First step was necessary become acquainted with function and running web application. Next it was created the design single pages of web application. Finally it was programmed function for all functional blocks.

It must be properly configured web browser so that it has authorized management of cookies for smooth behavior the web application e-health. In addition, you must have administrator authorized for access into database of system e-health. The design of web application e-health is not designed for any specific web browser. All blocks are using CSS absolute positioning styles.

It was created the web application e-health that satisfactory all requirements for smooth functional of medical office. This web application keeps the great requirements for security and basic users input.

CITOVANÁ LITERATURA

1. **MediaWiki.** *Wikipedia otevřená encyklopedie.* [Online] <http://cs.wikipedia.org>.
2. **doc. Ing. Ivan Zelinka, Ph.D, Ing. Bc. Bronislav Chramcov, Ph.D.** *Skripta do předmětu Základy Informatiky.* [Dokument PDF] Zlín : autor neznámý, 2003.
3. **Šilerová, Iveta.** *Informační systémy - přednášky.* [Dokument DOC]
4. **Štíbal, Martin.** *Strukturovaná analýza informačních systémů. Bakalářská práce.* Brno : MASARYKOVA UNIVERZITA FAKULTA INFORMATIKY, 2008.
5. **Ing. Vojtěch Merunka, Ph.D.** *Objektově orientovaná analýza informačních systémů.* [Dokument PDF] Praha : ProfComputing, 2005.
6. **Payne, Chris.** *Naučte se ASP.NET za 21 dní.* Praha : Computer Press, 2002.
7. **Prosise, Jeff.** *Programování v Microsoft .NET Webové aplikace v .NET Framework, C# a ASP.NET.* Praha : Computer Press, 2003.
8. **Dušan, Janovský.** *CSS. Jak psát web.* [Online] www.jakpsatweb.cz.
9. **Rastislav, Škultéty.** *JavaScript - programujeme internetové aplikace.* Praha : Computer Press, 2001.
10. **doc. Ing. Zdeňka Prokopová, CSc.** *Databáze - přednášky.* [Prezentace PPT]
11. **Petr Kaleta.** *MS SQL Server 2008. .netstudent.* [Online] microsoft.cz, 2006 - 2007. www.netstudent.cz.
12. *Elektronické zdravotnictví (eHealth).* [Online] 27. 11 2008. http://ec.europa.eu/health-eu/care_for_me/e-health/index_cs.htm.
13. *ASP.NET 2.0 a C-sharp: tvorba dynamických stránek profesionálně.* Brno : Zoner Press, 2006. 1376 s.
14. *Systems engineering principles and practice.* New York : Wiley, 2004. 465 s.
15. *Softwarové inženýrství.* Praha : Academia, 1991. 324 s.
16. **Šilhavý Petr, Šilhavý Radek, Snášel Václav.** *Web-based Patient-Physician Communication.* Ostrava : Faculty of Electrical Engineering and Computer Science, VŠB Technical University of Ostrava, 2008.

SEZNAM OBRÁZKŮ

Obrázek 1. Digram datových toků.....	15
Obrázek 2. Příklad využití nástroje Craft.CASE	17
Obrázek 3. Postup u klasické aplikace versus systém .NET	23
Obrázek 4. Vztah mezi klientem a serverem s ASP.NET stránkou.....	26
Obrázek 5. Ukázka HTML kódu	27
Obrázek 6. Skript umístěného uvnitř HTML.....	29
Obrázek 7. Skript umístěný externě.....	29
Obrázek 8. Relační diagram.....	38
Obrázek 9. Příklad tvarů pro SMS.....	43
Obrázek 10. Design a logické rozdělení aplikace.....	44
Obrázek 11. Přihlašovací stránka.....	45
Obrázek 12. Chybová stránka	46
Obrázek 13. Stránka „Nový Lékař“	47
Obrázek 14. Stránka „Lékaři“	48
Obrázek 15. Stránka „Nový pacient“	49
Obrázek 16. Stránka „Výběr Pacienta“	50
Obrázek 17. Stránka „Základní Informace“ z pohledu lékaře	51
Obrázek 18. Stránka „Základní Informace“ z pohledu pacienta	52
Obrázek 19. Stránka „Přehled Lékařů“	53
Obrázek 20. Stránka „Karta Pacienta“	54
Obrázek 21. Stránka „Očkovací Průkaz“	55
Obrázek 22. Stránka „Objednání“ z pohledu lékaře	57
Obrázek 23. Stránka „Objednání“ z pohledu pacienta	58
Obrázek 24. Stránka „Vypsání Receptu“	59
Obrázek 25. Stránka „Rozvrh“	60

SEZNAM TABULEK

Tabulka 1. Některé ze jmenných prostorů FCL.....	25
Tabulka 2. Uživatelé.....	33
Tabulka 3. Přihlašovací údaje.....	34
Tabulka 4. Uživatelské role.....	34
Tabulka 5. Informace o Pacientovi.....	34
Tabulka 6. Adresa.....	35
Tabulka 7. Pojišťovna.....	35
Tabulka 8. Informace o lékaři.....	35
Tabulka 9. Zdravotní karta.....	36
Tabulka 10. Očkovací karta.....	36
Tabulka 11. Typ očkování.....	36
Tabulka 12. Rozvrh.....	37
Tabulka 13. Typ vyšetření.....	37