

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Diplomová práce

Informační systém zdravotnického zařízení

Bc. Lukáš Tvrz

© 2020 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Lukáš Tvrz

Systémové inženýrství a informatika
Informatika

Název práce

Informační systém zdravotnického zařízení

Název anglicky

Information System for Medical Center

Cíle práce

Hlavním cílem diplomové práce je vytvoření informačního systému pro nestátní zdravotnické zařízení se zaměřením na sportovní medicínu. Informační systém bude vytvořen za použití moderních technologií pro tvorbu dynamických webových aplikací.

Metodika

Teoretická část diplomové práce bude založena na studiu odborné literatury. Bude popsána problematika návrhu informačních systémů a nástroje použité pro jeho návrh a vývoj. Praktická část diplomové práce se bude zabývat analýzou potřeb a návrhem informačního systému pro zdravotnické zařízení. Návrh bude zpracován pomocí modelovacího jazyka UML. Podle vytvořeného návrhu proběhne následná implementace samotného informačního systému.

Doporučený rozsah práce

50-60 stran

Klíčová slova

UML, OOP, modely, informační systém, návrh, diagram, webový informační systém, javascript, angularjs

Doporučené zdroje informací

BRUCKNER, T. Tvorba informačních systémů : principy, metodiky, architektury. Praha: Grada, 2012. ISBN 978-80-247-4153-6.

KANISOVÁ, H., MÜLLER M.: UML srozumitelně, Computer press, 2006, 176 s. ISBN: 80-251-1083-4

MERUNKA, V., POLÁK, J., CARDA, A., ČESKÁ SPOLEČNOST PRO SYSTÉMOVOU INTEGRACI. Umění systémového návrhu : objektivě orientovaná tvorba informačních systémů pomocí původní metody BORM. Praha: Grada, 2003. ISBN 80-247-0424-2.

ŘEPA, V., CHLAPEK, D., STANOVSKÁ, I. Analýza a návrh informačních systémů. Praha: Oeconomica, 2011. ISBN 978-80-245-1782-7.

Předběžný termín obhajoby

2019/20 LS – PEF

Vedoucí práce

Ing. Marek Pícka, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 19. 2. 2020

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 29. 11. 2020

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Informační systém zdravotnického zařízení" jsem vypracoval(a) samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.11.2020

Poděkování

Rád bych touto cestou poděkoval mému vedoucímu práce panu Ing. Marku Píckovi, Ph.D., za odborné rady a konzultace během vedení této diplomové práce

Informační systém zdravotnického zařízení

Abstrakt

V teoretické části této diplomové práce jsou vysvětlovány a definovány základní pojmy týkající této práce. Popsána je problematika vývoje informačních systému, dále je zde definován jazyk UML, který slouží pro návrh software. V závěrečné kapitole je popsána problematika Front-end technologií, které se využívají pro tvorbu webových stránek.

Praktická část je rozdělena na dvě hlavní kapitoly, nejprve je provedena analýza řešeného problému, jejímž výsledkem jsou požadavky na funkcionalitu, kterou by měl informační systém splňovat. Na základě takto vytvořených požadavků byly vytvořeny návrhové diagramy, podle nichž je systém sestaven, aby splňoval veškeré požadavky, které byly závěrem analýzy. Hlavní fází je poté samotná implementace za pomoci programovacího jazyka PHP, značkovacího jazyka HTML, kaskádových stylů a souvisejících webových technologií vytvořen finální informační systém.

Výsledná aplikace je otestována několika sérií různých testů, kde se ověřuje kvalita zabezpečení webového serveru, rychlost a odezva aplikace. V závěrečném kroku je takto otestovaná webová aplikace nasazena na webový server.

Klíčová slova: informace, systém, informační systém, diagram, UML, HTML, CSS, bootstrap, JQUERY, PHP, MYSQL, databáze, relační databáze

Information System for Medical Center

Abstract

The theoretical part of the theses defines and explains terms to development of an information system. Next in this section is define unified modelling language, which is used for design of software. Last theoretical section is about front end technologies used to development web application as object-oriented programming language PHP, hypertext mark-up language etc.

In the practical part of the thesis is created logical diagrams of the website information system based on analytic requirement. Main part of this theses is implementation section, using the PHP programming language and technologies for website development created final information system.

This system is subjected to some verify test for check quality, speed and security of system. Finally, this created system is deployed to a webserver.

Keywords: information, system, information system, diagram, UML, HTML, CSS, bootstrap, JQUERY, PHP, MYSQL, relation database

Obsah

1 Úvod.....	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Informace a systém	13
3.1.1 Typy informačních systémů.....	15
3.1.2 Projektování informačního systému	15
3.1.3 Typy CASE nástrojů.....	17
3.1.4 Životní cyklus vývoje informačního systému.....	17
3.1.4.1 Model vodopád.....	18
3.1.4.2 Model spirála	19
3.1.4.3 Prototypový model	19
3.1.4.4 Agile	20
3.2 Jazyk UML.....	21
3.2.1 Elementy UML	22
3.2.1.1 Předměty.....	22
3.2.1.2 Vztahy	23
3.2.1.3 Diagramy	24
3.2.1.4 Diagram tříd	25
3.2.1.5 Diagram aktivit.....	26
3.2.1.6 Stavový diagram.....	27
3.2.1.7 Entitně relační diagram.....	27
3.2.1.8 Use Case diagram	28
3.3 HTML Jazyk	29
3.3.1 Historie HTML	29
3.3.1.1 Počátky	29
3.3.1.2 První specifikace	30
3.3.1.3 Přehled základních elementů používaných v jazyce HTML.....	31
3.3.1.4 Nové tagy HTML5	31
3.4 Kaskádové styly	32
3.4.1 Syntaxe	32

3.4.2	Propojení kaskádových stylů s dokumentem HTML	33
4	Vlastní práce	34
4.1	Analýza informačního systému.....	34
4.1.1	Role systému.....	35
4.1.2	Diagram případů užití	35
4.1.3	Activity diagram	40
4.1.4	Class diagram.....	42
4.1.5	Sekvenční diagram.....	43
4.2	Návrh informačního systému	44
4.2.1	Datový slovník.....	44
4.2.2	Entity relationship diagram.....	46
4.2.3	Wireframe	46
4.3	Implementace	48
4.3.1	Vytvoření struktury databáze.....	49
4.3.2	Struktura aplikace	50
4.3.3	Připojení do databáze.....	51
4.3.4	Rozdělení uživatelských rolí.....	51
4.3.5	Předávání informací o přihlášeném uživateli.....	52
4.3.6	Úvodní strana, přihlašovací obrazovka a registrace	53
4.3.6.1	reCaptcha	62
4.3.6.2	Ověření emailové adresy	62
4.3.6.3	Souhlas se zpracováním osobních údajů	63
4.3.7	Rozhraní pacienta	64
4.3.7.1	Vytvoření nové rezervace.....	65
4.3.7.2	Funkce timeslots()	70
4.3.8	Rozhraní lékaře	71
4.3.9	Rozhraní administrátora.....	73
4.3.10	Nastavení osobních údajů	76
4.4	Testování	78
4.4.1	Zabezpečení	78
4.4.2	Test rychlosti	78
5	Výsledky a diskuse	80
6	Závěr	81
7	Seznam použitých zdrojů	82
8	Seznam obrázků	85

Seznam použitých zkratk

HTML – Značkovací jazyk pro webové stránky. (Hyper Text Markup Language)

CSS – Kaskádové styly. Formátovací jazyk určený pro popis způsobu jakým se zobrazují stránky napsané v jazycích HTML, XHTML, XML.

W3C – Konsorcium založené v roce 1994. Vydává směrnice a specifikace, které podporují rozvoj webu.

WWW – Systém vzájemně propojených webových stránek a dokumentů.

XML – Rozšiřitelný značkovací jazyk, který byl vyvinutý a standardizovaný konsorciem (eXtensible Markup Language)

HTTP – Protokol pro přenos html dokumentů mezi serverem a klientem služby WWW. (Hypertext Transfer Protocol)

1 Úvod

Informační technologie v současné době ovlivňují veškeré součásti každodenního života velkého množství obyvatel. Internet nám dává velké možnosti v usnadnění každodenních činností či práce. Trendem této doby je elektronizace a digitalizace většiny dostupných materiálů a zdrojů. Je až velkým překvapením, že digitalizace většiny agend ve zdravotnictví postupuje velmi pomalými kroky, a většinou velmi nesystematicky, kdy místo jednotného a jednoduchého systému vznikají složité a nespolutracující programy, které se do funkčního celku budou sladovat velmi komplikovaně. Většina soukromých ordinací nedisponuje propracovanějšími informačními systémy, které by lékařům i pacientům zjednodušili a zefektivnili každodenní činnosti. Ordinance většinou neposkytují rezervační systémy a pacienti se musí spolehnout pouze na telefonické objednání.

Teoretická část této diplomové práce se zabývá základními pojmy vývoje informačního systému, jaké jsou metodiky vývoje informačních systémů jejich hlavní přednosti či nevýhody. Dále jsou zde popsány možnosti modelovacího jazyka UML a nejpoužívanější diagramy, které jsou pro vývoj informačního systému důležité. Závěrem teoretické části, je představení možnosti značkovacího jazyka HTML používaného k tvorbě webových prezentací. V praktické části autor vytváří vlastní informační systém propojující základní moduly správy pacientů, možnosti vytvářet rezervace na vyšetření apod.

Motivací pro vývoj tohoto typu informačního systému je v této digitální době nedostatečné využití moderních technologií, výsledkem by mělo být zefektivnění administrativní práce lékařů, s možností vzdáleného přístupu ke kartám pacientů. Zpříjemnění pro pacienty v jednodušším přístupu ke svému lékaři, s možností kontaktování a objednávání.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavním cílem diplomové práce je vytvoření rezervačního informačního systému pro nestátní zdravotnické zařízení se zaměřením na sportovní medicínu. Informační systém bude umožňovat pacientům vytvoření rezervace na daný volný termín, lékaři umožní tyto rezervace schvalovat či zamítnout. Dílčím cílem je vytvoření analýzy a srovnání se současným stavem. Pro návrh systému bude využit jazyk UML.

2.2 Metodika

Teoretická část diplomové práce bude založena na studiu odborné literatury. Bude popsána problematika návrhu informačních systémů a nástroje použité pro jeho návrh a vývoj.

Praktická část diplomové práce se bude zabývat analýzou potřeb a návrhem informačního systému pro zdravotnické zařízení. Návrh bude zpracován pomocí modelovacího jazyka UML. Podle vytvořeného návrhu proběhne následná implementace samotného informačního systému.

Samotná implementace rezervačního systému bude provedena pomocí objektivě orientovaného programovacího jazyka PHP, pro uchování dat a přístup k nim je využita databáze MySQL, ke které bude zřízen přístup do administrace za využití PhpMyAdmin. Samotný webový portál bude vytvořen pomocí značkovacího jazyka HTML a stylizován pomocí kaskádových stylů. Pro vyhodnocení kvality bezpečnosti a rychlosti webového portálu bude využit online nástroj SSL Labs.

3 Teoretická východiska

3.1 Informace a systém

Informace jsou v současné době velmi důležitým faktorem úspěšnosti obchodních organizací. Je velmi důležité s nimi pracovat jako s kapitálem organizace, používat je, ale zbytečně s nimi neplýtvat. V případě informací není rozhodující velké množství údajů (informací). Je velmi nutné respektovat jejich význam. Běžně se stává, že zpráva s minimálním množstvím kvalitních informací může být významově hodnotnější. Počítače a další prostředky informačních technologií nám umožňují zpracovávat údaje a vytvářet vhodné informace na rozhodování. (2)

Informace v dnešní době řadíme vedle materiálních, energetických a finančních zdrojů k hlavním faktorům podmiňujících pokrok ve většině oborů lidské činnosti. Pro dobrou funkci jakýchkoliv aktivit je potřebné, aby existovala včasná korekce odchylek mezi stanovenými cíli a jejich realizací. K tomu je ale potřebné (1):

- Rychle a operativně se rozhodnout o daném problému
- Pružně a efektivně získávat, přenášet a zpracovávat informace
- Neustále zdokonalovat celý informační systém organizace, firmy

„Informační systém je soubor lidí, technických prostředků a metod zabezpečujících sběr, přenos, zpracovávání a uchovávání dat za účelem tvorby a prezentace informací podle potřeby uživatelů.“ (1)

V současnosti lze v běžném životě nalézt velké množství definic pojmu systém a informační systém. Každý z autorů systém definuje různě. Jednotliví autoři systém definují takto (3):

- systém je komplex prvků, nacházejících se ve vzájemné interakci,
- systém je množina objektů, jejich vztahů mezi nimi a jejich atributy,
- systém je organizované množství prvků, které jsou vzájemně provázané a plní různé funkce,
- systém je množství předmětů, jevů, dějů a poznatků, které spolu souvisí přesně vymezeným způsobem.

Definicí pojmu systém je velké množství (viz výše), ale všechny mají společné znaky (3):

- komplex vzájemně propojených prvků,
- systém vyjadřuje zvláštní jednotu s okolím,
- prvek systému může být současně systémem nižšího řádu,
- systém může být současně prvkem systému vyššího řádu.

U informačních systému, stejně jako u jiných druhů systému rozlišujeme dvě základní vlastnosti (1):

- Struktura systému – tou rozumíme jednak uspořádání, organizace jednotlivých prvků systému, ale také uspořádání vazeb mezi těmito prvky.
- Chování (fungování) systému – o tom lze hovořit až ve chvíli, kdy známe jednotlivé vstupy do systému s příslušnými podněty na jedné straně a výstup systému s odpovídající odezvou na straně druhé.

Informační systém z hlediska jednotlivých komponent lze rozdělit do následujících kategorií (4):

- Technické vybavení (hardware) – doplnění systému o periferní jednotky
- Programové vybavení (software) – tvoří systémové služby, které řídí chod počítače, efektivní práci s daty
- Datové zdroje – využití pro práci systému
- Organizační prostředky – obsahují pravidla a nařízení, kterými definují IS
- Lidská složka – řeší adaptaci člověka v počítačovém prostředí
- Reálný svět – kontext informačního systému

Cílem informačního systému firmy je efektivní podpora na úrovni informačních a rozhodovacích procesů v rámci celé organizace (firmy). Informační systém se skládá z řady na sebe nezávislých aplikací (informatických služeb), které jsou v rámci informačního systému implementovány do jednoho projektu (e-mail, správa majetku, účetnictví)

Informatická služba je relativně malá část informačního systému, viditelná koncovému uživateli. Jejím účelem je podpora procesů uvnitř organizace. (4)

3.1.1 Typy informačních systémů

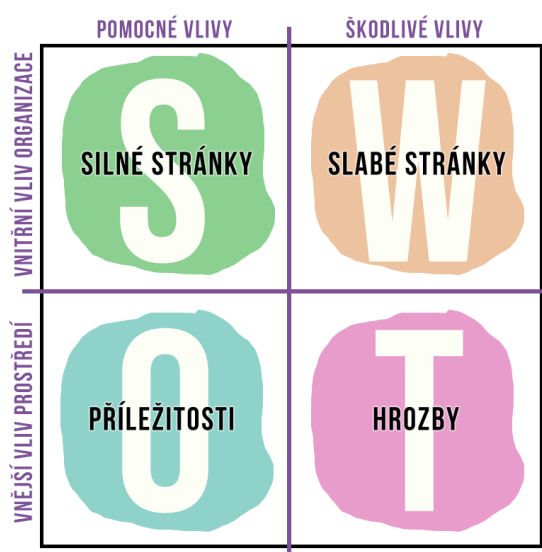
Účelem informačního systému je zajištění správných informací na správném místě a ve správném čase. Informační systémy je možné rozdělit do následujících skupin (6):

- **crm** – systémy pro řízení vztahů se zákazníky
- **scm** – systémy pro řízení dodavatelsko-odběratelských řetězců
- **erp** – systémy pro řízení podnikových zdrojů
- **dbms** – systémy pro řízení databází

3.1.2 Projektování informačního systému

Projektováním informačního systému se myslí proces tvorby nového SW a jeho následné zavedení do provozu ve společnosti. Proces je řízen určitými pravidly, kterými by se při vývoji mělo řídit, tomuto procesu se říká metodika. Metodika udává kdo, kdy, co a proč bude během vývoje IS dělat. (4)

Příkladem metodiky může být například iterativní model vývoje SW, případně agilní extrémní programování, Scrum atd. Udává nám, co by se v danou chvíli mělo dělat, kdo je za danou část odpovědný. Metodika také říká, jakým způsobem se dostat daného cíle, příkladem může být SWOT analýza (4), neboli systematická analýza, která usnadňuje porovnání vnějších hrozeb a příležitostí s vnitřními silnými a slabými stránkami organizace, týmu či projektu. (5)



Obrázek 1 - SWOT analýza (Zdroj:<https://zsf.cz/show/swot-analyza-rychleho-nasazeni>)

Při vývoji informačního systému lze využít podpůrných nástrojů, příkladem může být např. CASE (Computer aided software engineering) neboli počítačem podporované softwarové inženýrství. Nástrojů se využívá při vývoji či údržbě počítačových programů, hlavním účelem je dosažení vyšší kvality, bezchybnosti, udržitelnosti apod. (6)

Některé obvyklé funkce CASE nástrojů (4):

- Generování zdrojového kódu (reverse engineering)
- Datové modelování
- Objektově orientovaná analýza
- Design v UML, BPMN atd.
- Refaktoring

CASE nástroje jsou v mnohých případech nedílnou součástí při vývoji softwaru. Umožňují modelování IT systémů pomocí diagramů, propojení jednotlivých modelů a zdrojového kódu, generování zdrojového kódu z modelu a v neposlední řadě zpětné vytvoření modelu z již existujícího zdrojového kódu. Nástroje jsou uzpůsobené pro podporu týmové práce, správu vývoje a v neposlední řadě pro podporu nových programovacích technik. (7)

3.1.3 Typy CASE nástrojů

Existuje velké množství CASE nástrojů. Je to dané nejen podporovanou metodikou, ale taktéž tím, v jaké fázi vývoje se nástroje používají. Využívají se primárně ve fázích analýzy, návrhu, kódování a údržby. (7)

Podle životního cyklu vývoje software je možné CASE rozdělit do následujících skupin (7):

- **Pre CASE** – jsou určené k tvorbě globální strategie
- **Upper CASE** – podporují plánování, specifikaci požadavků, modelování organizace podniku, globální analýzu informačního systému
- **Middle CASE** – podporují podrobnou specifikaci požadavků a vlastní návrh informačního systému
- **Lower CASE** – obsahují nástroje na podporu testování, údržby a reverzního inženýrství
- **Post CASE** – zajišťují organizační činnosti, zavedení, údržbu a rozvoj informačního systému

3.1.4 Životní cyklus vývoje informačního systému

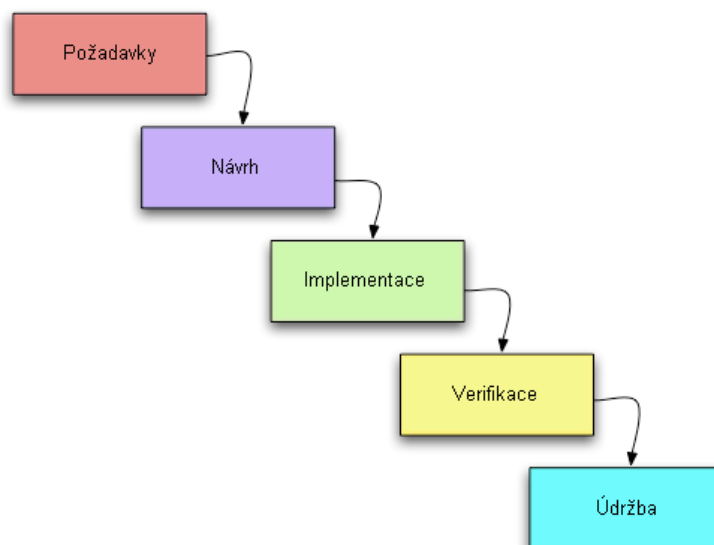
Životním cyklem vývoje informačního systému se rozumí posloupnost jednotlivých fází tvorby IS od úvodní analýzy až po ukončení provozu. Jsou to po sobě jdoucí etapy (7, 8):

- **Předběžná analýza, specifikace cílů** – Prvotní etapou je sběr požadavků uživatelů a určení cílů organizace, v této fázi se v hrubých rysech jednotlivé požadavky rozebírají a odhaduje se přibližná doba realizace a náklady na realizaci.
- **Analýza systému, specifikace požadavků** – hlavní analýza funkčnosti informačního systému jako celku, případné napojení na další informační systémy organizace, přesné určení časové náročnosti a nákladů, klíčová fáze projektu z hlediska struktury dat i systému
- **Návrh, projektová studie** – tato fáze je výsledkem analýzy systému, výsledkem je dokument (smlouva) o návrhu IS, časový harmonogram, cena, konkrétní vyvíjené moduly IS.
- **Implementace** – vlastní programování informačního systému. V této fázi jsou zainteresováni programátoři a analytik, který je zodpovědný za správnost řešení.

- **Testování** – V této fázi se provádí připravené testy na již hotovém informačním systému, v této fázi je nutno vyzkoušet veškeré možné reakce systému, zjištěné nedostatky je potřeba před zavedením systému odstranit.
- **Zavádění systému** – instalace do provozu organizace, poskytnutí manuálů případně školení uživatelům
- **Zkušební provoz** – doba, ve které je poskytovatel povinen zajistit okamžité odstranění chyb, zjištěných během provozu, dořešení dodatečných požadavků
- **Provoz a údržba** – závěrečná fáze informačního systému, systém je aktivně používán, poskytovatel zajišťuje v domluvené kvalitě servis a údržbu.
- **Reengineering** – přehodnocení požadavků na systém, pokud nelze pouhou úpravou systému, je to krok vedoucí na počátek životního cyklu.

3.1.4.1 Model vodopád

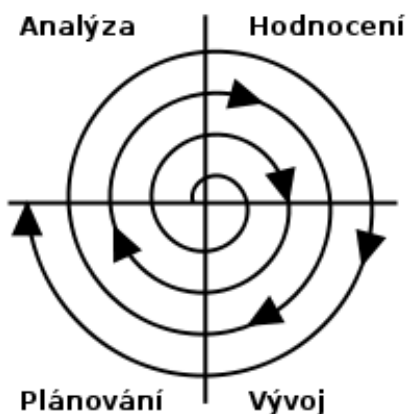
Vodopádový model je založený na sekvenčním přístupu k vývoji software, je vnímán jako tok činností, které na sebe jednotlivě navazují. Prvním formálním popisem vodopádového modelu byl článek publikovaný Winstonem Royce v roce 1970. Projekt je rozložen do několika fází, které se mohou překrývat či prolínat. Důraz se klade primárně na plánování, časové rozvrhy, rozpočet a finální implementaci celého projektu. Striktní model zapovídá možnost se do jakékoliv fáze vrátit a provést revizi, jakmile je fáze jednou dokončena. (21, 22)



Obrázek 2 - Model vodopád Zdroj: https://cs.wikipedia.org/wiki/Vodopádový_model

3.1.4.2 Model spirála

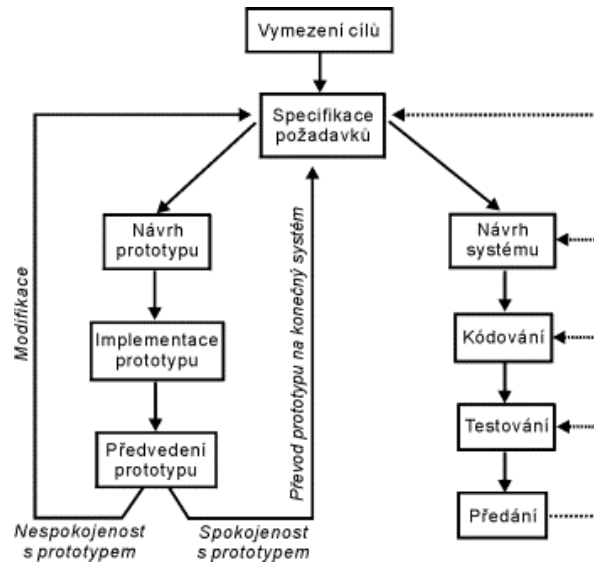
Spirálový model byl publikován roku 1988 Barrym Boehmem, je to model, který kombinuje některé důležité součásti vodopádového modelu a metodiky RAD. Cílem bylo zkombinovat výhody konceptů shora dolů, a zdola nahorů. Model poskytuje jinými metodikami přehlížené oblasti, například iterativní analýza rizik. Základním principem spirálového modelu je zaměření na vyhodnocení a minimalizaci rizik projektu rozděleného na menší části, a snaží změny v průběhu vývojového procesu. (8)



Obrázek 3 - Model spirála - Zdroj: https://cs.wikipedia.org/wiki/Metodika_v%C3%BDvoje_softwaru

3.1.4.3 Prototypový model

Prototypový model (prototypování) je založeno na vytváření částečných verzí systému. Je to zhmotnění určitého návrhu z myšlenek do reality a jeho následné dotažení do takové podoby, která bude funkční. Při prototypování se připravuje přenesení nápadu do fyzické formy, lze realizovat i takovým způsobem, při němž se nemusí do procesu investovat příliš mnoho času a peněz. Usnadňuje komunikaci a vymezení požadavků se zadavatelem, zadavatel má představu o reálném návrhu systému před samotným započítím vývoje (lze specifikovat další požadavky) (8)



Obrázek 4 - Prototypový model - Zdroj: <https://www.fi.muni.cz/~smid/mis-zivcyk.htm>

3.1.4.4 Agile

V případě agilní metodiky vývoje softwaru je dodavatel s klientem ve velmi těsném provázaném vztahu a fungují v rámci jednoho společného týmu. Jednotliví členové jsou v každodenním kontaktu při řešení jednotlivých úkolů. Pro spolupráci se využívají online nástroje případně vzájemná setkání. Díky tomu má zadavatel přehled o veškeré činnosti a postupu prací a v neposlední řadě kvalitě produktu. Na straně druhé tvůrci produktu mají rychlou zpětnou vazbu od klienta, což umožňuje efektivní řízení očekávání. (Nerozchází se očekávání business a IT) (23)

- **Extrémní programování** – mezi základní praktiky extrémního programování patří: **Plánovací hra:** zákazník a vývojářský tým si společně určí, jak dosáhnout maximálního ekonomického přínosu za co nejmenší čas. **Párové programování:** na produkčním kódu vždy pracují dva programátoři, společně pracují nad jedním zdrojovým kódem. **Vydávání malých verzí:** dle EP je potřeba vydávat novou verzi tak často, jak jen to je možno. Mezi další praktiky patří např. testování, refaktoring, zákazník na pracovišti apod. (24, 25)
- **Scrum** – je hojně využívanou agilní metodou, kdy je postavený na úzké spolupráci mezi všemi členy týmu, kteří spolu každodenně komunikují (osobně, online), a vzájemně se informují o odvedené práci, a plánech na aktuální den. Práce na projektu je rozdělena do jednotlivých časových úseků (sprintů), ve kterých má být

vyvinuta předem stanovená část aplikace. Délka sprintu se odvíjí od zvyklosti konkrétního týmu a povahy projektu, většinou trvá v rozmezí 1-4 týdny. (26)



Obrázek 5 - Schéma agilního vývoje <Zdroj> <https://m.systemonline.cz/it-pro-banky-a-financni-organizace/zkusenosti-ceskych-bank-s-agilnim-vyvojem.htm>

3.2 Jazyk UML

Unified Modeling Language (UML) je grafický jazyk, který je úředně definovaný skupinou Object Management Group (OMG) a je určený pro vizualizaci, specifikaci, konstrukci objektů softwarového systému. Tento jazyk nabízí standardní způsob psaní plánů systému jako např. obchodní procesy a systémové funkce a zároveň konkrétní problémy jako například příkazy programovacího jazyka, schéma databáze a opakovatelné použití softwarové komponenty. UML se běžně využívá na modelování podnikových procesů, modelování systémového inženýrství a reprezentování organizační struktury. (15)

Od svého počátku prošlo UML dvěma hlavními a vícero drobnými revizemi verze, po dobu, kterou se definice UML dále vyvíjela. Specifikace UML verze 2.5.1 definuje jazyk následovně: „UML je vizuální jazyk na specifikaci, konstrukci a dokumentaci objektů systému“ Je to univerzální modelovací jazyk, který je možné používat se všemi hlavními metodami objektů a komponent, které je možno aplikovat v různých sférách podniků. (15)

Unified Modeling Language se stal široce akceptovaným standardem pro objektivně orientovanou analýzu a design krátce poté co byl poprvé standardizován a platí dodnes.

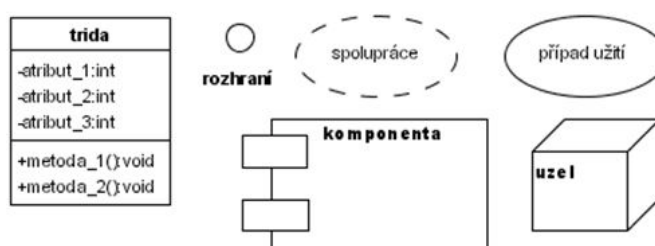
3.2.1 Elementy UML

Stavební bloky jsou základní elementy modelování v UML. Jazyk UML obsahuje tři stavební bloky, které jsou konkrétně popsány níže. (15, 16)

3.2.1.1 Předměty

Předměty (*Things*) – představují samotné modelovací prvky. Dělí se dále do následujících čtyřech kategorií: (15, 16)

- **Strukturální abstrakce** – Jsou v UML modelu jména (např. rozhraní, třída, případ použití, uzel)



Obrázek 6 - příklad strukturální abstrakce <Zdroj> www.slideplayer.cz/slide/2938240

- **Předmět chování** – reprezentuje dynamické chování objektu



Obrázek 7 - příklad předmětu chování <Zdroj> www.slideplayer.cz/slide/2938240

- **Předmět seskupení** – představuje organizační složky modelu v podobě balíčku, do kterého se seskupují související prvky



Obrázek 8 - příklad seskupení <Zdroj> www.slideplayer.cz/slide/2938240

- **Poznámky** – se používají pro komentáře pro bližší vysvětlení části modelu.

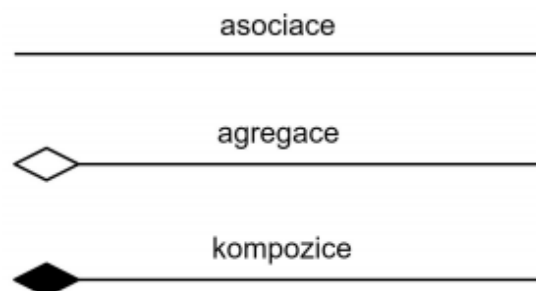


Obrázek 9 - příklad poznámky <Zdroj> www.slideplayer.cz/slide/2938240

3.2.1.2 Vztahy

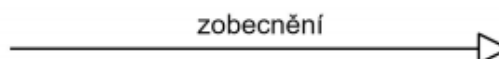
Sémantickou vazbu mezi dvěma a více předměty umožňují vyjádřit vztahy (*relationships*). Rozlišujeme vícero druhů vztahů, které se zobrazují podle různých typů čar. Mezi nejčastější druhy vztahů patří: (15, 16)

- **Asociace** – definují vztah mezi dvěma předměty. Upřesnit je lze pomocí dvou speciálních čar, prvním typem jsou takzvané **Agregace**, kdy oba předměty mohou existovat nezávisle na sobě. Druhým typem jsou **Kompozice**, kdy cílový předmět nemůže existovat bez celku.



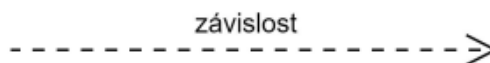
Obrázek 10 – Vztah asociace, agregace, kompozice <Zdroj> www.slideplayer.cz/slide/2938240

- **Zobecnění** – používá se v případě kdy předmět, z kterého vztah vychází je specializovaným případem cílového předmětu a může být kdykoliv nahrazen.



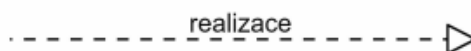
Obrázek 11 - Vztah zobecnění <Zdroj> www.slideplayer.cz/slide/2938240

- **Závislost** – vyjadřuje vztah mezi dvěma předměty tak, že závislý předmět využívá závislou část jiného předmětu.



Obrázek 12 - Vztah závislosti <Zdroj> www.slideplayer.cz/slide/2938240

- **Realizace** – zdrojový předmět zaručuje, že splní všechny podmínky dohody v cílovém předmětu.



Obrázek 13 - Vztah realizace <Zdroj> www.slideplayer.cz/slide/2938240

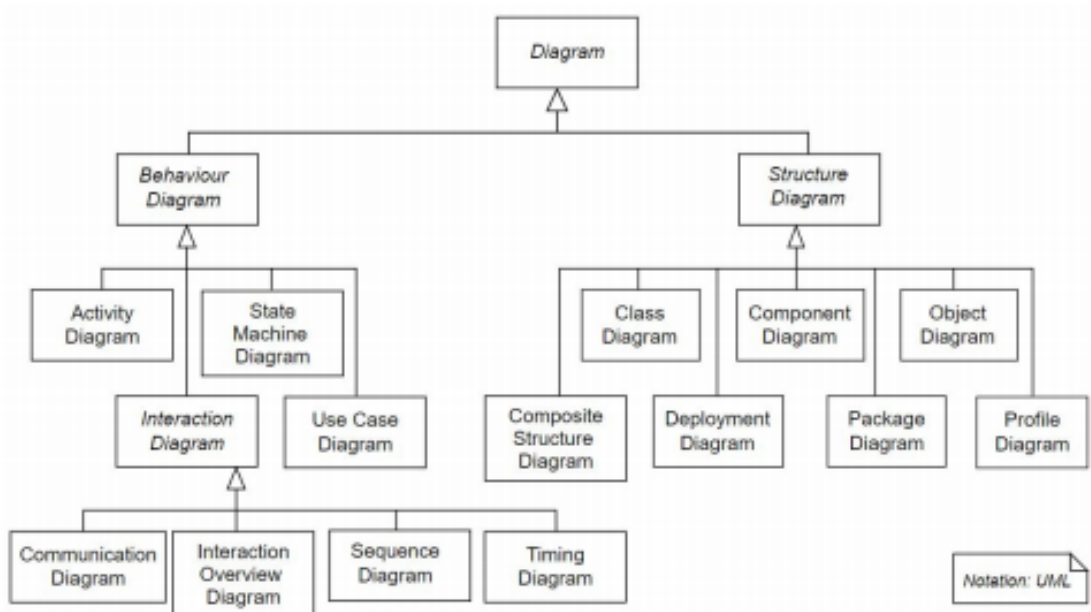
3.2.1.3 Diagramy

Posledním stavebním blokem Unified Modeling Language se nazývají diagramy. Diagramy jsou nosnou částí tohoto grafického jazyka, protože vizuálně vyjadřují samotný obsah.

Systém je možno specifikovat z různých pohledů na základě toho, jaký diagram se v UML nakreslí. V kontextu vývoje existuje pět důležitých pohledů, které jsou důležité pro vizualizaci, specifikaci, konstrukci, dokumentaci softwarové architektury (15, 16)

- pohled uživatele
- designový pohled
- interaktivní pohled
- realizační pohled
- pohled nasazení

Každý z těchto pohledů v sobě obsahuje dva typy modelování. Modelování strukturální, které je považované za statický aspekt systému. A behaviorální modelování, které představuje dynamický aspekt systému. Diagram lze definovat jako grafickou prezentaci skupiny prvků, které jsou nejčastěji vyjádřeny jako graf předmětů a vztahů mezi nimi. (17, 18)



**Obrázek 14 - Přehled všech UML diagramů a jejich hierarchie <Zdroj>
https://commons.wikimedia.org/wiki/File:UML_diagrams_overview.svg**

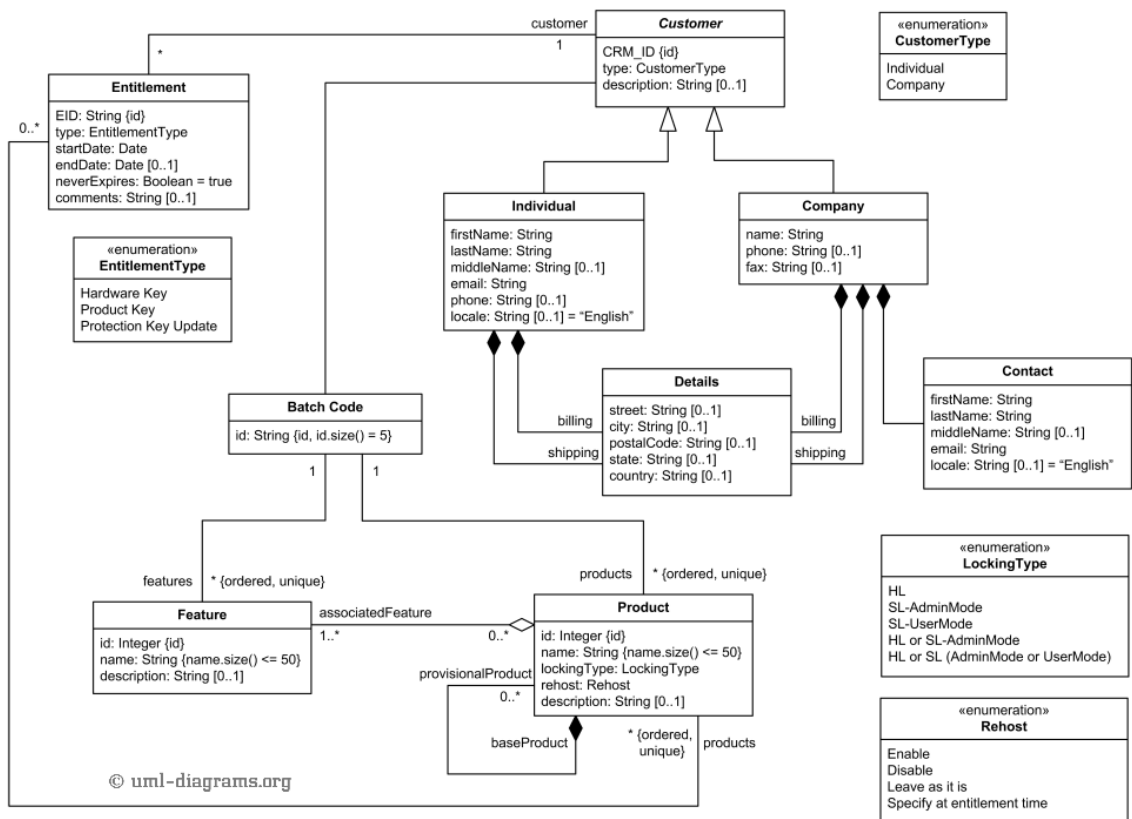
První specifikace UML (verze 1.1) obsahovala pouze devět základních typů diagramů.

Jedná se konkrétně o diagramy: (18)

- diagram tříd
- diagram komponent
- objektový diagram
- diagram nasazení
- diagram aktivit
- diagram případů použití
- diagram stavového automatu
- diagram posloupnosti
- diagram spolupráce

3.2.1.4 Diagram tříd

Diagram tříd (Analytic class diagram) je jeden z nejpoužívanějších modelovacích diagramů, který poskytuje statický náhled na základní systémovou strukturu. Tvoří jej třídy (classes), atributy jednotlivých tříd včetně datových typů, operace a vztahy mezi jednotlivými entitami, přičemž nezachází do hlubších detailů v implementaci. (27)

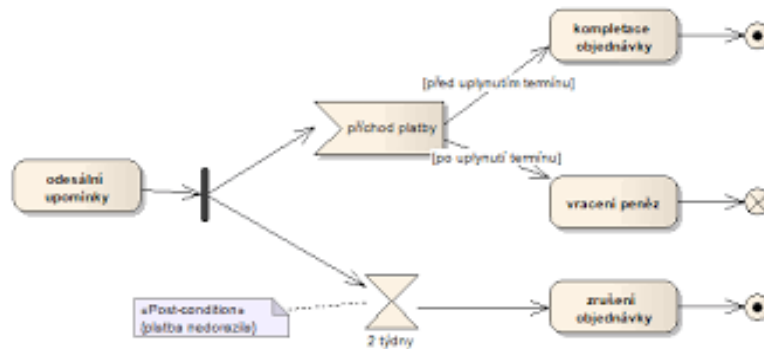


Obrázek 15 - Ukázka diagramu tříd <Zdroj> <https://www.uml-diagrams.org/software-licensing-domain-diagram-example.html>

3.2.1.5 Diagram aktivit

Diagram aktivit (Activity diagram), v tomto diagramu se procesy modelují jako aktivity, které se skládají z uzlů, které jsou spojeny orientovanými hranami. Reprezentuje nejčastěji tok v případech užití, modelování business procesů uvnitř firmy. Diagram lze rozdělit pomocí svislých, vodorovných a zakřivených čar. (15, 16, 28)

- Uzly (Nodes) – akční, řídicí, objektové uzly: představují atomický krok výpočtu.
- Hrany (Edges) – spojnice jednotlivých uzlů, která umožňuje přechod mezi nimi.



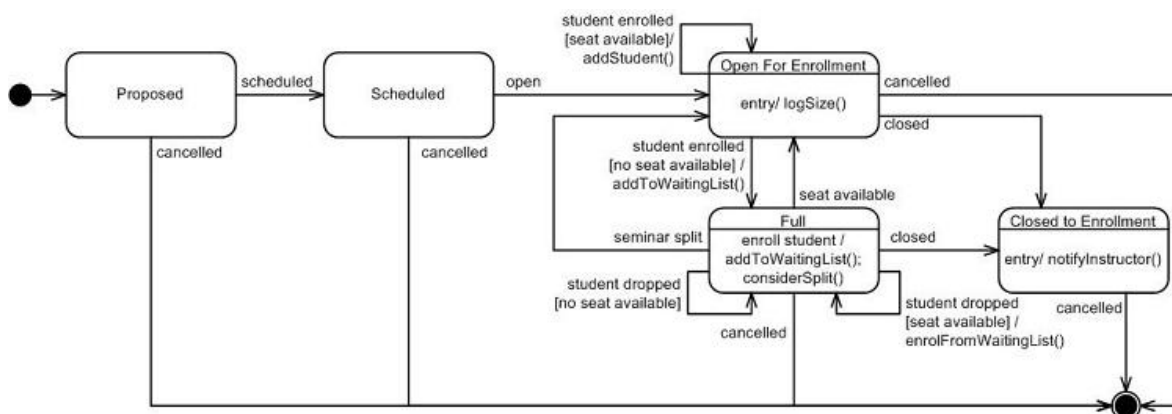
Obrázek 16 - Ukázka diagramu aktivit <Zdroj> https://cs.wikipedia.org/wiki/Diagram_aktivit

3.2.1.6 Stavový diagram

Stavový diagram (State Machine diagram) se využívá pro modelování dynamického chování objektu. Životní cyklus objektu nám zobrazuje jako konečný stavový automat, který v sobě obsahuje konečnou množinu stavů a podle přesně nadefinovaných pravidel na základě jednotlivých reakcí na událost mezi stavy prochází. (15, 16)

Skládá se z (16):

- **Stavy (States)**
- **Přechody (Transitions)**
- **Události (Events)**
- **Pseudostavy (Pseudostates)**
- **Složené stavy (Composite states)**



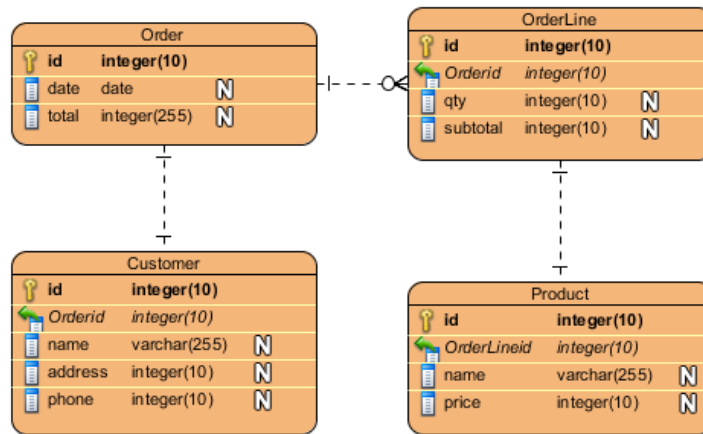
Obrázek 17 - Ukázka State Machine diagramu
<Zdroj><http://agilemodeling.com/artifacts/stateMachineDiagram.htm>

3.2.1.7 Entitně relační diagram

Entitně relační diagram který nám v grafické podobě zobrazuje datovou a logickou strukturu databáze. Zobrazuje v relacích (vztahy mezi objekty) entity a následně data uložená v jednotlivých objektech (atributy) (29)

- **Entity (Entities)** – objekt, který je schopný samostatně existovat a lze ho jednoznačně na základě předem daných pravidel identifikovat. Objekt si lze představit jako věc z reálného světa (např. auto, osoba...), či jako abstraktní pojem (kterým může být třeba objednávka)
- **Relace (Relationships)** – Mezi jednotlivými entitami mohou vzniknout souvislosti (relace), které se modelují pomocí čáry spojující dané entity. Relace má čtyři hlavní parametry: název, stupeň, kardinalitu, členství.

- **Atributy (Attributes)** – Atributy jednotlivých relací

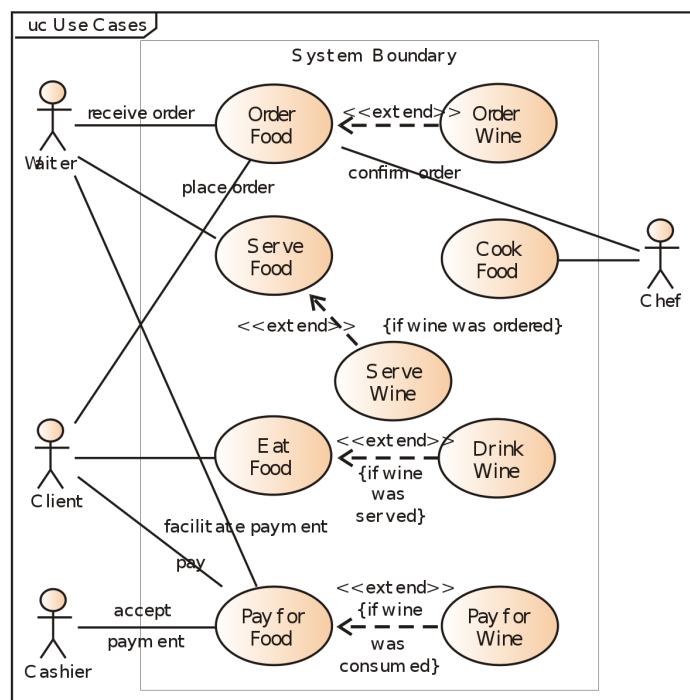


Obrázek 18 - Ukázka ERD diagramu <Zdroj><https://www.visual-paradigm.com/VPGallery/datamodeling/EntityRelationshipDiagram.html>

3.2.1.8 Use Case diagram

Diagram případů užití (Use Case diagram) nám v grafickém zobrazuje funkční požadavky na to co systém na dělat. Díky jednoduché interpretaci je to výhodný diagram pro komunikaci se zákazníky. Vytváří se většinou na začátku vývojového cyklu jako jeden z prvních diagramů. Tvoří ho čtyři hlavní komponenty (16, 30):

- **hranice systému** – udává nám kde leží hranice systému, tudíž co je a co v systému není obsaženo. Zobrazení pomocí obdélníku označeným názvem systému.
- **aktéři** – jakákoliv externí entita, která se systémem přímo komunikuje, využití při testování uživatelského chování.
- **případy užití** – reprezentují jednu činnost systému, kterou zpravidla vyvolá aktér. V PU je popsáno, jak se systém na určité činnosti aktéra zachová.
- **vztahy** – nám zobrazuje propojení mezi aktérem a PU, mezi aktéry i propojení více případů užití. (Vztahy: asociace, generalizace, <<include>>, <<extend>>)



Obrázek 19 - ukázka Use Case diagramu [online](https://en.wikipedia.org/wiki/Use_case_diagram) Zdroj: https://en.wikipedia.org/wiki/Use_case_diagram

3.3 HTML Jazyk

HTML, je zkratka pro HyperText Markup Language, je to hlavní značovací jazyk používaný pro tvorbu webových stránek. Poskytuje prostředky pro vytváření strukturovaných dokumentů sémanticky rozdělených na nadpisy, odstavce, odkazy, citace a další položky. Umožňuje do stránek přidávat obrázky a další objekty, a může být použit pro vytváření interaktivních formulářů. HTML jazyk je napsaný formou HTML elementů které se skládají ze značek, které jsou obklopené lomnými závorkami, mezi kterými se nachází obsah webové stránky. Může obsahovat skripty napsané např. v jazyce Java Script a kaskádové styly (CSS), které definují vzhled stránky, uspořádání textu a dalších objektů. (9, 10, 20)

3.3.1 Historie HTML

3.3.1.1 Počátky

V roce 1980 fyzik Tim Berners-Lee navrhl prototyp ENQUIRE, systém určený pro pracovníky a vědce v CERNu, na používání a sdílení dokumentů. V roce 1989 Berners-Lee napsal dokument navrhuující internet založený na hypertextovém systému (W3C, 1990). Berners-Lee specifikoval HTML a vytvořil software pro prohlížeče a

servery. V tomto roce Berners-Lee a datový inženýr v CERNU Robert Cailliau spolupracovali na společné žádosti o financování, jejich projektu v organizaci nebyl formálně přijatý. (20)

3.3.1.2 První specifikace

První veřejně dostupná specifikace jazyka HTML byl dokument, který se nazýval „HTML tags“. Berners-Lee popisuje 18 prvků, které zahrnují původní relativně jednoduchou konstrukci HTML. Hypertextový značkovací jazyk je takový jazyk, který slouží pro interpretaci textů, obrázků a jiných materiálů. Předvolené hodnoty pro každou HTML položku jsou definované v prohlížeči, a tyto charakteristiky lze změnit či upravit případným použitím CSS. HTML byla formálně definována IETF (Internet Engineering Task Force) v polovině roku 1993 zveřejněním prvního návrhu pro specifikaci HTML jazyka publikací „HyperText Markup Language“ napsanou Berners-Leem a Danom Connollym, který zahrnuje také SGML deklaraci „Document Type Definition“ na definování typu dokumentu. (SGML – vnitropodniková příručka používaných SGML formátů v CERNU). Podobně také Dave Raggett vydal konkurenční publikaci „HTML +“ navrhoval standardizaci jazyka HTML+ s implementací funkcí pro tabulky a formuláře. (20)

Začátkem roku 1994 návrhům HTML a HTML+ vypršela platnost, IETF vytvořil HTML pracovní skupinu, která v roce 1995 napsala „HTML 2.0“, první HTML specifikaci, která je považována jako standard, podle kterého by měla být následná implementace odvozována. Další vývoj pod záštitou IETF byl kvůli konkurenčním zájmům pozastaven. Od roku 1996 byla specifikace HTML udržována pod záštitou W3C – World Wide Web Consortium. V roce 2000 se HTML stal mezinárodním standardem ISO/IEC 15445:2000. HTML 4.01 vyšla na konci roku 1999. V roce 2004 se vývojem nového standardu HTML5 začala zabývat skupina „Web Hypertext Application Technology Working Group“ která se roku 2008 spojila s W3C.

HTML značky se skládají z několika důležitých částí, každá ze značek má přesnou syntaxi. Zapišeme-li značku chybně (např. neexistující značku), prohlížeč ji bude ignorovat a chybu nedetekuje. Pro správnost html kódu se využívá validační nástroj na webových stránkách W3C konsorcia. (20)

Rozlišujeme párové <html>, <body>... značky, které jsou na konci ukončeny ukončovacím tagem (např. </html>) a nepárové tagy např. pro vložení obrázku do stránky. Text mezi <html> a </html> popisuje webovou stránku. Tagy <body> a </body> ohraničují viditelný obsah stránky. Značky a text <title>Název</title> definují název stránky v prohlížeči. (9,10)

Jednotlivé HTML elementy se skládají z počátečního tagu, který může být rozšířen o atribut, dále obsah a ukončovací tag (pokud se jedná o párové značky) (9)

```
<p class="trida">Odstavec</p>
```

p	název elementu
class	jméno atributu
trida	hodnota atributu

3.3.1.3 Přehled základních elementů používaných v jazyce HTML

h1	hlavní nadpis
h2...h6	podnadpis
p	odstavec
br	zalomení řádku
a	odkaz
img	obrázek
ol	číslovaný seznam
ul	seznam s odrážkami
li	položka seznamu
table	tabulka
tr	řádek tabulky
td	buňka tabulky
th	buňka záhlaví tabulky
div	blok

3.3.1.4 Nové tagy HTML5

header	záhlaví webové stránky, článku
nav	navigační menu
main	obsah stránky

article	článek
section	blok stránky, obsahující články určitého zaměření
asside	postranní panel
footer	zápatí stránky

3.4 Kaskádové styly

Kaskádové styly „Cascading Style Sheets“ (CSS) je jazyk pro popis způsobu zobrazení stránek napsaných ve značkovacích jazycích HTML, XHTML případně pro XML dokument. (12)

Jazyk byl navrhnutý standardizační organizací W3C. Autorem prvotního návrhu byl Håkon Wium Lie. Zatím nejnovější dokončená specifikace CSS3 byla vydána v roce 2016. Jde o třetí specifikaci CSS, která přidává např. prvky pro práci s multimédií, nové barevné modely RGBA, HSL, HSLA, podporu zaoblených rohů a stínování textu.

Hlavním smyslem CSS je umožnit vývojáři oddělení vzhledu dokumentu od jeho struktury a obsahu stránek. Původně to měl umožňovat již jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se jazyk vyvinul jinak. (12, 31)

3.4.1 Syntaxe

Tabulka kaskádových stylů se skládá z několika pravidel. Každé pravidlo obsahuje selektor a blok deklarácí. Každý blok deklarácí obsahuje seznam deklarácí a každá deklarace se skládá z vlastností, po které za dvojtečkou je hodnota vlastnosti. Deklarace končí středníkem. Příklad pravidla: (31)

```
body{
    background-color: black;
    color: white;
    padding: 10px;
}
```

Blok „body“ je selektor, v množinových závorkách se nachází blok deklarácí, řádek „background-color: black;“ je deklarace samotná, „background-color“ je identifikátor

vlastnosti, a „black“ je samotná hodnota vlastnosti. Celý kód nastavuje pozadí stránky na černou, barva písma bude bílá a okraj bude mít 10 pixelů.

CSS definuje hodně různých selektorů, které obvykle lze kombinovat. Mezi nejzákladnější patří: (31, 32)

- **body** – tato deklarace bude platit pro všechny výskyty elementu „body“
- **body p** – tato deklarace bude platit pro všechny elementy „p“, které se nachází v elementu „body“
- **body > div** – tato deklarace bude platit pro všechny elementy „div“, které jsou potomky elementu „body“. Například pokud by se element div nacházel uvnitř elementů <body><section><div></div></section> tak by deklarace pro tento případ neplatila, jelikož <div> není přímým potomkem elementu <body>
- **.třída** – tato deklarace bude platit pro všechny elementy, které mají nastavené v HTML třídu na hodnotu „třída“ – atribut „class“
- **#identifikátor** – tato deklarace bude platit pro všechny elementy, které mají v HTML nastavený identifikátor na „identifikátor“ – atribut „id“
- **sel1, sel2, sel3** – pomocí čárek lze jednotlivé selektory seskupovat, tato deklarace bude platit pro všechny selektory.

3.4.2 Propojení kaskádových stylů s dokumentem HTML

Pro připojení kaskádových stylů do HTML dokumentu existuje několik možností. V praxi se nejvíce využívá způsob odkazu na externí soubor (soubor s koncovkou .css)

Připojení externího souboru pomocí HTML elementu <link>, který je v HTML dokumentu součástí hlavičky <head>...</head>

```
<head>  
<link rel='stylesheet' href='<odkaz.css>' type='text/css'>  
</head>
```

Používání kaskádových stylů v porovnání se samotným HTML přináší v praxi výhody jako rozsáhlejší možnosti formátování. Pro formátování bloku textu a určení vzdálenosti od jiných elementů či okrajů stránky HTML žádné možnosti nenabízí. CSS má vlastnosti „padding“ a „margin“. Další výhodou CSS je jednodušší údržba webových stránek. Například pokud chceme změnit nějaký detail na webové stránce, například barvu nadpisu, není potřeba procházet všechny HTML stránky/šablony, ale změníme pouze jednu vlastnost v CSS dokumentu, který je připojený na všechny stránky. (32)

4 Vlastní práce

Centrum zdravého pohybu

Jedná se o menší nestátní zdravotnické zařízení se specializací na sportovní odvětví, zařízení poskytuje svým klientům služby v oblasti fyzioterapie, výživového poradenství, poradenství v životním stylu či sportovní prohlídky se zaměřením na vrcholové sportovce.

V současné době probíhá komunikace pacientů s lékařem případně objednávání jednotlivých vyšetření v obdobných zařízeních pomocí e-mailových zpráv případně telefonicky. Finální agenda je poté ve větší míře stále vedena na papíře. To má za následek velké vytížení personálu v čase, který mohl být věnován přítomnému klientovi. Níže vyvíjený rezervační systém by mohl zjednodušit lékaři tuto agendu a zefektivnit jeho práci.

Pro návrh informačního systému byl využit modelovací jazyk UML (unified modeling language). Diagramy byly modelovány za použití nástroje Virtual Paradigm – Community edition.

4.1 Analýza informačního systému

Analýza a samotný návrh informačního systému popsany v následujících kapitolách se týká systému pro organizování rezervací ve sportovním lékařském zařízení. Klient (pacient) po registraci získá možnost plně využívat klientskou část systému. Bude mít k dispozici přehled jeho již uskutečněných rezervací služeb, či získá možnost správy jeho budoucích rezervací. Systém klientovi umožní vytvořit rezervaci na jednotlivá vyšetření podle aktuálních časových možností lékaře. Z druhé strany bude informační systém lékaře informovat o nově vzniklých rezervacích. Lékař bude mít možnost rezervaci potvrdit, případně ji bude moci zamítnout. Závěrem bude skrz informační systém možno spravovat jednotlivé uživatele, služby a specializace lékaře. Systém bude potřeba navrhnout a implementovat tak aby v budoucnu bylo snadné na systém navázat další funkčnost (komunikační zařízení s pacienty atd.).

Požadavky na systém jsou následující:

Funkční požadavky:

- Registrace klienta
- Možnost registrace nových lékařů
- Vytvoření nové rezervace služby

- Možnost potvrzení či zamítnutí rezervace
- Možnost zrušení již potvrzené rezervace
- Nabídka služeb, které zařízení nabízí
- Správa uživatelů
- Správa služeb
- Správa specializací jednotlivých lékařů

Nefunkční požadavky:

- Využití programovacího jazyka PHP
- Databáze bude zvolena MySQL
- Načtení veškerých komponent systému nebude trvat déle než 1750ms

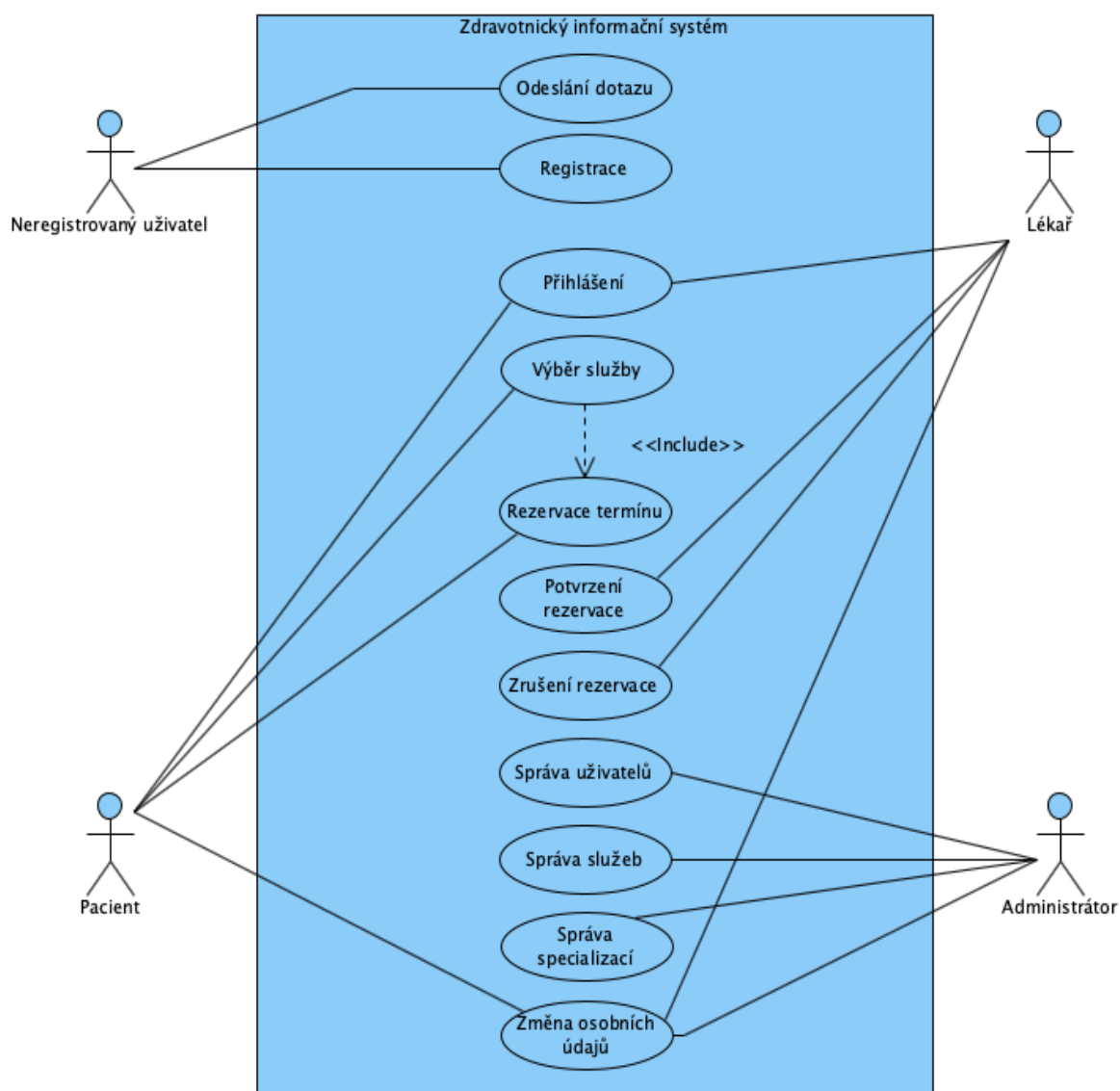
4.1.1 Role systému

Systém se bude skládat ze třech rolí uživatelských účtů. Hlavními účty bude Pacient a Lékař, kteří budou spolu skrz IS komunikovat. Správce systému bude mít k dispozici účet Administrátora.

- **Pacient:** Bude mít možnost vytvářet nové rezervace za účelem objednání služeb zařízení, bude mít přehledné informace o ceně služeb, již proběhlých vyšetření. Bude mu umožněno upravovat své uživatelské údaje uvedené při registraci. Bude mu umožněno podat žádost na odstranění účtu ze systému.
- **Lékař:** Získá po přihlášení do svého uživatelského účtu přehledné informace o vytvořených rezervacích na následující období, bude mu umožněno rezervace potvrzovat či zamítnout. Již potvrzenou rezervaci bude možno ze systému smazat.
- **Administrátor:** Hlavní správce systému získá jako nejnadřazenější role možnost spravovat všechny uživatelské účty, možnost přidat nové uživatele (primárně pro přidání nových účtů lékaře), správu nabízených služeb a správu specializací lékařů.

4.1.2 Diagram případů užití

Diagram případů užití znázorňuje všechny základní procesy, které jednotlivý uživatelé (aktéři) systému mohou využít. Diagram obsahuje celkem čtyři aktéry: Neregistrovaný uživatel, Pacient, Lékař, Administrátor



Obrázek 20 - Zdravotnické zařízení - diagram užití

Pro nejdůležitější aktivity z diagramu užití bude níže uvede scénář (use-case), který slouží pro lepší pochopení konkrétní situace.

Scénáře budou zpracovány podle vzorové šablony viz níže:

Use Case	
Popis	Krátký popis funkcionality
Aktéři	Aktéři spojeni s konkrétním případem užití
Podmínky spuštění	Podmínky, které musí být splněny pro začátek situace
Základní scénář	Posloupnost kroků vedoucím k provedení případu užití
Alternativní scénář	Posloupnost alternativních kroků, které vychází z hlavního scénáře
Podmínky ukončení	Podmínky, které jsou splněny po průchodu scénáře

UC1 – Registrace

Use Case	
Popis	Registrace nového uživatele (pacienta)
Aktéři	Neregistrovaný uživatel
Podmínky spuštění	
Základní scénář	<ol style="list-style-type: none"> 1) Uživatel se přes tlačítko <i>Přihlásit</i> dostane na přihlašovací obrazovku 2) Informační systém načte přihlašovací formulář 3) Uživatel klikne na odkaz <i>Registrovat nového uživatele</i> 4) Systém načte pole pro e-mailovou adresu 5) Uživatel zadá platnou e-mailovou adresu 6) Systém načte registrační formulář 7) Uživatel vyplní registrační formulář a klikne na tlačítko <i>Registrovat</i> 8) Systém uživateli zobrazí přihlašovací stránku a zobrazí hlášku <i>Registrace proběhla úspěšně, byl zaslán email pro validaci</i>
Alternativní scénář	<ol style="list-style-type: none"> 6) Nelze se registrovat, emailová adresa již existuje 8) Nelze se registrovat, špatně zvolené heslo
Podmínky ukončení	Uživatel je úspěšně registrován

UC2 – Přihlášení

Use Case	
Popis	Přihlašování uživatele do systému
Aktéři	Pacient, Lékař, Administrátor
Podmínky spuštění	Uživatelský účet musí být ověřený
Základní scénář	<ol style="list-style-type: none"> 1) Uživatel se přes tlačítko <i>Přihlásit</i> dostane na přihlašovací obrazovku 2) Informační systém načte přihlašovací formulář 3) Uživatel zadá přihlašovací údaje 4) Uživatel klikne na <i>Přihlásit se</i> 5) Systém načte úvodní obrazovku aplikace
Alternativní scénář	<p>5a) Nelze se přihlásit, uživatel s danou emailovou adresou neexistuje</p> <p>5b) Nelze se přihlásit, špatně vyplněné heslo</p>
Podmínky ukončení	Uživatel je úspěšně přihlášen

UC3 – Vytvoření nové rezervace

Use Case	
Popis	Vytvoření nové rezervace pro konkrétní službu
Aktéři	Pacient
Podmínky spuštění	Uživatel musí být úspěšně přihlášený, musí mít nastavenou roli <i>Pacient</i>
Základní scénář	<ol style="list-style-type: none"> 1) Uživatel klikne v menu na položku <i>Rezervace</i> 2) Systém načte obrazovku s rezervačním kalendářem 3) Uživatel vybere z volných termínů kalendáře termín 4) Systém zkontroluje dostupnost volných kapacit v daný termín a zobrazí obrazovku volných časových slotů pro daný den 5) Uživatel vybere požadovaný časový slot 6) Systém načte tabulku <i>Rezervace</i> 7) Uživatel vybere požadovanou službu a klikne na tlačítko <i>Odeslat</i> 8) Systém načte stránku s termíny, zvolený termín se označí jako <i>Rezervovaný</i> a vypíše hlášku <i>Rezervace byla úspěšná</i>

Alternativní scénář	
Podmínky ukončení	Rezervace daného termínu byla úspěšná

UC4 – Schválení/zamítnutí rezervace

Use Case	
Popis	Schválení/Zamítnutí uživatelské rezervace
Aktéři	Lékař
Podmínky spuštění	Uživatel musí být přihlášený, musí mít nastavenou roli <i>Lékař</i> , musí být vytvořena <i>Uživatelem-pacient</i> rezervace
Základní scénář	<ol style="list-style-type: none"> 1) Uživatel klikne na tlačítko <i>Dashboard</i> 2) Systém načte obrazovku hlavní nástěnky a vypíše informace o nepotvrzených a potvrzených rezervacích 3) Uživatel vybere požadovanou rezervaci, kterou chce schválit/zamítnout a klikne na tlačítko <i>Schválit/ resp. Zamítnout</i> 4) Systém provede schválení/zamítnutí rezervace, odešle Uživateli notifikaci o výsledné akci.
Alternativní scénář	2) Systém v databázi nenalezne žádnou nepotvrzenou rezervaci, vypíše hlášku <i>Žádné rezervace k potvrzení</i>
Podmínky ukončení	Rezervace úspěšně potvrzena/zamítnuta

UC5 – Přidání nové služby

Use Case	
Popis	Přidání nové služby do databáze (tabulka cSluzba)
Aktéři	Administrátor
Podmínky spuštění	Uživatel musí být přihlášený, musí mít nastavenou roli <i>Administrátor</i>
Základní scénář	<ol style="list-style-type: none"> 1) Uživatel klikne na tlačítko <i>Dashboard</i> 2) Systém vypíše hlavní stránku nástěnky a zobrazí výpis uživatelů, služeb a specializací 3) Uživatel klikne v bloku <i>Služby</i> na tlačítko <i>Přidat novou službu</i>

	<ul style="list-style-type: none"> 4) Systém načte formulář pro přidání nové služby 5) Uživatel vyplní jméno a z rozbalovacího menu vybere specializaci, ve které se služba bude nabízet. Poté klikne na tlačítko <i>Přidat</i> 6) Systém službu přidá do databáze, uživatele přemístí na úvodní stránku a zobrazí hlášku <i>Služba úspěšně přidána do databáze</i>
Alternativní scénář	6b) Systém zobrazí chybovou hlášku, že daná služba se již v databázi nachází
Podmínky ukončení	Služba je úspěšně přidána do databáze

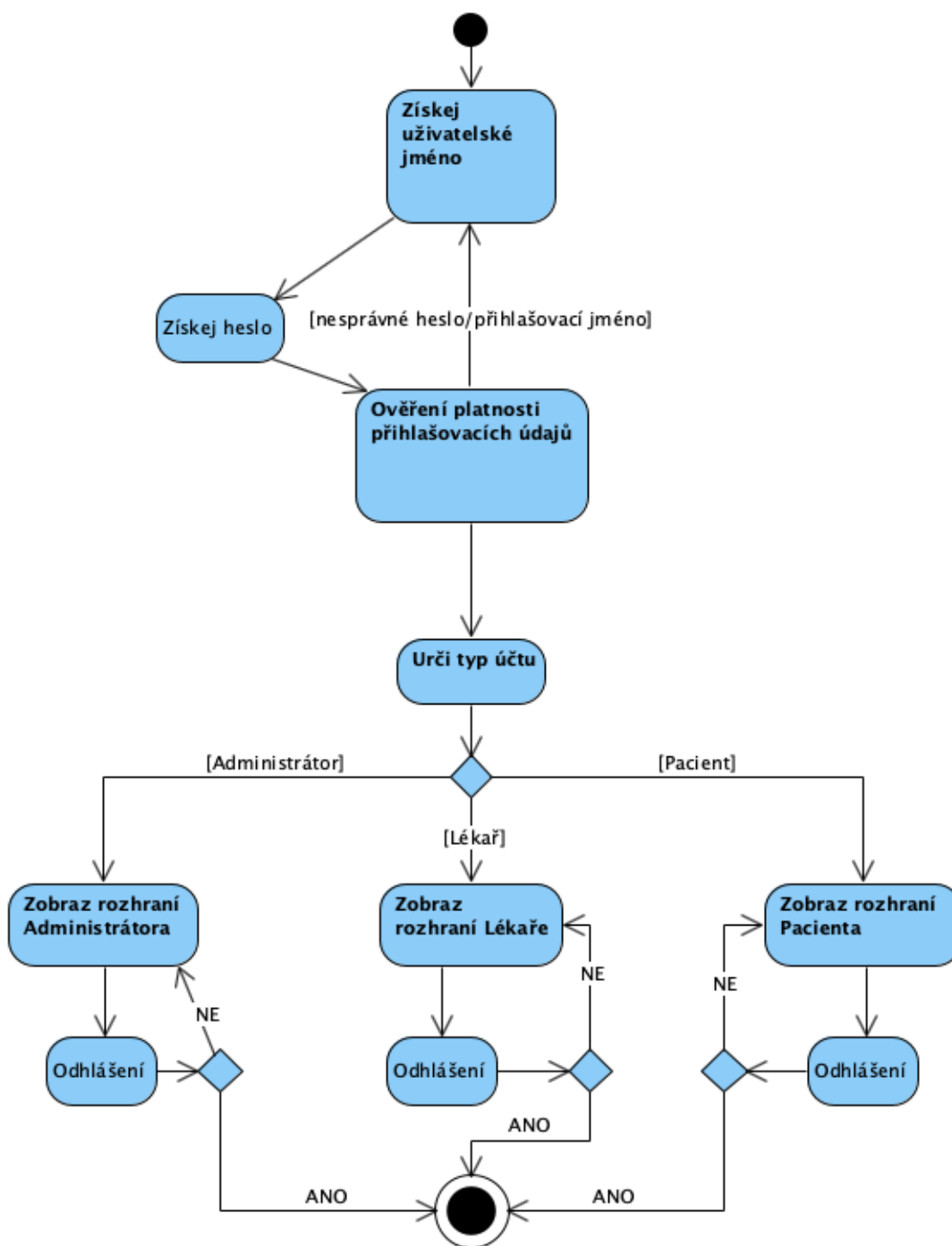
UC6 – Změna osobních údajů

Use Case	
Popis	Změna osobních údajů uživatele
Aktéři	Pacient, Lékař, Administrátor
Podmínky spuštění	Uživatel musí být úspěšně přihlášen
Základní scénář	<ul style="list-style-type: none"> 1) Uživatel spustí informační systém 2) Informační systém se zobrazí 3) Uživatel klikne na záložku <i>Nastavení</i> 4) Systém zobrazí záložku <i>Nastavení</i> s formulářem pro úpravu osobních údajů, načte z databáze již uložené údaje 5) Uživatel změní požadované pole a změnu potvrdí tlačítkem <i>Uložit změny</i> 6) Systém změnu vloží do databáze a uživateli vypíše hlášku <i>Změny úspěšně uloženy</i>
Alternativní scénář	6b) Systém vypíše chybovou hlášku, že nejsou vyplněna všechna povinná pole
Podmínky ukončení	Změny uloženy do databáze

4.1.3 Activity diagram

Pomocí diagramu aktivit se zobrazují jednotlivé kroky při průchodu složitějším algoritmem. V tomto případě je krok po kroku zobrazen proces přihlašování uživatele do webového informačního systému. Nejdříve se kontroluje správnost uživatelského jména a

hesla. V případě nekorektních přihlašovacích údajů je uživatel vyzván na nové zadání. Po korektním vyplnění přihlašovacích údajů systém ověřuje roli, kterou má daný uživatel přiřazenou. V tomto případě se jedná o tři role a to administrátor, lékař a pacient. Toto je nutné pro zobrazení odlišného uživatelského prostředí. Proces končí po úspěšném odhlášení uživatele ze systému.

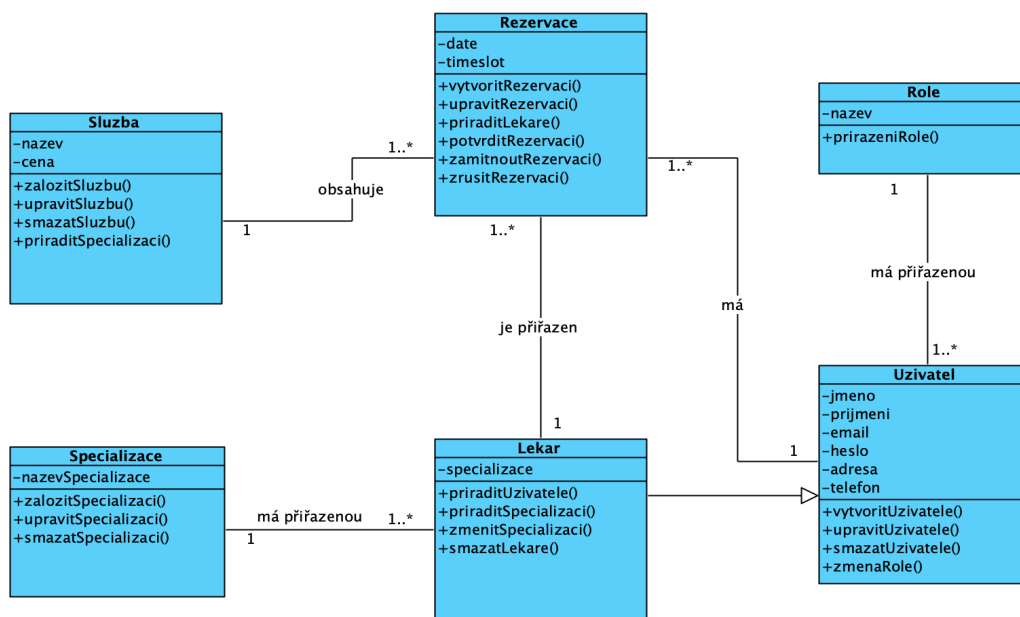


Obrázek 21 - Activity diagram

4.1.4 Class diagram

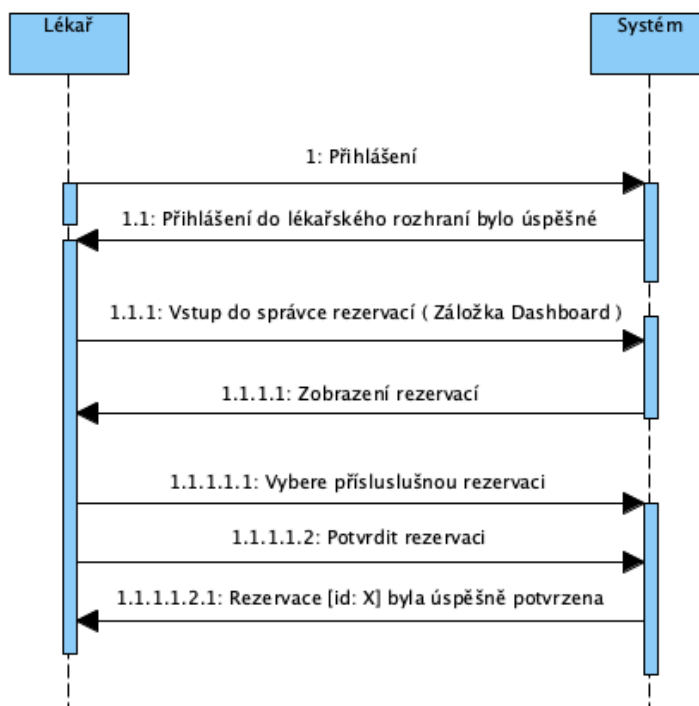
Dalším z uvedených diagramů je class diagram neboli diagram tříd. Jedná se o implementační diagram, který složí pro programátora jako návod při zpracování řešení systému. Pro zde navrhovaný informační systém bylo vytvořeno šest tříd. První část tříd se zabývá údaji i uživatelských účtech a uživateli samotných. Hlavní třídou je třída rezervací, kde se uchovávají informace o budoucích i proběhlých rezervacích služeb zdravotnického centra. Pro propojení s uživatelem a lékařem slouží vazby na objekt Uživatel, kde se vybere uživatel, který si službu objednal a objekt Lékař, což je objekt propojující jednotlivé lékaře a jejich specializace. Kardinalita vztahu mezi rezervací a uživatelem je, že jeden uživatel může mít vždy více rezervací, ale rezervace může mít pouze jednoho uživatele. Stejný případ je i v případě vazby na lékaře, jeden lékař může být na více rezervacích klientů, ale právě jedna rezervace má pouze jednoho lékaře.

Dalšími třídami v systému jsou třídy role, specializace a služba, které fungují jako číselník. Mají statický obsah, který se upravuje jen ve výjimečných případech. Pro přístup k jejich úpravám má pouze administrátor systému. Kardinalita v tomto případě je v případě rolí, že jedna role může mít nespočet uživatelů, ale jeden uživatel může mít pouze jednu roli. Obdobně je řešena i u tříd specializace a služba.

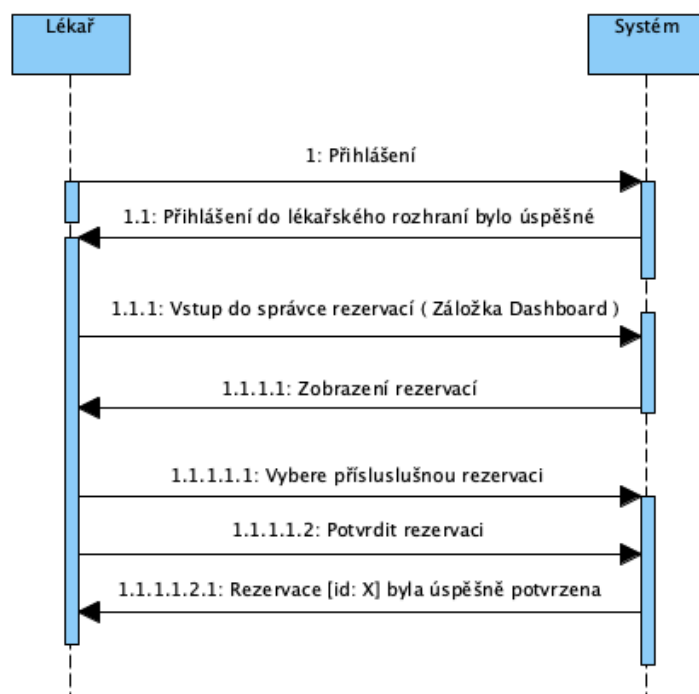


Obrázek 22 - Class diagram

4.1.5 Sekvenční diagram



Obrázek 23 - Sekvenční diagram: Potvrzení rezervace



Obrázek 24 - Sekvenční diagram: Vytvoření nové rezervace

4.2 Návrh informačního systému

4.2.1 Datový slovník

cUzivatel

Tato třída obsahuje veškeré informace o uživatelském účtu, pro identifikaci a přihlašování do systému se využívá e-mailová adresa. Dále tato třída obsahuje informaci o bydlišti a kontaktní údaje.

Pole	Typ	Nulový	Výchozí	Komentáře
id	int(11)	Ne		Primární klíč
titul	varchar(255)	Ano	NULL	
jmeno	varchar(255)	Ne		
prijmeni	varchar(255)	Ne		
email	varchar(255)	Ne		
heslo	varchar(255)	Ne		Hash hesla
token	varchar(255)	Ano	NULL	
verify	bit(1)	Ne		
adresa	varchar(255)	Ne		
mesto	varchar(255)	Ne		
stat	varchar(255)	Ne		
psc	smallint(5)	Ne		
telefon	int(11)	Ne		
cRole_id	int(11)	Ne		Cizí klíč

cRezervace

Třída rezervací obsahuje informaci o datu rezervace, vybraném časovém slotu, a skrz cizí klíče informaci o zvolené službě, pacientovi a lékaři. Také v sobě drží informaci o potvrzení případně smazání záznamu.

Pole	Typ	Nulový	Výchozí	Komentáře
id_rezervace	int(11)	Ne		Primární klíč
date	date	Ne		
timeslot	varchar(255)	Ne		
cSluzba_id	int(11)	Ne		Cizí klíč

cUzivatel_id	int(11)	Ne		Cizí klíč
cLekar_id	int(11)	Ne		Cizí klíč
isDeleted	bit(1)	Ne	b'0'	
isValidate	bit(1)	Ne	b'0'	

cSluzba

Tato třída obsahuje základní informace o službě, jako její název a cenu za jednotku, a informaci k jaké specializaci se služba váže.

Pole	Typ	Nulový	Výchozí	Komentáře
id	int(11)	Ne		Primární klíč
nazev	varchar(255)	Ne		
cena	int(11)	Ne		
cSpecializace_id	int(11)	Ne		Cizí klíč

cSpecializace

Třída specializace obsahuje pouze základní informace o specializaci.

Pole	Typ	Nulový	Výchozí	Komentáře
id	int(11)	Ne		Primární klíč
nazevSpecializace	varchar(255)	Ne		

cLekar

Vazební tabulka lékař sama o sobě žádnou informaci nenes, ale propojuje dvě další tabulky: cUzivatel a cSpecializace.

Pole	Typ	Nulový	Výchozí	Komentáře
id	int(11)	Ne		Primární klíč
cSpecializace_id	int(11)	Ne		Cizí klíč
cUzivatel_id	int(11)	Ne		Cizí klíč

cRole

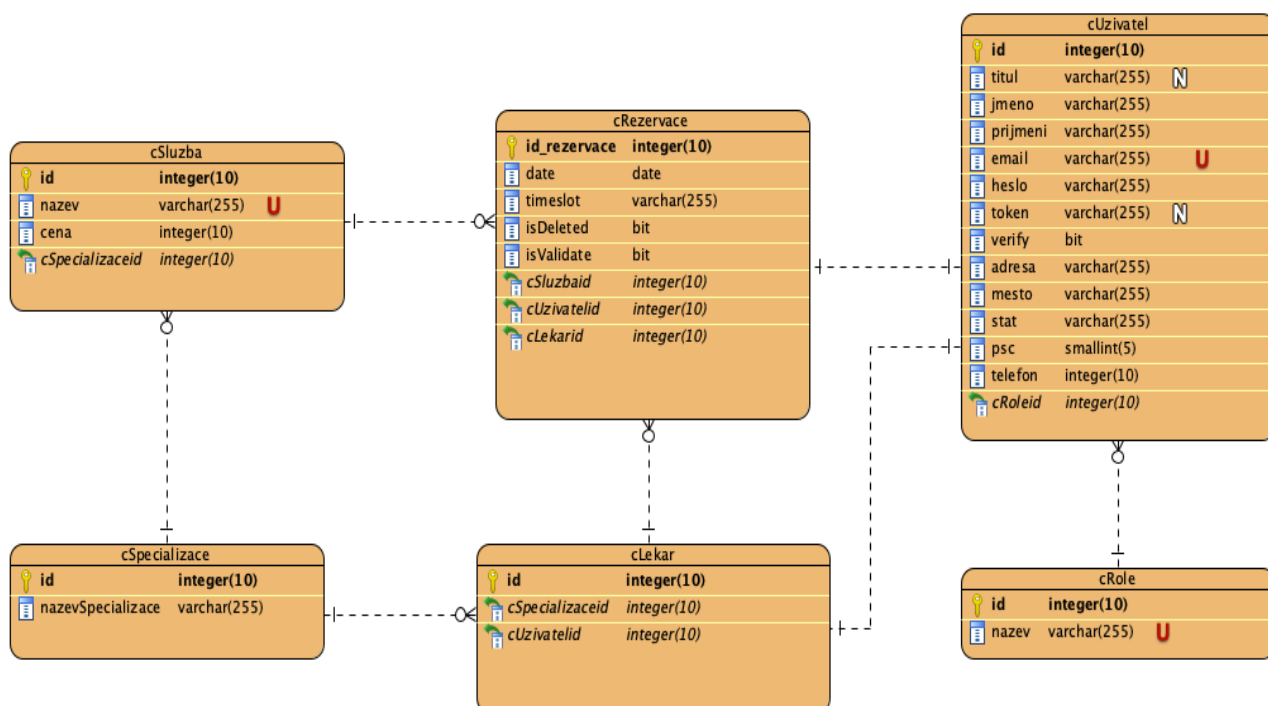
V třídě se nachází informace o uživatelských rolích, které jsou v informačním systému využívány.

Pole	Typ	Nulový	Výchozí	Komentáře
id	int(11)	Ne		Primární klíč

nazev	varchar(255)	Ne		
-------	--------------	----	--	--

4.2.2 Entity relationship diagram

Následující diagram znázorňuje vytvořenou strukturu v databázi. Hlavní entitou systému je tabulka rezervací a uživatelů. Tabulky role, služba a specializace jsou statické číselníky. Poslední entitou systému je tabulka lékař, která v sobě obsahuje vazbu mezi uživatelem (s rolí „lékař“) a jeho specializací.

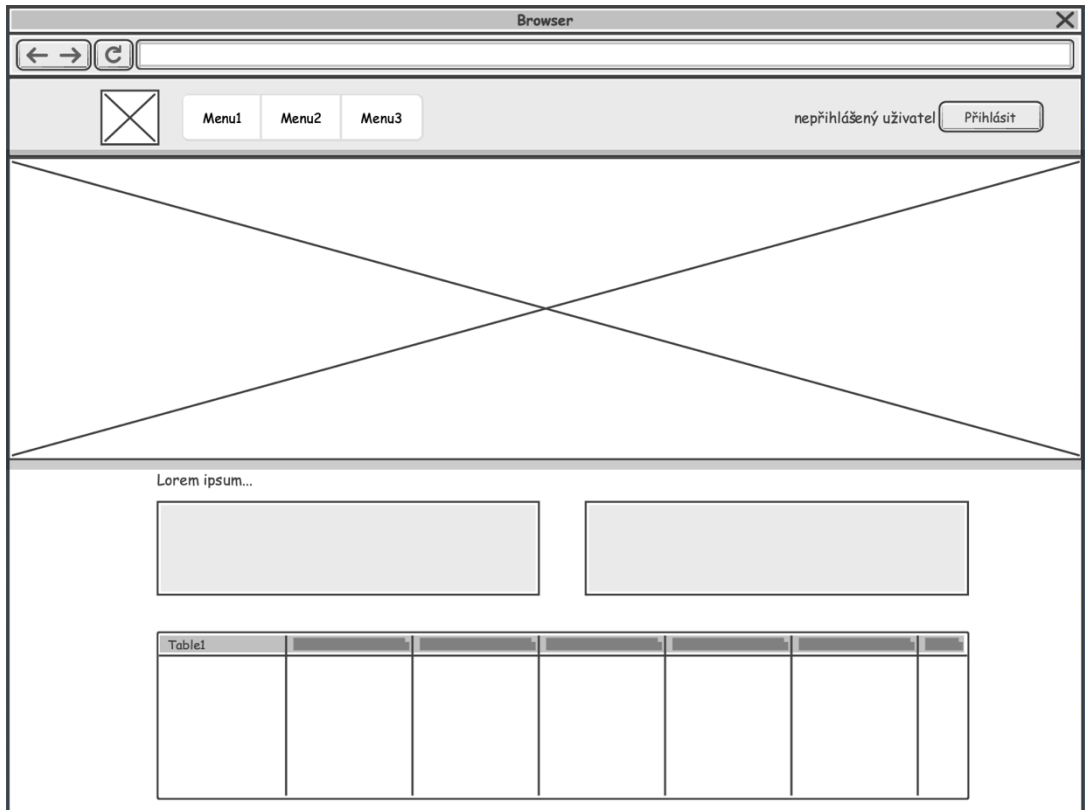


Obrázek 25 - Entitně relační diagram

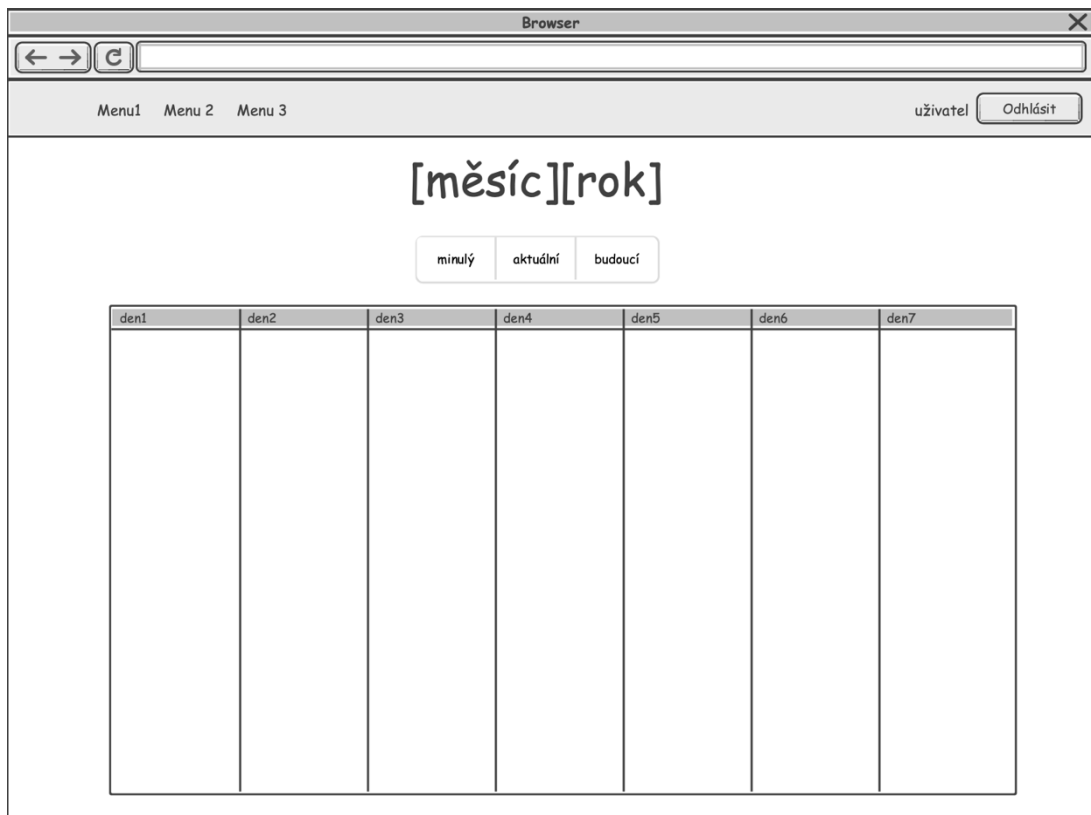
4.2.3 Wireframe

Wireframe (drátový model) je ukázka logického náčrtu vzhledu webové stránky. Zobrazuje kde se přibližně bude nacházet menu, jednotlivé obrázky na webové stránce, formuláře a přihlašovací údaje. Po potřeby tohoto informačního systému byly vytvořeny následující tři wireframe:

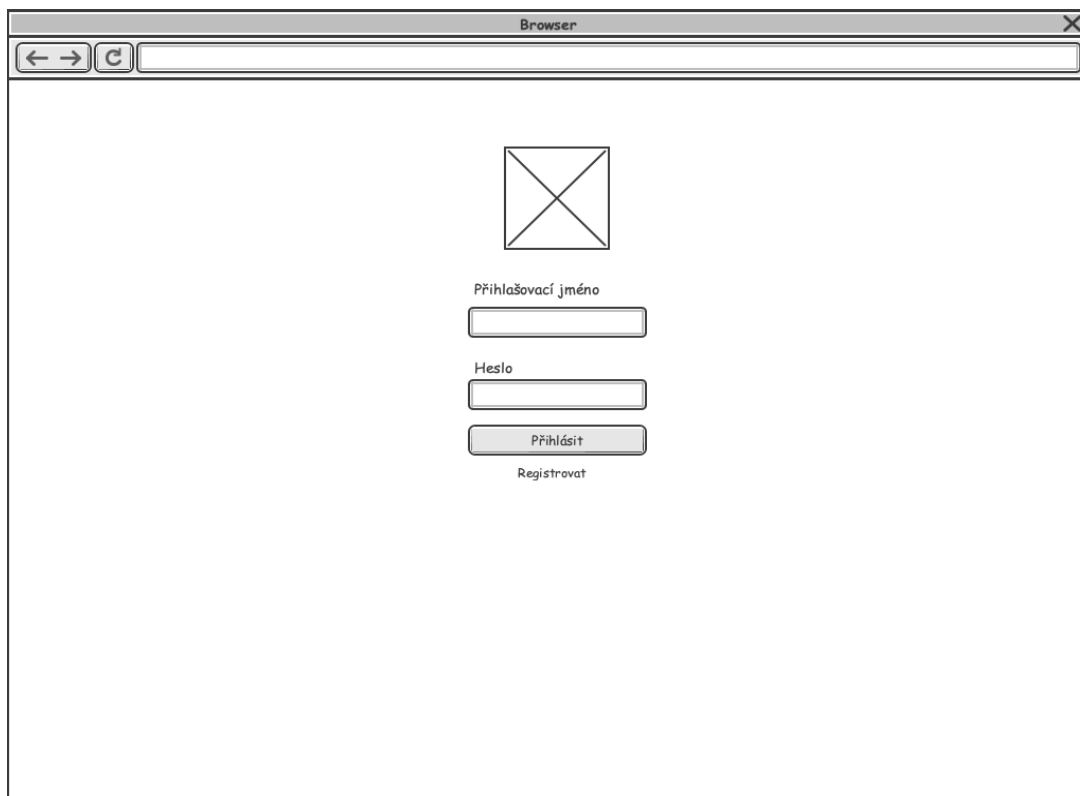
- Wireframe úvodní stránky, která se zobrazuje i neregistrovaným uživatelům
- Wireframe rezervačního kalendáře
- Wireframe přihlašovací obrazovky



Obrázek 26 - Wireframe 1: Úvodní obrazovka



Obrázek 27 - Wireframe 2: Rezervační kalendář



Obrázek 28 - Wireframe 3: Přihlašovací obrazovka

4.3 Implementace

Následující kapitoly se již zabývají samotnou implementací webového informačního systému. Pro potřeby tohoto informačního systému byla zvolena relační databáze. Ta slouží k uchování všech dat a pro jednotlivé komponenty systému. V dalších kapitolách budou uvedeny jednotlivé části funkcionalit systému, které jsou pro chod důležité. Kompletní zdrojový kód je přiložen v rámci příloh této práce. Pro implementaci byl využit software Visual Studio Code, pro práci s databází byla využita webová služba PhpMyAdmin, která je součástí webového serveru.

4.3.1 Vytvoření struktury databáze

V následujících částech kódu je vzorově ukázán skript pro vytvoření tabulky, jejich vzájemné propojení skrz cizí klíče a přidání integritního omezení. Níže je SQL příkaz pro vytvoření tabulky rezervací, se všemi potřebnými hodnotami, případně defaultně nastavenými hodnotami.

```
CREATE TABLE IF NOT EXISTS `cRezervace` (  
    `id_rezervace` int(11) NOT NULL,  
    `date` date NOT NULL,  
    `timeslot` varchar(255) COLLATE  
cp1250_czech_cs NOT NULL,  
    `cSluzba_id` int(11) NOT NULL,  
    `cUzivatel_id` int(11) NOT NULL,  
    `cLekar_id` int(11) NOT NULL,  
    `isDeleted` bit(1) NOT NULL DEFAULT  
b'0',  
    `isValidate` bit(1) NOT NULL DEFAULT  
b'0'  
) ENGINE=InnoDB DEFAULT CHARSET=cp1250 COLLATE=cp1250_czech_cs;
```

Dále přidání primárního klíče a vytvoření indexů pro přístup do dalších tabulek:

```
ALTER TABLE `cRezervace`  
    ADD PRIMARY KEY (`id_rezervace`), ADD KEY `cSluzba_id`  
(`cSluzba_id`),  
    ADD KEY `cUzivatel_id` (`cUzivatel_id`), ADD KEY `cLekar_id`  
(`cLekar_id`);
```

Modifikování primárního klíče, aby se id generovalo automaticky:

```
ALTER TABLE `cRezervace`  
    MODIFY `id_rezervace` int(11) NOT NULL AUTO_INCREMENT;
```

Posledním krokem před finálním naplněním tabulky je přidání jednotlivých omezení v rámci referenční integrity dat.

```
ALTER TABLE `cRezervace`  
    ADD CONSTRAINT `cRezervace_ibfk_1` FOREIGN KEY  
(`cUzivatel_id`) REFERENCES `cUzivatel` (`id`),  
    ADD CONSTRAINT `cRezervace_ibfk_2` FOREIGN KEY  
(`cLekar_id`) REFERENCES `cLekar` (`id`),  
    ADD CONSTRAINT `cRezervace_ibfk_3` FOREIGN KEY  
(`cSluzba_id`) REFERENCES `cSluzba` (`id`)  
ON DELETE RESTRICT ON UPDATE RESTRICT;
```

4.3.2 Struktura aplikace

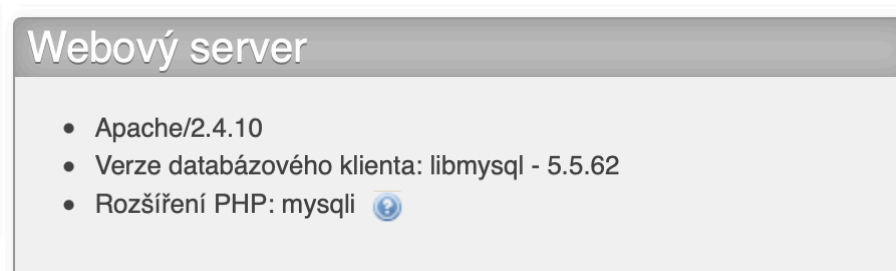
Aplikace je tvořena pomocí programovacího jazyka PHP, pro ukládání dat do databáze byla využita databáze MySQL (MariaDb), která je spravována pomocí webového rozhraní phpMyAdmin. Pro hostování informačního systému byl zvolena varianta od firmy Zikum, která obsahuje možnost využití technologií:

- PHP5
- PHP6
- PHP7
- MySQL
- phpMyAdmin

Webový systém je spuštěn na adrese: <https://lttri.www3.cz>.



Obrázek 29 - Zdravotnické zařízení - Nastavení databázového serveru



Obrázek 30 - Zdravotnické zařízení - Nastavení webového serveru

Tabulka	Operace	Řádků	Typ	Porovnávání	Velikost	Navíc
<input type="checkbox"/> cLekar	★ Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	1	InnoDB	utf8_czech_ci	48 KiB	-
<input type="checkbox"/> cParametry	★ Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	0	InnoDB	utf8_czech_ci	16 KiB	-
<input type="checkbox"/> cRezervace	★ Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	11	InnoDB	utf8_czech_ci	64 KiB	-
<input type="checkbox"/> cRole	★ Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	3	InnoDB	utf8_czech_ci	16 KiB	-
<input type="checkbox"/> cSluzba	★ Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	11	InnoDB	utf8_czech_ci	32 KiB	-
<input type="checkbox"/> cSpecializace	★ Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	4	InnoDB	utf8_czech_ci	16 KiB	-
<input type="checkbox"/> cUzivatel	★ Projít Struktura Vyhledávání Vložit Vyprázdnit Odstranit	3	InnoDB	utf8_czech_ci	32 KiB	-
7 tabulek	Celkem	33	InnoDB	utf8_czech_ci	224 KiB	0 B

Obrázek 31 - Zdravotnické zařízení - Seznam využitých tabulek v databázi

4.3.3 Připojení do databáze

Pro vytvoření spojení stránek s databází se využívá externího připojovacího souboru *dbconnect.php*, která je součástí každé stránky.

```
<?php
```

```
$servername = "free1db.zikum.cz";
$username = "lttri.www3.cz";
$password = "*****";
$database = "lttri_www3.cz";
```

```
$link=
mysqli_connect($servername,$username,$password,$database);
```

```
if (!$link) {
echo "Error: Unable to connect to MySQL." .PHP_EOL;
echo "Debugging errno: ".mysqli_connect_errno().PHP_EOL;
echo "Debugging error: ".mysqli_connect_error().PHP_EOL;
exit();
}
?>
```

4.3.4 Rozdělení uživatelských rolí

V informačním systému jsou použity tři typy rolí uživatelských účtů. Jednotlivé role jsou uchovány v podobě cizího klíče v tabulce *cUzivatel* položka *cRole_id*, která odkazuje na tabulku *cRole*. Druhy uživatelských účtů jsou následující:

- Administrátor (*cRole_id = 1*) - Administrátor má v rámci systému hlavní oprávnění nahlížet do tabulky Uživatelů, kde má přístup ke všem informacím o účtech, může přidávat nové uživatelské účty (především lékařské účty), přidávat jednotlivým lékařským účtům specializaci. Prohlížet data v tabulkách specializací a služeb, do kterých je možno vkládat nová data.

- Lékař (*cRole_id* = 2) – Uživatel lékařského účtu má primárně práva na práci s jednotlivými rezervacemi, na hlavní stránce se zobrazují nepotvrzené, potvrzené a proběhlé rezervace, s budoucími rezervacemi lze pracovat, lze je potvrdit, zamítnout nebo zrušit.
- Pacient (*cRole_id* = 3) – Pacient v rámci systému má práva na vytváření nové rezervace, prohlížení svých vytvořených rezervací a přehled již proběhlých vyšetření.

Všechny účty navíc mají právo na úpravu uložených osobních dat uložených v databázi.

```

$email = $_SESSION['user'];
$query = "select * from cUzivatel where email = '$email'";
$run = mysqli_query($conn, $query);
    if(mysqli_num_rows($run)>0){
        $row = mysqli_fetch_array($run);
        $cRole_id = $row['cRole_id'];
    }
if($cRole_id == 1){
    $setRole = "<span style='margin-right:
0.5em;color:white;'>".$email."<span> </span><span class='badge
badge-pill badge-danger'>Administrátor</span></span>";
}elseif($cRole_id == 2){
    $setRole = "<span style='margin-right:
0.5em;color:white;'>".$email."<span> </span><span class='badge
badge-pill badge-warning'>Lékař</span></span>";
}elseif($cRole_id == 3){
    $setRole = "<span style='margin-right:
0.5em;color:white;'>".$email."<span> </span><span class='badge
badge-pill badge-primary'>Pacient</span></span>";
}else{
    $msg = "<div class='alert alert-danger'>Problém s
cRole_id v databázi, kontaktujte      správce</div>";
}
$conn->close();

```

Víše je ukázka kódu na rozdělení uživatelských rolí, v této variantě různě zobrazuje v hlavičce menu informaci o přihlášeném uživateli a typu jeho účtu.

4.3.5 Předávání informací o přihlášeném uživateli

Informace o tom, zda je uživatel v systému přihlášen, je globálně v tomto informačním systému přenášena v rámci globální proměnné *\$_SESSION*. Na každé stránce, si systém

zjišťuje zda je v proměnné `SESSION["user"]` uložena nějaká hodnota. V případě že se v proměnné emailová adresa nachází, je uživatel do systému přihlášen a je mu umožněn vstup do sekce pacienta/lékaře/admina.

```
if(isset($_SESSION['user'])){\n    $email = $_SESSION['user'];\n    $query = "select * from cUzivatel where email =\n'$email'";\n    $run = mysqli_query($conn, $query);\n    if(mysqli_num_rows($run)>0){\n        $row = mysqli_fetch_array($run);\n        $cRole_id = $row['cRole_id'];\n        }\n        if($cRole_id == 1){...\n        }\n    elseif($cRole_id == 2){...\n    }\n    elseif($cRole_id == 3){...\n    }\n    else{...\n    }\n    $conn->close();\n}\nelse{\n    header("location: index.php");\n}
```

V tomto případě tato část kódu zjišťuje, zda je nějaký uživatel uložený v session proměnné. Pokud konstrukce if podmínku vyhodnotí jako true, pokračuje systém dál a zjišťuje jaká role je uživateli přiřazena. Pokud se podmínka vyhodnotí jako false, tak je uživatel vrácen na úvodní stránku. Tím je zajištěno, že se do rozhraní aplikace nedostane žádný nepřihlášený uživatel.

4.3.6 Úvodní strana, přihlašovací obrazovka a registrace

Při vstupu uživatele na webové stránky se zobrazí hlavní stránka webu. Na ní se nachází v horní části menu, které slouží pro jednoduchou orientaci v jednotlivých částech webu. V pravé části menu se pak nachází informace o přihlášeném uživateli a jeho typu účtu případně v případě nepřihlášeného uživatele tlačítko Přihlásit.

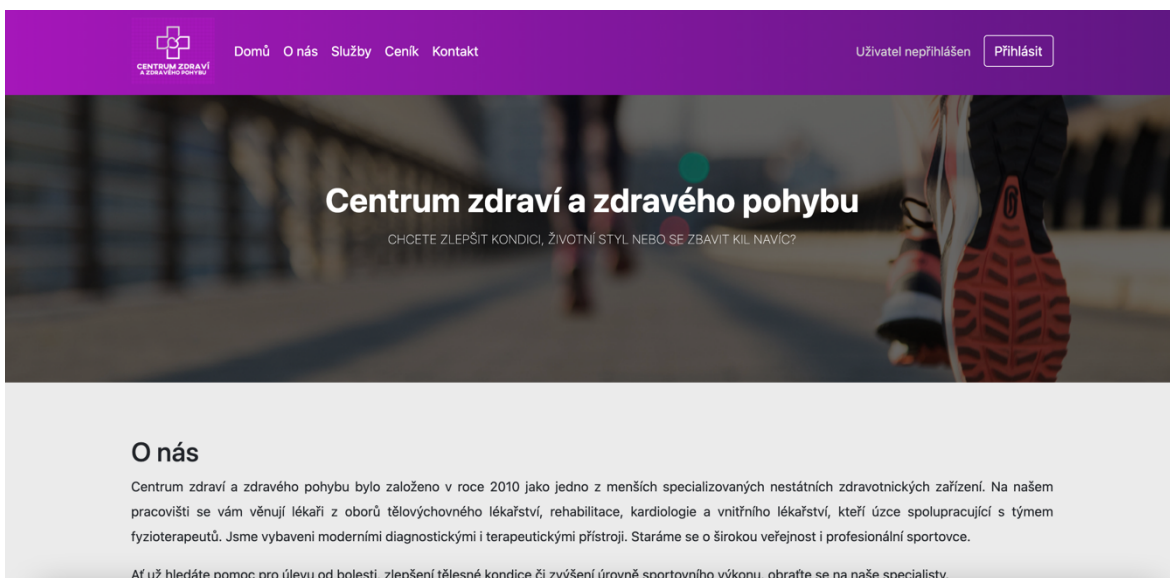


Obrázek 32 - Zdravotnické zařízení - Menu (nepřihlášený uživatel)



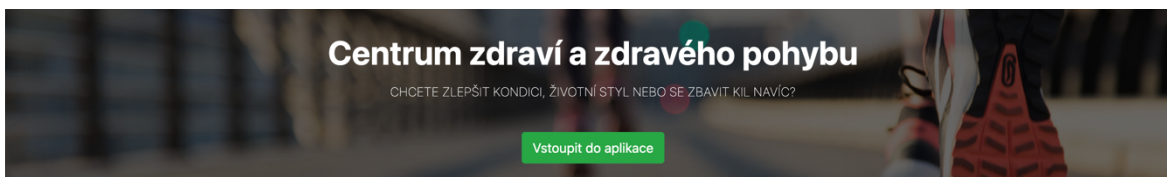
Obrázek 33 - Zdravotnické zařízení - Menu (přihlášený uživatel)

Dále na úvodní stránce nalezneme informace o jednotlivých typech služeb, které centrum nabízí. Při rozkliknutí jednotlivých služeb se zobrazí detailní popis vyšetření a služby. Níže pak nalezneme kompletní ceník služeb zdravotního centra a kontaktní formulář, kterým lze komunikovat s recepcí centra. Po vyplnění formuláře a odeslání emailu, je zaslaná zpráva na kontaktní adresu centra.



Obrázek 34 - Zdravotnické zařízení - Úvodní strana

Pro vstup do aplikace z hlavní stránky, v případě přihlášeného uživatele, slouží tlačítko uprostřed úvodního loga.




Obrázek 35 - Zdravotnické zařízení – Úvodní stránka – vstup do aplikace

Služby

Nabízíme komplexní služby pro rekreační i výkonnostní běžce, cyklisty, triatlonisty a všechny, kdo chtějí získat kondici a zdraví.

Trénujte správně a bez rizika!

Všechny služby (funkční vyšetření, analýzy techniky pohybu, fyzioterapii, tréninkové plány, výživové poradenství...) nabízíme samostatně nebo v oblíbených balíčcích.




Běh

Služby pro rekreační i výkonnostní běžce

- Vyšetření laktátové křivky
- Klidové i zátěžové EKG, klidová spirometrie
- Určení bezrizikových tréninkových zón
- Kontrola techniky běhu
- Individuální tréninkový plán

Zjistit více



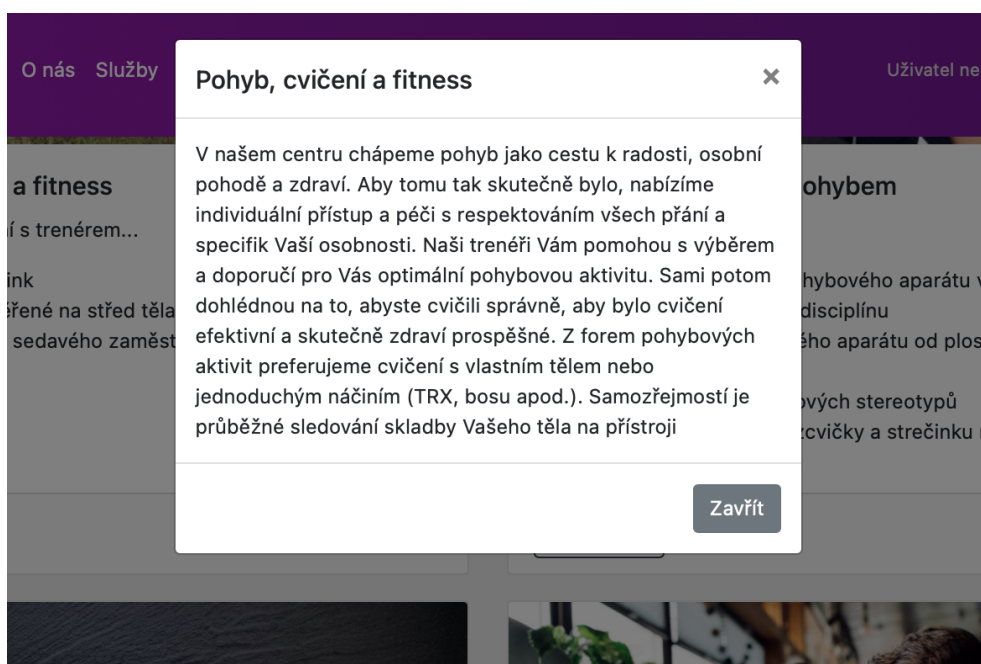
Cyklistika a Triatlon

Funkční vyšetření, tréninkové plány...

- Vyšetření laktátové křivky
- Klidové i zátěžové EKG, klidová spirometrie
- Určení bezrizikových tréninkových zón
- Vyšetření srdce a laktátové křivky + klidová spirometrie
- Individuální tréninkový plán

Zjistit více

Obrázek 36 - Zdravotnické zařízení – Úvodní stránka (část – Služby)



Pohyb, cvičení a fitness

V našem centru chápeme pohyb jako cestu k radosti, osobní pohodě a zdraví. Aby tomu tak skutečně bylo, nabízíme individuální přístup a péči s respektováním všech přání a specifik Vaší osobnosti. Naši trenéři Vám pomohou s výběrem a doporučí pro Vás optimální pohybovou aktivitu. Sami potom dohlédnou na to, abyste cvičili správně, aby bylo cvičení efektivní a skutečně zdraví prospěšné. Z forem pohybových aktivit preferujeme cvičení s vlastním tělem nebo jednoduchým náčiním (TRX, bosu apod.). Samozřejmostí je průběžné sledování skladby Vašeho těla na přístroji

Zavřít

Obrázek 37 – Zdravotnické zařízení – Úvodní stránka (část – Služby)

Na obrázku výše je rozkliknutá jedna ze služeb *Pohyb, cvičení a fitness* a zobrazeno okno, kde jsou souhrnné informace o službě, které by se na hlavní stránku již nevešly. Pro návrat je připraveno tlačítko *Zavřít*.

Ceník

Služba	Cena
Poradenství v oblasti životního stylu	1200,-
Funkční diagnostika - vyšetření laktátové křivky + analýza tělesné skladby + konzultace	1900,-
<i>Funkční diagnostika - vyšetření laktátové křivky - opakované vyšetření do 12 měsíců</i>	1700,-
Vstupní vyšetření pohybového aparátu fyzioterapeutem (1 hodina)	1250,-
Fyzioterapie (1hodina)	1000,-
Fyzioterapie pro děti (30 minut)	550,-
Analýza tělesné skladby + konzultace	300,-
Modelace ortopedických stélek	1800,-
Základní výživové poradenství + analýza tělesné skladby	1600,-
Výživové poradenství s kontrolní konzultací	2900,-
Výživové poradenství se dvěma kontrolními konzultacemi a jídelníčkem na 3 dny	5800,-

Obrázek 38 - Zdravotnické zařízení – Úvodní stránka (část – Ceník)

Kontakt

Provozní doba: PO-PÁ 7:00 - 22:00

Adresa

Poděbradská 10
PSČ: 190 00 - Praha 9

Email

test@email.cz
contact@email.cz

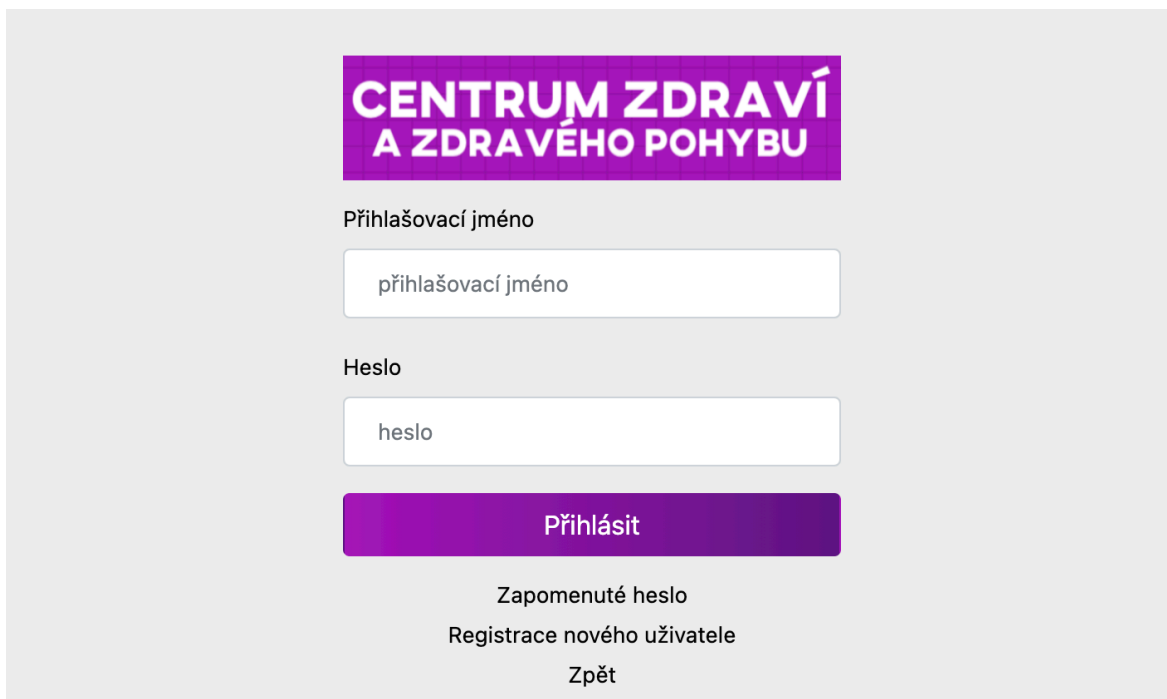
Telefon

+ 420 606 123 123
+ 420 226 123 123

Copyright © Lukáš Tvrz 2020

Obrázek 39 - Zdravotnické zařízení – Úvodní stránka (část – Kontakt)

Po kliknutí na tlačítko přihlásit se zobrazí přihlašovací formulář, který se skládá ze dvou formulářových polí. Do pole uživatelského jména je vkládaná emailová adresa, a do druhého pole se vloží heslo. Po kliknutí na tlačítko Přihlásit, systém zkontroluje, zda přihlašovací údaje souhlasí, a v případě že vyhodnotí kladně, je uživatel přihlášen do systému.



**CENTRUM ZDRAVÍ
A ZDRAVÉHO POHYBU**

Přihlašovací jméno

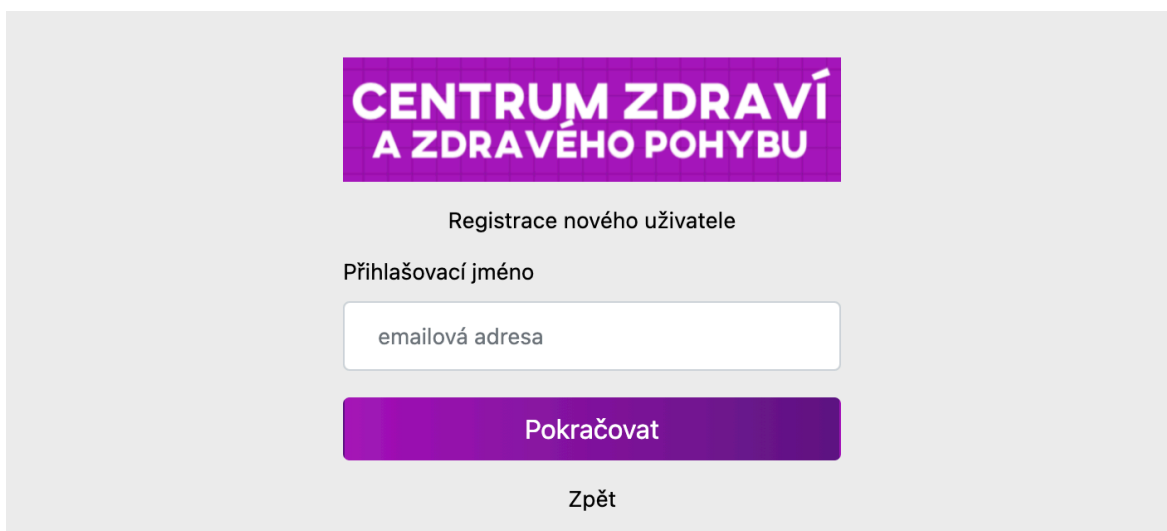
Heslo

Přihlásit

[Zapomenuté heslo](#)
[Registrace nového uživatele](#)
[Zpět](#)

Obrázek 40 - Zdravotnické zařízení – Přihlašovací obrazovka

V případě, že uživatel ještě nemá uživatelský účet, klikne na tlačítko Registrovat nového uživatele, systém jej přesměruje do registračního formuláře. Nejprve je uživatel dotázán na emailovou adresu, systém po odeslání dat formuláře zkontroluje, zda se v databázi stejná adresa již nenachází (jelikož se do systému skrz emailovou adresu přihlašuje, tak musí být unikátní). Pokud adresa použita ještě nebyla, je uživateli umožněn vyplnit zbytek uživatelských údajů. V opačném případě je potřeba zvolit jiné uživatelské jméno.



**CENTRUM ZDRAVÍ
A ZDRAVÉHO POHYBU**

Registrace nového uživatele

Přihlašovací jméno

Pokračovat

[Zpět](#)

Obrázek 41 - Zdravotnické zařízení – Nová registrace

**CENTRUM ZDRAVÍ
A ZDRAVÉHO POHYBU**

Registrace nového uživatele

Uživatel s touto emailovou adresou
již existuje

Přihlašovací jméno

emailová adresa

Pokračovat

Zpět

Obrázek 42 - Zdravotnické zařízení – Nová registrace (kontrola na existenci uživatele)

**CENTRUM ZDRAVÍ
A ZDRAVÉHO POHYBU**

Registrace nového uživatele

Přihlašovací jméno

lukastvrz@ssw.com

Titul Jméno * Příjmení *

titul jméno příjmení

Heslo * Ověření hesla *

heslo ověření hesla

Adresa * Telefon *

adresa +420 telefon

Město * Stát * PSČ *

město ČESKÁ REPUBLIKA psč

Nejsem robot reCAPTCHA
Ochrana soukromí - Smluvní podmínky

Souhlasíte se zpracováním osobních údajů? [Ochrana osobních údajů GDPR](#)

Registrovat

Zpět

(*) Povinné

Obrázek 43 - Zdravotnické zařízení – Nová registrace

Po úspěšné kontrole je emailová adresa uzamčena, aby již nemohla být změněna, jelikož by kontrola v předchozím kroku byla bezpředmětná. Po vyplnění všech povinných polí a stisknutí tlačítka Registrovat, je uživatel přesměrován na přihlašovací obrazovku. Zobrazí se mu notifikace, že uživatel byl založen a na jeho emailovou adresu byl zaslán validační email. Do chvíle než uživatel neklikne na odkaz v emailové adrese a nepotvrdí registraci, tak uživateli nebude povolen přístup do systému. Po aktivaci účtu se zobrazí notifikace, že účet byl úspěšně ověřen a uživatel se přihlásí do úvodní sekce systému. Ověření uživatelského jména při přihlašování je řešeno viz níže:

```

if(isset($_POST['prihlasit'])){
    $email = mysqli_real_escape_string($conn, ($_POST['email']));
    $heslo = mysqli_real_escape_string($conn, ($_POST['password']));
    $heslo = md5($heslo);

    $query = "select * from cUzivatel where email = '$email'";
    $run = mysqli_query($conn, $query);
    if(mysqli_num_rows($run)>0){
        $row = mysqli_fetch_array($run);
        $db_heslo = $row['heslo'];
        $db_email = $row['email'];
        $verify = $row['verify'];
        if($db_heslo == $heslo){

            if($verify == 1){
                $_SESSION['user'] = $email;
                header("location: dashboard.php");
            }else{
                $msg = "<div class='alert alert-danger'>Emailová adresa není
ověřena</div>";
            }
        }else{
            $msg = "<div class='alert alert-danger'>Zadaná hesla se
neshodují</div>";
        }
    }else{
        $msg = "<div class='alert alert-danger'>Emailová adresa
nenalezena</div>";
    }
}

```

V další části zdrojového kódu (verifyRegistration.php) je ukázka zjišťování zda již emailová adresa nebyla použita.

```

<?php
session_start();
include 'dbconn.php';

// připojení k databázi
$conn = new mysqli($servername,$username,$password,$database);

```

```

$conn->set_charset("utf8");
if(isset($_POST['submit'])){
    $email = mysqli_real_escape_string($conn, ($_POST['email']));

    $query = "select * from cUzivatel where email = '$email'";
    $run = mysqli_query($conn, $query);
    if(mysqli_num_rows($run)>0){
        $msg = "<div class='alert alert-danger text-center'>Uživatel s
touto          emailovou adresou již existuje</div>";
    }else{
        header("location: register.php");
        $_SESSION['registrace'] = $email;
    }
}

```

?>

Zdrojový kód kontroly správnosti vyplněných údajů a kvalita hesla je kontrolována způsobem viz. níže, na závěr je odeslán validační email. Pro odeslání emailu byla využita zabudovaná funkce PHP (musí být podporována poskytovatelem webového serveru)

```

if(isset($_POST['submit'])){
    $email = $_SESSION['registrace'];
    $titul = mysqli_real_escape_string($conn, ($_POST['titul']));
    $name = mysqli_real_escape_string($conn, ($_POST['name']));
    $surname = mysqli_real_escape_string($conn, ($_POST['surname']));
    $password1 = mysqli_real_escape_string($conn, ($_POST['password1']));
    $password2 = mysqli_real_escape_string($conn, ($_POST['password2']));
    $address = mysqli_real_escape_string($conn, ($_POST['address']));
    $phone = mysqli_real_escape_string($conn, ($_POST['phone']));
    $city = mysqli_real_escape_string($conn, ($_POST['city']));
    $state = mysqli_real_escape_string($conn, ($_POST['state']));
    $zip = mysqli_real_escape_string($conn, ($_POST['zip']));

    if($password1 != $password2){
        $msg = "<div class='alert alert-danger text-center'>Zadaná hesla se
neshodují</div>";
    }
    else{

        // kontrola síly hesla
        if ( strlen($password1 ) < 8 ) {
            $msg = "<div class='alert alert-danger text-center'>Zadané heslo
je           příliš krátké</div>";
        }
        elseif (!preg_match("#[0-9]+#", $password1 ))
        {
            $msg = "<div class='alert alert-danger text-center'>Zadané heslo
musí        obsahovat alespoň jedno číslo</div>";
        }
        elseif (!preg_match("#[a-z]+#", $password1)){
            $msg = "<div class='alert alert-danger text-center'>Zadané heslo
musí        obsahovat alespoň jedno písmeno</div>";
        }
    }
}

```

```

elseif (!preg_match("#\W+#", $password)){
    $msg = "<div class='alert alert-danger text-center'>Zadané heslo
musí obsahovat alespoň jeden speciální znak</div>";
}else{

    // $password1 = strtolower($password1);
    $h_password = md5($password1);
    $token = uniqid(md5(time()));
    $sql = "INSERT INTO cUzivateL
(titul,jmeno,prijmeni,email,heslo,token,verify,adresa,mesto,psc,telefon,stat
,cRole_id)
values('$titul','$name','$surname','$email','$h_password','$token',0,'$addre
ss','$city','$zip','$phone','$state','3')";

    if ($conn->query($sql) === TRUE){
        // ODESLÁNÍ VALIDAČNÍHO MAILU

        $pro = $email; // nastavíme příjemce e-mailu
        $predmet = 'Ověření registrace - Centrum zdraví a zdravého
pohybu';
        $zprava = "Dobrý den,\npro ověření uživatelského účtu stačí
kliknout na odkaz níže:\n "; // samotná zpráva
        $zprava .=
"lttri.www3.cz/verify.php?token". "=3D". $token. "\n\n\n";
        $zprava .= "Toto je automatický e-mail, neodpovídejte!\n";
        $zprava .= "Centrum zdraví a zdravého pohybu\n";
        // hlavičky
        $hlavicky = 'From: lttriwww3@gmail.com'. "\n"; // můj e-mail
        $hlavicky .= "MIME-Version: 1.0\n";
        $hlavicky .= "Content-Transfer-Encoding: QUOTED-PRINTABLE\n";
        // způsob kódování
        $hlavicky .= "X-Mailer: PHP\n";
        $hlavicky .= "X-Priority: 1\n";
        // priorita (1 nejvyšší, 2 velká, 3normální ,4 nejmenší)
        $hlavicky .= 'Return-Path: <lttriwww3@gmail.com>'. "\n";
        // Návratová cesta pro chyby
        $hlavicky .= "Content-Type: text/plain; charset=UTF-8\n";
        // Kódování
        $mail = @mail($pro, $predmet, $zprava, $hlavicky);

        if($mail) {
            $_SESSION['reg-suc'] = "<div class='alert alert-success
text-center'>Registrace proběhla v pořádku, byl zaslán validační e-
mail</div>";
            header("Location: login.php");
        }
        else echo 'E-mail se bohužel nepodařilo odeslat!';

    }
    else{
        echo "Error: " . $sql . "<br>" . $conn->error;
    }
}
}
}
}

```

Pro ověření zda se nejedná o robota při odeslání registrace je využita technologie firmy Google reCaptcha ve verzi 2. Ta funguje na principu výměně veřejného a privátního klíče se serverem Google. Uživatel musí před potvrzením rezervace odpovědět na otázku. Implementace této technologie do kódu stránky je popsána níže.

4.3.6.1 reCaptcha

```
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    function post_captcha($user_response) {
        $fields_string = '';
        $fields = array(
            'secret' => '6LfD5vAZAAAAAGhU1VXZF_tYI2XXBNFcWt1rSgQW',
            'response' => $user_response
        );
        foreach($fields as $key=>$value)
            $fields_string .= $key . '=' . $value . '&';
        $fields_string = rtrim($fields_string, '&');

        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL,
'https://www.google.com/recaptcha/api/siteverify');
        curl_setopt($ch, CURLOPT_POST, count($fields));
        curl_setopt($ch, CURLOPT_POSTFIELDS, $fields_string);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, True);

        $result = curl_exec($ch);
        curl_close($ch);

        return json_decode($result, true);
    }

    // Call the function post_captcha
    $res = post_captcha($_POST['g-recaptcha-response']);

    if (!$res['success']) {
        // What happens when the CAPTCHA wasn't checked
        $msg = "<div class='alert alert-danger text-center'>Prosíme
potvrďte, že nejste robot</div>";
    } else {
        // If CAPTCHA is successfully completed...

        .. zde pokračuje kód registrace (kontrola hesla, odeslání dat do databáze..)
    }
}
```

4.3.6.2 Ověření emailové adresy

Ověření emailové adresy poté probíhá pomocí funkce v souboru *verify.php*, na který odkazuje hypertextový odkaz, který byl zaslán na registrovanou emailovou adresu.

```

if(isset($_GET['token'])){
    $t = $_GET['token'];
    $query = "select * from cUzivatel where token='$t'";
    $run = mysqli_query($conn, $query);
    if(mysqli_num_rows($run)>0){
        $row = mysqli_fetch_array($run);
        $id = $row['id'];
        $query2 = "update cUzivatel set verify='1', token='' where
id='$id'";
        $run2 = mysqli_query($conn, $query2);
        $_SESSION['message'] = "<div class='alert alert-success text-
center'>Účet byl úspěšně ověřen</div>";
        header("location: login.php");

    }else{
        header("location: login.php");
    }
}else{
    header("location: login.php");
}
}

```

4.3.6.3 Souhlas se zpracováním osobních údajů

Jelikož v databázi se nachází citlivé informace o jednotlivých pacientech (jméno, příjmení, adresa, telefon atd.), je potřeba aby před úspěšným odesláním registrace byl potvrzený checkbox „Souhlasím se zpracováním osobních údajů“. Kontrola je aplikována pomocí následující části zdrojového kódu.

```

if(isset($_POST['gdpr'])) {

    // ... kód, který proběhne ve chvíli, kdy checkbox "gdpr" aktivní

}else{

    $msg = "<div class='alert alert-danger text-center'>Prosíme potvrďte, že
souhlasíte se zpracováním osobních údajů</div>";

}

```

Na stránce je dále přidán odkaz na podrobný dokument, který popisuje nakládání s osobními údaji v souladu s GDPR pravidly.

INFORMAČNÍ MEMORANDUM O ZPRACOVÁNÍ OSOBNÍCH ÚDAJŮ SPOLEČNOSTI
CENTRUM ZDRAVÍ A ZDRAVÉHO POHYBU

(A) ÚVODNÍ USTANOVENÍ

1. ZÁKLADNÍ INFORMACE O ZPRACOVÁNÍ OSOBNÍCH ÚDAJŮ

- 1.1 Společnost **Centrum zdraví a zdravého pohybu** (dále jen „**Společnost**“ dále jen „**my**“) vydává toto informační memorandum o zpracování osobních údajů prováděném Společností (dále jen „**Memorandum**“).
- 1.2 Společnost je provozovatelem nestátního zdravotnického zařízení a předmětem podnikání Společnosti je tak především poskytování zdravotních služeb ve smyslu zákona č. 372/2011 Sb., o zdravotních službách, ve znění pozdějších předpisů (dále jen „**Zákon o zdravotních službách**“). Kromě Zákona o zdravotních službách se na péči o zdraví a zdravotní služby poskytované Společností dále vztahují příslušná ustanovení § 2636 a násl. zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů (dále jen „**Občanský zákoník**“).

Obrázek 44 - Výňatek z dokumentu GDPR

4.3.7 Rozhraní pacienta

Po úspěšném přihlášení do aplikace a zjištění o jaký typ uživatelského účtu jde (defaultně je při registraci nastavena role *Pacienta*), změnu role lze pouze pomocí administrátorského účtu. Rozhraní pacienta se skládá ze tří modulů:

- Dashboard
- Rezervace
- Nastavení

Na hlavní stránce aplikace (*dashboard*) se přehledně zobrazuje informace o budoucích a již uplynulých rezervacích. Rezervace se v tabulce řadí podle data rezervace a z důležitých údajů je zobrazena například cena služby, ošetřující lékař, a v neposlední řadě velmi důležité políčko. Zda je rezervace potvrzena. Rezervace se může dostat do třech různých stavů:

- **Nepotvrzena** – nepotvrzená rezervace, je rezervace která byla vytvořena, je považována za platnou rezervaci, ale ještě nebyla lékařem potvrzena.
- **Potvrzena** – potvrzená rezervace, je rezervace, která již byla lékařem potvrzena
- **Zrušena** – poslední stav rezervace je zrušená rezervace, do toho stavu se může rezervace dostat, poté co již byla lékařem potvrzena. Pokud z nějakého důvodu je rezervaci nutno zrušit, je to lékaři umožněno. Pacientovi se poté rezervace zobrazí s tímto stavem.

Rezervace, která byla zrušena z databáze fyzicky odstraněna není, jen je jí v databázi přiřazena hodnota sloupce *isDeleted* = 1, v tu chvíli se již tento termín v kalendáři nabízí jako dostupný.

The screenshot shows a patient dashboard with a dark header. The header contains 'Dashboard' (repeated), 'Rezervace', and 'Nastavení' on the left, and 'tvrzlukas@icloud.com' and 'Pacient' on the right, with an 'Odhlásit' button. The main content area has two sections:

Mé budoucí rezervace 3

Datum	Slot	Název vyšetření	Lékař	Cena	Potvrzení
2020-11-09	11:00AM - 12:00PM	Funkční diagnostika - Laktátová křivka, tělesná skladba, konzultace (opakované vyšetření do 12 měsíců)	MUDr. Jan Novák	1700 Kč	Nepotvrzena
2020-11-13	16:00PM - 17:00PM	Základní výživové poradenství	MUDr. Jan Novák	1600 Kč	Nepotvrzena
2020-11-20	08:00AM - 09:00AM	Fyzioterapie	MUDr. Jan Novák	1000 Kč	Potvrzena

Proběhlé rezervace 0

V databázi není žádná uskutečněná rezervace

Obrázek 45 - Zdravotnické zařízení Dashboard – pacient

4.3.7.1 Vytvoření nové rezervace

Pro vytvoření nové rezervace je potřeba se přesunout přes menu na stránku Rezervace, kde se nejprve zobrazí seznam aktuálně dostupných lékařů, ze kterých si pacient vybere.

The screenshot shows a selection screen with a large heading: 'Nejprve vyberte požadovaného lékaře:'. Below the heading are two dark grey buttons with white text:

- MUDr. Jan Novák - Fyzioterapie
- MUDr. Filip Hubený - Funkční diagnostika

At the bottom left, there is a red button with white text: 'Zpět'.

Obrázek 46 - Zdravotnické zařízení – výběr lékaře

Kliknutím na konkrétního lékaře se zobrazí kalendář s aktuálním měsícem. V kalendáři se poté zobrazují tlačítka pro vytvoření rezervace na zvolený den. Tlačítko může být ve čtyřech stavech:

- N/A (not available) - na tento den již rezervaci vytvořit nelze, lze vždy vytvořit nejdříve na následující den než je aktuální datum.
- Víkend – o víkendu centrum nefunguje, tudíž rezervace možná není
- Termíny plné – stav kdy již byla vyčerpána kapacita všech časových slotů, rezervaci již provést nelze
- Rezervovat – na tento ještě rezervaci provést lze

Listopad 2020

Minulý měsíc
Aktuální měsíc
Příští měsíc

Zvolený lékař: MUDr. Filip Hubený - Funkční diagnostika

Pondělí	Úterý	Středa	Čtvrtek	Pátek	Sobota	Neděle
						1 Víkend
2 N/A	3 N/A	4 N/A	5 N/A	6 N/A	7 Víkend	8 Víkend
9 N/A	10 N/A	11 N/A	12 N/A	13 N/A	14 Víkend	15 Víkend
16 N/A	17 N/A	18 N/A	19 N/A	20 N/A	21 Víkend	22 Víkend
23 N/A	24 N/A	25 N/A	26 N/A	27 N/A	28 Víkend	29 Víkend
30 Rezervovat <small>7 míst zbývá</small>						

Obrázek 47 - Zdravotnické zařízení Dashboard – rezervační kalendář

```

if($dayOfWeek=='5' || $dayOfWeek=='6'){
    $calendar.= "<td><h5>$currentDay</h5> <button class='btn btn-warning btn-sm'>Víkend</button></td>";
}
elseif($date<=date('Y-m-d')){

```

```

        $calendar.= "<td><h5>$currentDay</h5> <button class='btn btn-
danger btn-sm'>N/A</button></td>";
    }else{
        $totalbookings = checkSlots($mysqli,$date);
        if($totalbookings == 9){
            $calendar.= "<td class='$today'><h5>$currentDay</h5> <a href='#'
class='btn btn-danger btn-sm'>Termíny plné</a></td>";
        }else{
            $availableslots = 9 - $totalbookings;
            $calendar.= "<td class='$today'><h5>$currentDay</h5> <a
href='book.php?date=".$date."&lekar=".$lekari."' class='btn btn-
success btn-sm'>Rezervovat</a>
            <br><small><i>$availableslots míst zbývá</i></small></td>";
        }
    }
}

```

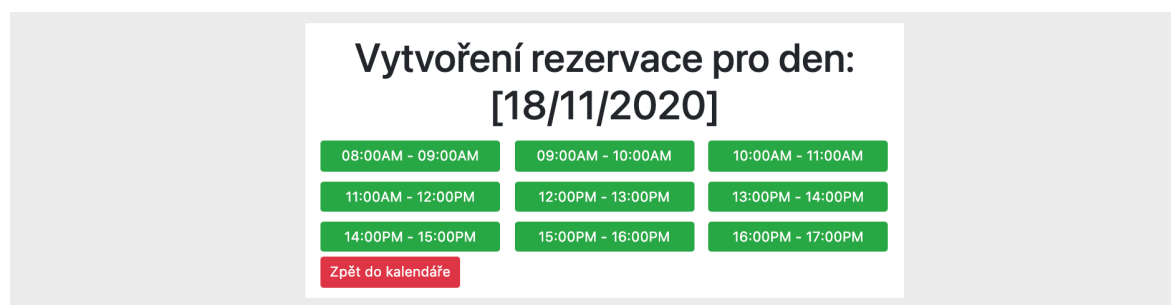
Víše je zobrazena část kódu, jakým způsobem systém zjišťuje, zda je na aktuální termín ještě kapacita. Je zde využita funkce checkSlots(), která zjišťuje počet rezervací na aktuální den.

```

function checkSlots($mysqli,$date){
    $stmt = $mysqli->prepare("select * from cRezervace where date = ?
and isDeleted = 0");
    $stmt->bind_param('s',$date);
    $totalbookings = 0;
    if($stmt->execute()){
        $result = $stmt->get_result();
        if($result->num_rows >0){
            while($row = $result->fetch_assoc()){
                $totalbookings++;
            }
            $stmt->close();
        }
    }
    return $totalbookings;
}

```

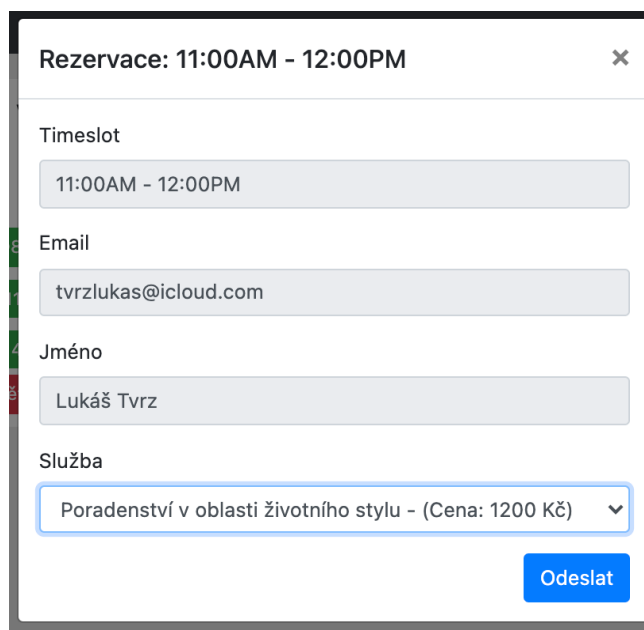
V případě, že uživatel si vybere ze seznamů volných dnů, je přesměrován na stránku s volnými časovými sloty pro vybraný den.



Obrázek 48 - Zdravotnické zařízení Dashboard – rezervační kalendář

Zde se jednotlivé časové sloty mohou nacházet ve dvou stavech, pokud je termín zelený tak je stále aktivní, v případě červené barvy je již termín obsazen a uživateli již možnost registrace nezobrazí.

Po rozkliknutí zvoleného časového termínu se zobrazí registrační tabulka, kde si uživatel pouze zkontroluje své zadané údaje a vybere službu kterou si chce rezervovat.



Rezervace: 11:00AM - 12:00PM

Timeslot
11:00AM - 12:00PM

Email
tvrzlukas@icloud.com

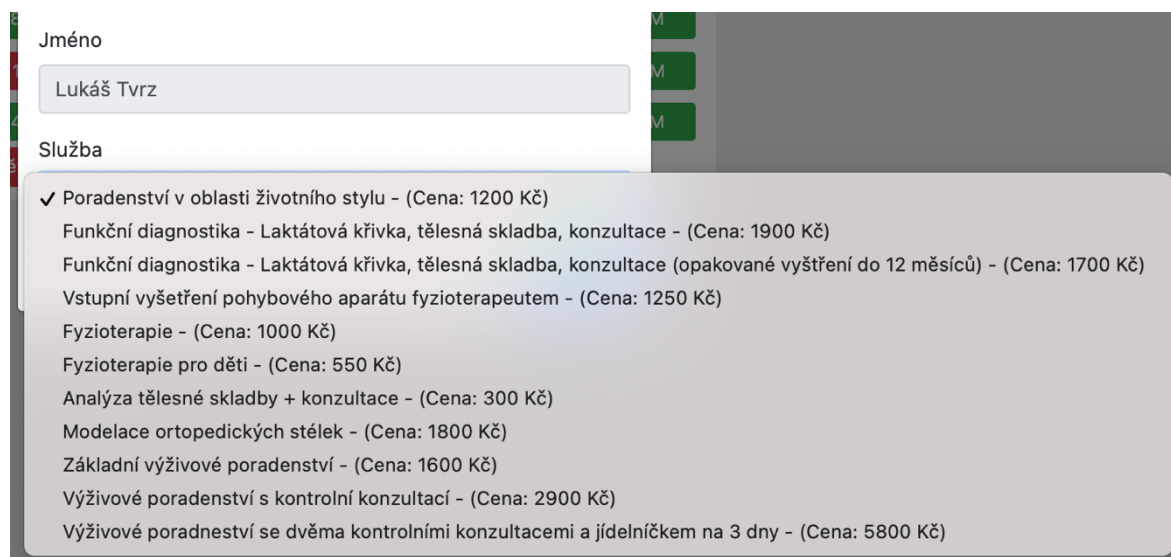
Jméno
Lukáš Tvrz

Služba
Poradenství v oblasti životního stylu - (Cena: 1200 Kč)

Odeslat

Obrázek 49 - Zdravotnické zařízení Dashboard – rezervace

Načítání dat do pole služba v tomto případě probíhá dynamicky pomocí dotazu do tabulky *cSluzba*, ze které systém získá veškeré informace o aktuálně nabízených službách.



Jméno
Lukáš Tvrz

Služba

- ✓ Poradenství v oblasti životního stylu - (Cena: 1200 Kč)
- Funkční diagnostika - Laktátová křivka, tělesná skladba, konzultace - (Cena: 1900 Kč)
- Funkční diagnostika - Laktátová křivka, tělesná skladba, konzultace (opakované vyšetření do 12 měsíců) - (Cena: 1700 Kč)
- Vstupní vyšetření pohybového aparátu fyzioterapeutem - (Cena: 1250 Kč)
- Fyzioterapie - (Cena: 1000 Kč)
- Fyzioterapie pro děti - (Cena: 550 Kč)
- Analýza tělesné skladby + konzultace - (Cena: 300 Kč)
- Modelace ortopedických stélek - (Cena: 1800 Kč)
- Základní výživové poradenství - (Cena: 1600 Kč)
- Výživové poradenství s kontrolní konzultací - (Cena: 2900 Kč)
- Výživové poradenství se dvěma kontrolními konzultacemi a jídelníčkem na 3 dny - (Cena: 5800 Kč)

Obrázek 50 - Zdravotnické zařízení Dashboard – rezervace

```

// načtení Služeb do pole

$sql3 = "SELECT * FROM cSluzba";
$result3 = mysqli_query($conn, $sql3);
$sqlArray = array();
if(mysqli_num_rows($result3)>0){
    while($a = mysqli_fetch_assoc($result3)){
        $sqlArray[] = $a;
    }
}
<div class="modal-body">
    <div class="row">
        <div class="col-md-12">
            <form action="" method="post">
                <div class="form-group">
                    <label for="">Timeslot</label><br>
                    <input type="text" class="form-control" readonly
                </div>
                <div class="form-group">
                    <label for="">Email</label><br>
                    <input type="email" class="form-control" readonly
                </div>
                <div class="form-group">
                    <label for="">Jméno</label><br>
                    <input type="text" class="form-control" readonly
                </div>
                <div class="form-group">
                    <label for="">Služba</label><br>
                    <select name="sluzba" id="sluzba" class="form-
                        <?php
                            foreach($sqlArray as $option){
                                echo "<option
value='". $option['id'] ."'>". $option['nazev'] . " - (Cena: ". $option['cena'] . "
Kč)</option>";
                            }
                        ?>
                    </select>
                </div>
                <div class="form-group">
                    <button type="submit" class="btn float-right btn-
                </div>
            </form>
        </div>
    </div>
</div>
</div>

```

Víše je ukázka zdrojového kódu modální okna, které je použito pro zobrazení registračního formuláře. Po kliknutí na tlačítko *Odeslat*, je záznam o registraci uložen do databáze s hodnotou *isValidate = 0* a odeslán ke schválení lékaři.

4.3.7.2 Funkce *timeslots()*

Pro výpis všech volných slotů je použita funkce *timeslots()*, které jako parametry jsou předány proměnné *duration*, která udává jak dlouhý je časový úsek v minutách. Dalším parametrem je proměnná *cleanup*, ta udává na jakou hodnotu v minutách se funkce vrátí po uplynutí časového úseku, třetí a čtvrtou proměnnou je *start* a *end*, v těch je nastaveno od kolika do kolika je potřeba časové sloty vygenerovat.

```
$duration = 60;
$cleanup = 0;
$start = "08:00";
$end = "17:00";

function timeslots($duration,$cleanup,$start,$end){
    $start = new DateTime($start);
    $end = new DateTime($end);
    $interval = new DateInterval("PT".$duration."M");
    $cleanupInterval = new DateInterval("PT".$cleanup."M");
    $slots = array();

    for ($intStart = $start;$intStart<$end;$intStart->add($interval)-
>add($cleanupInterval)){
        $endPeriod = clone $intStart;
        $endPeriod->add($interval);
        if($endPeriod>$end){
            break;
        }
        $slots[] = $intStart->format("H:iA")." - ".$endPeriod-
>format("H:iA");
    }

    return $slots;
}
```

Jednotlivá modální okna, která se zobrazují po kliknutí na vybraný časový slot se otevírá díky JQuery scriptu, který je na stránce.

```
<script>
    $(".click").click(function(){
        var timeslot = $(this).attr('data-timeslot');
        $("#slot").html(timeslot);
        $("#timeslot").attr('value',timeslot);
        $("#myModal").modal("show");
    })
</script>
```

4.3.8 Rozhraní lékaře

Po přihlášení uživatele s nastavenou rolí lékaře do informačního systému se zobrazí obdobné prostředí jako pacientovi, jen s tím rozdílem, že lékař zde uvidí okno rozdělené na tři velké bloky. V prvním bloku uvidí všechny nepotvrzené budoucí rezervace, které čekají na potvrzení/zamítnutí. V druhém bloku se nachází všechny budoucí potvrzené rezervace. V posledním třetím bloku je souhrn všech potvrzených rezervací (budoucích i minulých)

Nepotvrzené rezervace 2						
Datum	E-mailová adresa	Jméno	Příjmení	Vyšetření	Slot	Akce
2020-11-09	tvrzlukas@icloud.com	Lukáš	Tvrz	Funkční diagnostika - Laktátová křivka, tělesná skladba, konzultace (opakované vyšetření do 12 měsíců)	11:00AM - 12:00PM	Potvrdit Zamítnout
2020-11-09	tvrzlukas@icloud.com	Lukáš	Tvrz	Poradenství v oblasti životního stylu	14:00PM - 15:00PM	Potvrdit Zamítnout

Budoucí rezervace 2						
Datum	E-mailová adresa	Jméno	Příjmení	Vyšetření	Slot	Akce
2020-11-20	tvrzlukas@icloud.com	Lukáš	Tvrz	Fyzioterapie	08:00AM - 09:00AM	Zrušit rezervaci
2020-12-03	servis@admin.cz	Admin	admin	Poradenství v oblasti životního stylu	12:00PM - 13:00PM	Zrušit rezervaci

Všechny rezervace 4						
Datum	E-mailová adresa	Jméno	Příjmení	Vyšetření	Slot	
2020-11-09	tvrzlukas@icloud.com	Lukáš	Tvrz	Funkční diagnostika - Laktátová křivka, tělesná skladba, konzultace (opakované vyšetření do 12 měsíců)	11:00AM - 12:00PM	
2020-11-09	tvrzlukas@icloud.com	Lukáš	Tvrz	Poradenství v oblasti životního stylu	14:00PM - 15:00PM	

Obrázek 51 – Zdravotnické zařízení Dashboard – rozhraní lékaře

V další části je názorně ukázané načítání z databáze s uloženými informacemi o rezervacích.

```
// SELECTY PRO DASHBOARD - LÉKAŘSKÁ SEKCE
```

```
// ZJIŠTĚNÍ POČTU NEPOTVRZENÝCH REZERVACÍ
```

```
$sql = "SELECT * FROM cRezervace as r JOIN cSluzba as s ON s.id = r.cSluzba_id JOIN cUzivatel as u ON u.id = r.cUzivatel_id JOIN cLekar as l ON l.id = r.cLekar_id WHERE r.date >= curdate() and isValidate = 0 and isDeleted = 0 ORDER BY date";  
$result = $conn->query($sql);  
$nepotvrzene_rezervace = $result->num_rows;
```

```
// ZJIŠTĚNÍ POČTU POTVRZENÝCH BUDOUCÍCH REZERVACÍ
```

```
$sql = "SELECT * FROM cRezervace as r JOIN cSluzba as s ON s.id =  
r.cSluzba_id JOIN cUzivatel as u ON  
u.id = r.cUzivatel_id JOIN cLekar as l ON l.id = r.cLekar_id  
WHERE r.date >= curdate() and isValidate = 1 and isDeleted = 0 ORDER BY  
date";  
$result = $conn->query($sql);  
$potvrzene_rezervace = $result->num_rows;
```

```
// ZJIŠTĚNÍ POČTU VŠECH REZERVACÍ V DATABÁZI
```

```
$sql = "SELECT * FROM cRezervace as r JOIN cSluzba as s ON s.id =  
r.cSluzba_id JOIN cUzivatel as u ON  
u.id = r.cUzivatel_id JOIN cLekar as l ON l.id = r.cLekar_id  
WHERE isDeleted = 0 ORDER BY date";  
$result = $conn->query($sql);  
$vsechny_rezervace = $result->num_rows;
```

Ukázka načítání dat do sekce nepotvrzený rezervací.

```
/*  
*****  
DASHBOARD - LÉKAŘ  
*****  
*/  
  
if($cRole_id == 2){  
  
    echo "<section id='dashboard-doctor' style='display: inline;'  
class='wrapper-index aboutus mt-4 mb-2 py-1'>  
    <div class='dashboard'><div class='box'><h2 class='text-left'>Nepotvrzené  
rezervace <span class='badge badge-pill badge-  
danger'>$nepotvrzene_rezervace</span></h2>";  
  
    if($nepotvrzene_rezervace > 0){  
        echo "<table class='table table-sm'>  
        <thead class='thead table-dark bg-dark'>  
        <tr>  
            <th scope='col' width='130px' >Datum</th>  
            <th scope='col'>E-mailová adresa</th>  
            <th scope='col'>Jméno</th>  
            <th scope='col'>Příjmení</th>  
            <th scope='col'>Vyšetření</th>  
            <th scope='col'>Slot</th>  
            <th scope='col' width='200px'>Akce</th>  
        </tr>  
        </thead>";  
        $sql = "SELECT * FROM cRezervace as r JOIN cSluzba as s ON s.id =  
r.cSluzba_id JOIN cUzivatel as u ON  
u.id = r.cUzivatel_id JOIN cLekar as l ON l.id = r.cLekar_id  
WHERE r.date >= curdate() and isValidate = 0 and isDeleted = 0  
ORDER BY date";  
        $result = $conn->query($sql);
```



```

if ($result->num_rows > 0){
    while ($row = $result->fetch_assoc()){
        echo
        "<tr><td>".$row['date'].</td><td>".$row['email'].</td><td>".$row['
jmeno' ].</td><td>".$row['prijmeni'].</td><td>".$row['nazev'].
        "</td><td>".$row['timeslot'].</td><td>".<div style='margin: 0
auto;'><a href='dashboard.php?potvrdit=".$row['id_rezervace']."'
class='btn badge badge-success'>Potvrdit</a>".
        " ".<a href='dashboard.php?zamitnout=".$row['id_rezervace']."'
class='btn badge badge-danger'>Zamitnout</a></div>".
        "</td></tr>";
    }
}
echo "</table></div>";

```

4.3.9 Rozhraní administrátora

Poslední varianta role uživatelského účtu v informačním systému je role administrátora. Administrátor po přihlášení získá přehledné informace o stavu databázových objektů, jaká data jsou uložena v tabulkách uživatelů, služeb a specializací lékařů. V případě části o uživateli je připravena možnost přes jednoduchý formulář přidávat uživatele do systému, primárně využitelné pro zadávání nových lékařů. Lékařům lze pak změnit případně přidat specializaci. Dále administrátor může přidávat data do statických číselníků nabízených služeb a lékařských specializací.

The screenshot displays two main sections of the administrator dashboard:

- Uživatelé 2**: A table listing users with columns for ID, Jmeno, Příjmení, Email, and Role. Two buttons are present: "Přidat nového uživatele" (red) and "Přiřadit lékaři specializaci" (yellow).
- Specializace 4**: A table listing specializations with columns for ID and Název. A button "Přidat nového specializaci" (blue) is located below the table.

ID	Jmeno	Příjmení	Email	Role
2	Jan	Novák	novakj@lekar.cz	Lékař Specializace: 1 - Fyzioterapie
3	Lukáš	Tvrz	tvrzlukas@icloud.com	Pacient

ID	Název
1	Fyzioterapie
2	Výživové poradenství
3	Funkční diagnostika
4	Životní styl

Obrázek 52 - Zdravotnické zařízení Dashboard – rozhraní administrátora

Služby 11

ID	Název	Cena	Patří do specializace
1	Poradenství v oblasti životního stylu	1200 Kč	Životní styl
2	Funkční diagnostika - Laktátová křivka, tělesná skladba, konzultace	1900 Kč	Funkční diagnostika
3	Funkční diagnostika - Laktátová křivka, tělesná skladba, konzultace (opakované vyšetření do 12 měsíců)	1700 Kč	Funkční diagnostika
4	Vstupní vyšetření pohybového aparátu fyzioterapeutem	1250 Kč	Fyzioterapie
5	Fyzioterapie	1000 Kč	Fyzioterapie
6	Fyzioterapie pro děti	550 Kč	Fyzioterapie
7	Analýza tělesné skladby + konzultace	300 Kč	Výživové poradenství
8	Modelace ortopedických stélek	1800 Kč	Fyzioterapie
9	Základní výživové poradenství	1600 Kč	Výživové poradenství
10	Výživové poradenství s kontrolní konzultací	2900 Kč	Výživové poradenství
11	Výživové poradenství se dvěma kontrolními konzultacemi a jídelníčkem na 3 dny	5800 Kč	Výživové poradenství

Přidat novou službu

Obrázek 53 - Zdravotnické zařízení Dashboard – rozhraní administrátora

Vložení nového záznamu do tabulky [cUživatel]

Emailová adresa

Titul

Jméno

Příjmení

Heslo

Kontrola hesla

Adresa

Město

Stát

PSČ

Telefon

Typ účtu

Vložit

Zpět

Obrázek 54 - Zdravotnické zařízení Dashboard – přidání záznamu do tabulky cUživatel

Přiřazení specializace lékaři

Zvolit lékaře

Přiřadit specializaci

Vložit

Zpět

Obrázek 55 - Zdravotnické zařízení Dashboard – přiřazení lékaři specializaci

Vložení nového záznamu do tabulky [cSluzba]

Název služby

Cena

Přiřadit specializaci

Fyzioterapie

Vložit Zpět

Obrázek 56 - Zdravotnické zařízení Dashboard – vložení nové služby

Při vkládání nových specializací či služeb, je nejprve zjišťováno, zda obdobná služba se stejným názvem již neexistuje, pokud ještě neexistuje je vložena do databáze.

Níže je ukázka načítání dat do sekce Uživatelé, v případě že se jedná o uživatele s rolí *Pacient*, do tabulky se vloží informace o přidělené roli, v případě lékaře se načte z tabulky specializací i jeho specializace.

```
if($cRole_id == 1){
```

```
ECHO "<section id='dashboard-admin' class='wrapper-index aboutus mt-4 mb-2 py-1'>";
```

```
ECHO "<div class='container'><div class='box'><h2 class='text-left'>Uživatelé <span class='badge badge-pill badge-danger'>$num_users</span></h2>";
```

```
if($num_users > 0){
```

```
ECHO "<table class='table table-sm'>
<thead class='thead table-dark bg-dark'>
<tr>
<th scope='col'>ID</th>
<th scope='col'>Jmeno</th>
<th scope='col'>Příjmení</th>
<th scope='col'>Email</th>
<th scope='col' width='300px'>Role</th>
</tr>
</thead>";
```

```
$sql = "SELECT u.id as `uid`,u.jmeno,u.prijmeni,u.email,u.cRole_id,s.id
as
`sid`,s.nazevSpecializace FROM cUzivatel as u
LEFT JOIN cLekar as l ON l.cUzivatel_id = u.id
LEFT JOIN cSpecializace as s ON s.id = l.cSpecializace_id
WHERE cRole_id in ('2','3')
ORDER BY uid ASC";
$result = $conn->query($sql);
```

```

if ($result->num_rows > 0){
    while ($row = $result->fetch_assoc()){
        echo
            "<tr><td>".$row['uid'].</td><td>".$row['jmeno'].</td><td>".
$row['prijmeni'].</td><td>".$row['email'].</td><td>";
        if ($row['cRole_id'] == 2){
            echo "<span class='badge badge-warning'>Lékař</span>".
"."Specializace: ".$row['sid']." - ".$row['nazevSpecializace'];
        }elseif ($row['cRole_id'] == 3){
            echo "<span class='badge badge-primary'>Pacient</span>";
        }
        echo "</td></tr>";
    }
    echo "</table>";
} else {
    echo "<p>V databázi se nenachází žádný uživatel</p>";
}
ECHO "<a href='insert.php?insert=1' class='btn btn-danger'>Přidat nového
uživatele</a>";
echo " ";
ECHO "<a href='insert.php?insert=4' class='btn btn-warning'>Přiřadit
lékaři specializaci</a>";

echo "</div>";

```

4.3.10 Nastavení osobních údajů

Ve všech třech typech uživatelských účtů se jako třetí položka v menu zobrazuje tlačítko Nastavení, které slouží pro přístup ke stránce s veškerým nastavením uživatelského účtu. Po vstupu se zobrazí formulář s načtenými daty z databáze. Uživatel může změnit veškeré své již zadané údaje vyjma emailové adresy a role v systému. Ty jsou přístupné pouze pro čtení.

Nastavení uživatelského účtu

Emailová adresa

tvrzlukas@icloud.com

Titul	Jméno	Příjmení
Bc.	Lukáš	Tvrz
Adresa	Město	Stát
Bílinská 515	Praha	Česká Republika
PSČ	Telefon	Role
19000	606067945	Pacient

Uložit změny

Obrázek 57 - Zdravotnické zařízení – nastavení uživatelského účtu

Po řádném vyplnění všech povinných políček a stisknutí tlačítka Uložit změny, systém uloží veškeré změněné údaje do databáze.

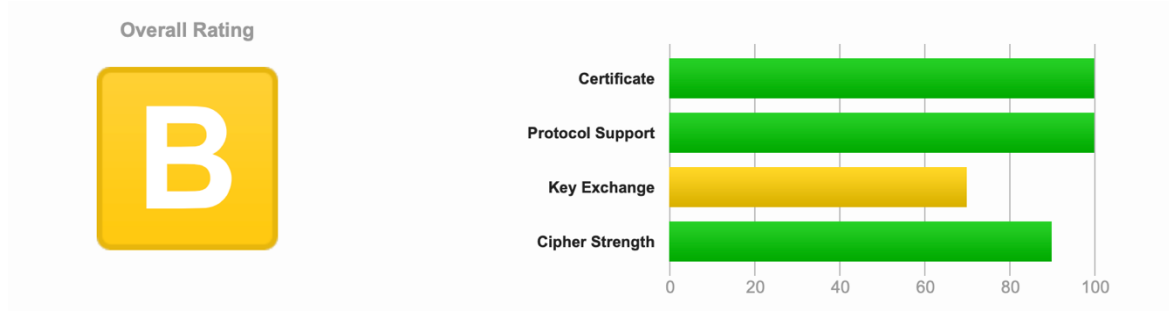
```
if(isset($_POST['save'])){
    $user = $u_id;
    $titul = mysqli_real_escape_string($conn, ($_POST['titul']));
    $jmeno = mysqli_real_escape_string($conn, ($_POST['jmeno']));
    $prijmeni = mysqli_real_escape_string($conn,
($_POST['prijmeni']));
    $adresa = mysqli_real_escape_string($conn, ($_POST['adresa']));
    $telefon = mysqli_real_escape_string($conn, ($_POST['telefon']));
    $mesto = mysqli_real_escape_string($conn, ($_POST['mesto']));
    $stat = mysqli_real_escape_string($conn, ($_POST['stat']));
    $psc = mysqli_real_escape_string($conn, ($_POST['psc']));

    $sql = "update cUzivatel set
        titul = '$titul',
        jmeno = '$jmeno',
        prijmeni = '$prijmeni',
        adresa = '$adresa',
        telefon = '$telefon',
        psc = '$psc',
        mesto = '$mesto',
        stat = '$stat'
        where id = $u_id";
    if ($conn->query($sql) === TRUE){
        $msg = "<div class='alert alert-success text-center'>Změny
úspěšně uloženy do databáze</div>";
        //header("Location: settings.php");
    }
}
```

4.4 Testování

4.4.1 Zabezpečení

Nasazení webových stránek proběhlo na webovém serveru s SSL vrstvou, pro otestování základního zabezpečení, a především kvality SSL vrstvy byl zvolen online nástroj SSL Labs. Tento nástroj webovou stránku prověřil z hlediska síly jednotlivých šifer, které se používají při komunikaci a ohodnotil zabezpečení aplikace od nuly od sta procent.

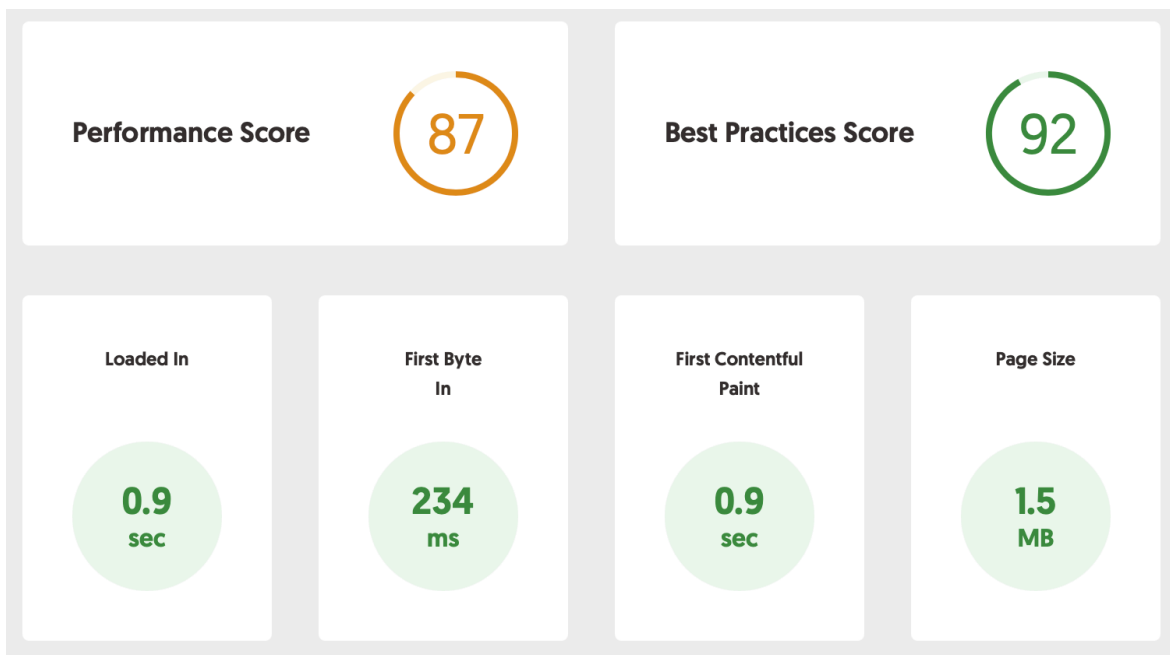


Obrázek 58 - Výsledky testu zabezpečení (zdroj: www.ssllabs.com)

Jelikož pro potřeby této diplomové práce byl zvolen jen základní SSL certifikát, který je hostingovou společností vydáván zdarma, nejsou výsledky stoprocentní. Jsou zde rezervy v oblasti využívání starších druhů šifer, které jsou podle SSL labs možnou potenciální hrozbou. Celkově, ale výsledky testu zabezpečení jsou velmi dobré. Z hlediska zabezpečení stránek je dále implementována kontrola proti robotům captcha, jedná se o Turingův test, který na webu používá ve snaze odlišit skutečné uživatele od robotů. Do databáze uživatelů je heslo dále ukládáno zašifrované hashovací šifrou. Heslo musí navíc splňovat bezpečnostní náležitosti, musí obsahovat minimálně 8 znaků, musí obsahovat jedno velké písmeno a číslici, v poslední řadě také jeden speciální znak. Jelikož se jedná o informační systém pro nestátní společnost, jsou v databázi uchovány jen nejpotřebnější osobní údaje, které ukončením platnosti uživatelského účtu/smazáním uživatelského účtu se v databázi dále neuchovávají.

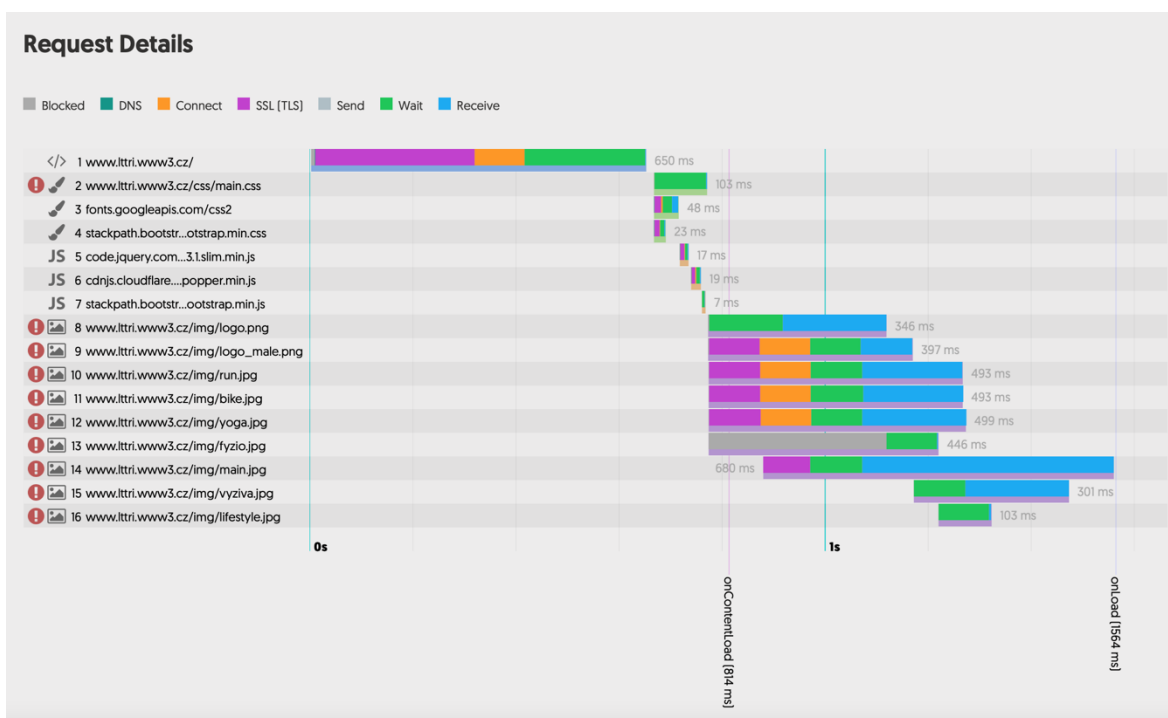
4.4.2 Test rychlosti

Ověřování rychlosti odezvy webových stránek bylo testováno pomocí online nástroje Geekflare, který se zaměřuje na celkový objem přenášených dat mezi aplikací a koncovým uživatelem.



Obrázek 59 - Výsledky testu rychlosti (zdroj: www.gf.dev)

Vyvíjený informační systém dosáhl dobrého rychlostního hodnocení. Potenciální prostor pro zlepšení je ve výběru poskytovatele webového serveru, který poskytuje http/2 protokol, dalším doporučením je nevyužívat jQuery knihovny, které ale jsou pro běh systému důležité.



Obrázek 60 - Výsledky testu rychlosti (zdroj: www.gf.dev)

5 Výsledky a diskuse

Cílem této diplomové práce bylo vytvořit webový informační systém pro nestátní zdravotnické zařízení se sportovním zaměřením. Dle požadavků na funkčnost informačního systému, které byly specifikovány v analytické části byl tento cíl splněn. Byl vytvořen informační systém, do kterého mají přístup tři typy uživatelů. Pacient, který k informačnímu systému přistupuje za účelem vytvoření rezervace na vyšetření/službu má k dispozici přehledný systém svých rezervací, kde vidí, zda jeho rezervace byla potvrzena či zamítnuta. Lékař po vstupu do své části systému přehledně vidí všechny budoucí rezervace všech uživatelů napříč celým informačním systémem. Posledním uživatelem systému je hlavní administrátor, který má přehled nad hlavními tabulkami systému, do kterých může přidávat nové záznamy.

Výsledné stránky jsou optimalizovány pro provoz na desktopových zařízeních a mobilních zařízeních. Veškerá data v tomto informačním systému byla vyplněna pouze pro zobrazení funkčnosti systému.

6 Závěr

V teoretické části této diplomové práce byl čtenář seznámen se základními pojmy, kterými se tato práce zabývá. Byl zde popsán informační systém, z hlediska jednotlivých typů informačních systémů, metodik projektování informačních systému a jaké CASE nástroje se při projektování využívají. Dále se zde dočte o životním cyklu vývoje IS, z metodického hlediska a reálné využitelnosti pro vývoj software. Další kapitolou, se kterou je seznámen, je seznámení s jazykem UML, který se využívá pro vizualizaci, specifikaci a konstrukce objektů softwarového systému. Jsou zde detailně popsány veškeré diagramy, které jsou poté v praktické části práce využity. Poslední kapitola teoretické části se zabývá technologiemi použitými pro vývoj informačního systému, mezi ně patří například značkovací jazyk HTML, či kaskádové styly.

Druhou část práce se skládá z analýzy problému, vytyčení důležitých cílů při vývoji systému a vytvoření logického návrhu informačního systému. Nalezneme zde jednotlivé scénáře užití systému, které říkají, jak se systém chová v interakci s uživatelem, dále datový slovník pro náhled uspořádání databáze, sekvenční diagram a diagram tříd.

V poslední části diplomové části je již sestaven samotný informační systém. Jen navržen za pomoci frameworku Bootstrap, který je pro potřeby aplikace upraven. Pro tvorbu dynamického obsahu a celé klientské části je zvolen jazyk PHP, který také slouží pro komunikaci s databází, která v tomto případě byla zvolena MySQL.

Autor se rozhodl pro tvorbu takto zaměřeného informačního systému z důvodu, že v současné době většina obdobně zaměřujících se středisek, vůbec podobným systémem nedisponuje, a stále se většinou ve veliké míře můžeme setkat s objednávkovými diáři, i když moderní technologie by to oběma stranám velice zjednodušila a zefektivnila.

7 Seznam použitých zdrojů

- [1] **POŽÁR, J.** 2005. *Informační bezpečnost*, Vydavatelství a nakladatelství Aleš Čeněk, s.r.o., ISBN 80-86898-38-5
- [2] **ČIHOVSKÁ A KOL.,** 2005. *Management obchodní organizace*, Vydavatelství Ekonom, ISBN 80-225-1930-8
- [3] **BERTALANFFY L.,** 1969. *General system theory*, ISBN 978-0807604533
- [4] **ŽÁČEK, J.** 2012. *Informační systémy I* <online> Dostupné z: <https://www1.osu.cz/~zacek/6inf2/XINF1.pdf> [cit. 2020-02-02]
- [5] **GRANT, M.** 2019 – *SWOT analysis* <online> Dostupné z: <https://www.investopedia.com/terms/s/swot.asp> [cit. 2020-02-02]
- [6] **BRUCKNER, T., VOŘÍŠEK, J., BUCHALCEVOVÁ, A.** 2012. *Tvorba informačních systémů*. Praha : Grada. ISBN 978-80-247-4153-6
- [7] **PROCHÁZKA, J., KLIMEŠ, C.** 2009 – *Softwarové inženýrství*, druhé vydání, Ostravská univerzita v Ostravě
- [8] **ŠMÍD, V.,** *Management informačního systému* <online> Dostupné z: <https://www.fi.muni.cz/~smid/mis-ziveyk.htm> [cit. 2020-02-16]
- [9] **HTML Tutorial,** 2020. *W3Schools* <online> Dostupné z: <https://www.w3schools.com/html> [cit. 2020-09-01]
- [10] **W3.org, 1988 HTML5 Overview,** Michael Smith <online> Dostupné z: <http://w3.org/2008/Talks/04-24-smith/index.html>
- [11] **KOSEK, J.:** *HTML. Tvorba dokonalých WWW stránek*, Grada,1998, ISBN 80-7169-608-0
- [12] **STANÍČEK, P.:** *CSS Kaskádové styly Kompletní průvodce*. 2003. 192 s. ISBN 80-7226-872-4.
- [13] **STEJSKAL, Jan.** *Vytváříme WWW stránky pomocí HTML, CSS a JavaScriptu*. 2004. 256 s. ISBN 80-2510-167-3
- [14] **WELLING, L. a THOMSON, L.:** *PHP a MySQL – rozvoj webových aplikací*. Praha: SoftPress, 2002. 720 s. ISBN 80-86497-20-8.
- [15] **RUMBAUGH J, JACOBSON I, BOOCH G. :** *The Unified Modeling Language Reference Manual*. USA: Addison-Wesley, 2. edice 2004
- [16] **KANISOVÁ, H.; MÜLLER, M.** *UML srozumitelně. 2., aktualiz. vyd.* Brno: Computer Press, 2006. ISBN 8025110834.

- [17] **ARLOW, Jim a Ila NEUSTADT.** *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky.* 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [18] **ČÁPKA, David.** Lekce 1 - Úvod do UML. *ITnetwork* [online]. ©2020 [cit. 2020-07-11]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-uvod-historie-vyznam-a-diagramy/>
- [19] **ČÁPKA, David.** Lekce 2 - UML - Use Case Diagram. *ITnetwork* [online]. ©2020 [cit. 2020-07-28]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-use-case-diagram>
- [20] Historie a vývoj HTML. *HTML5* [online]. Jirka Kosek, c1997-2014 [cit. 2020-07-05]. Dostupné z: <http://htmlguru.cz/uvod-historie.html>
- [21] **HLAVA, Tomáš.** Vodopádový model. *Testování softwaru* [online]. 2016 [cit. 2020-07-21]. Dostupné z: <http://testovanisoftware.cz/manualni-testovani/modely-zivotniho-cyklu-softwaru/vodopadovy-model/>
- [22] Vodopádový model (Waterfall model). *Management Mania* [online]. 2015 [cit. 2020-07-05]. Dostupné z: <https://managementmania.com/cs/vodopadovy-model-waterfall-model>
- [23] Agilní metodiky řízení vývoje software (Agile software development methodologies). *Management Mania* [online]. 2016 [cit. 2020-07-05]. Dostupné z: <https://managementmania.com/cs/agilni-metodiky-rizeni-vyvoje-software>
- [24] **BECK, Kent.** *Extrémní programování.* Praha: Grada, 2002. ISBN 9788024703008.
- [25] **KOLESNIKOV, Pavel.** *Extrémní programování v praxi.* *Interval* [online]. 2002, [cit. 2020-07-11]. Dostupné z: <https://www.interval.cz/clanky/extremni-programovani-v-praxi/>
- [26] **KNESL, Jiří.** Co je Scrum? Jiří Knesl [online]. 2016 [cit. 2020-07-08]. Dostupné z: <http://www.knesl.com/co-je-scrum>
- [27] **ČÁPKA, David.** Lekce 5 - UML - Class diagram. *ITnetwork* [online]. ©2020 [cit. 2020-07-11]. Dostupné z: <https://www.itnetwork.cz/navrh/uml/uml-class-diagram-tridni-model>

- [28] **ČÁPKA, David.** Lekce 7 - UML – Activity diagram. *ITnetwork* [online]. ©2020 [cit. 2020-07-11]. Dostupné z:
<https://www.itnetwork.cz/navrh/uml/uml-activity-diagram>
- [29] Introduction to Databases E-R Data Modeling [online]. [cit. 2020-08-01]
Dostupné z: <http://labe.felk.cvut.cz/~stepan/AE3B33OSD/Lesson08-IntroDatabases.pdf>
- [30] **MOCK, Liza.** UML History & Use Cases [online]. [cit. 2020-07-22].
Dostupné z: <https://www.gliffy.com/blog/uml-history-use-cases>
- [31] **CASTRO, Elizabeth; HYSLOP, Bruce.** HTML5 a CSS3: názorný průvodce tvorbou WWW stránek. Brno: Computer Press, 2012. ISBN 978-80-251-3733-8.
- [32] **JANOVSKÝ, Dušan.** Jakpsatweb.cz. Jak psát web [online]. [cit. 2020-03-21].
Dostupné z: <https://www.jakpsatweb.cz/css/css-uvod.html>

8 Seznam obrázků

Obrázek 1 - SWOT analýza (Zdroj: https://zsf.cz/show/swot-analyza-rychleho-nasazeni).	16
Obrázek 2 - Model vodopád Zdroj: https://cs.wikipedia.org/wiki/Vodopádový_model	18
Obrázek 3 - Model spirála - Zdroj: https://cs.wikipedia.org/wiki/Metodika_v%C3%BDvoje_softwaru	19
Obrázek 4 - Prototypový model - Zdroj: https://www.fi.muni.cz/~smid/mis-zivcyk.htm ..	20
Obrázek 5 - Schéma agilního vývoje <Zdroj> https://m.systemonline.cz/it-pro-banky-a-financni-organizace/zkusenosti-ceskych-bank-s-agilnim-vyvojem.htm	21
Obrázek 6 - příklad strukturální abstrakce <Zdroj> www.slideplayer.cz/slide/2938240	22
Obrázek 7 - příklad předmětu chování <Zdroj> www.slideplayer.cz/slide/2938240	22
Obrázek 8 - příklad seskupení <Zdroj> www.slideplayer.cz/slide/2938240	22
Obrázek 9 - příklad poznámky <Zdroj> www.slideplayer.cz/slide/2938240	23
Obrázek 10 – Vztah asociace, agregace, kompozice <Zdroj> www.slideplayer.cz/slide/2938240	23
Obrázek 11 - Vztah zobecnění <Zdroj> www.slideplayer.cz/slide/2938240	23
Obrázek 12 - Vztah závislosti <Zdroj> www.slideplayer.cz/slide/2938240	24
Obrázek 13 - Vztah realizace <Zdroj> www.slideplayer.cz/slide/2938240	24
Obrázek 14 - Přehled všech UML diagramů a jejich hierarchie <Zdroj> https://commons.wikimedia.org/wiki/File:UML_diagrams_overview.svg	25
Obrázek 15 - Ukázka diagramu tříd <Zdroj> https://www.uml-diagrams.org/software-licensing-domain-diagram-example.html	26
Obrázek 16 - Ukázka diagramu aktivit <Zdroj> https://cs.wikipedia.org/wiki/Diagram_aktivit	26
Obrázek 17 - Ukázka State Machine diagramu <Zdroj> http://agilemodeling.com/artifacts/stateMachineDiagram.htm	27
Obrázek 18 - Ukázka ERD diagramu <Zdroj> https://www.visual-paradigm.com/VPGallery/datamodeling/EntityRelationshipDiagram.html	28
Obrázek 19 - ukázka Use Case diagramu <online> Zdroj: https://en.wikipedia.org/wiki/Use_case_diagram	29
Obrázek 20 - Zdravotnické zařízení - diagram užití	36
Obrázek 21 - Activity diagram	41
Obrázek 22 - Class diagram.....	42
Obrázek 23 - Sekvenční diagram: Potvrzení rezervace	43
Obrázek 24 - Sekvenční diagram: Vytvoření nové rezervace	43
Obrázek 25 - Entitně relační diagram	46
Obrázek 26 - Wireframe 1: Úvodní obrazovka	47
Obrázek 27 - Wireframe 2: Rezervační kalendář	47
Obrázek 28 - Wireframe 3: Přihlašovací obrazovka.....	48
Obrázek 29 - Zdravotnické zařízení - Nastavení databázového serveru	50
Obrázek 30 - Zdravotnické zařízení - Nastavení webového serveru	50
Obrázek 31 - Zdravotnické zařízení - Seznam využitých tabulek v databázi.....	51

Obrázek 32 - Zdravotnické zařízení - Menu (nepřihlášený uživatel).....	53
Obrázek 33 - Zdravotnické zařízení - Menu (přihlášený uživatel).....	54
Obrázek 34 - Zdravotnické zařízení - Úvodní strana.....	54
Obrázek 35 - Zdravotnické zařízení - Úvodní stránka - vstup do aplikace.....	54
Obrázek 36 - Zdravotnické zařízení - Úvodní stránka (část - Služby).....	55
Obrázek 37 - Zdravotnické zařízení - Úvodní stránka (část - Služby)	55
Obrázek 38 - Zdravotnické zařízení - Úvodní stránka (část - Ceník).....	56
Obrázek 39 - Zdravotnické zařízení - Úvodní stránka (část - Kontakt)	56
Obrázek 40 - Zdravotnické zařízení - Přihlašovací obrazovka	57
Obrázek 41 - Zdravotnické zařízení - Nová registrace	57
Obrázek 42 - Zdravotnické zařízení - Nová registrace (kontrola na existenci uživatele)...	58
Obrázek 43 - Zdravotnické zařízení - Nová registrace	58
Obrázek 44 - Výňatek z dokumentu GDPR.....	64
Obrázek 45 - Zdravotnické zařízení Dashboard - pacient.....	65
Obrázek 46 - Zdravotnické zařízení - výběr lékaře.....	65
Obrázek 47 - Zdravotnické zařízení Dashboard - rezervační kalendář.....	66
Obrázek 48 - Zdravotnické zařízení Dashboard - rezervační kalendář.....	67
Obrázek 49 - Zdravotnické zařízení Dashboard - rezervace	68
Obrázek 50 - Zdravotnické zařízení Dashboard - rezervace	68
Obrázek 51 - Zdravotnické zařízení Dashboard - rozhraní lékaře	71
Obrázek 52 - Zdravotnické zařízení Dashboard - rozhraní administrátora.....	73
Obrázek 53 - Zdravotnické zařízení Dashboard - rozhraní administrátora.....	74
Obrázek 54 - Zdravotnické zařízení Dashboard - přidání záznamu do tabulky cUživatel .	74
Obrázek 55 - Zdravotnické zařízení Dashboard - přiřazení lékaři specializaci	74
Obrázek 56 - Zdravotnické zařízení Dashboard - vložení nové služby	75
Obrázek 57 - Zdravotnické zařízení - nastavení uživatelského účtu.....	76
Obrázek 58 - Výsledky testu zabezpečení (zdroj: www.ssllabs.com).....	78
Obrázek 59 - Výsledky testu rychlosti (zdroj: www.gf.dev).....	79
Obrázek 60 - Výsledky testu rychlosti (zdroj: www.gf.dev).....	79

9 Zdrojový kód

Kompletní zdrojový kód a veškeré vyvíjené komponenty jsou k dispozici na GitHubu
autora: <https://github.com/tvrzluke/diplomovaPrace/tree/master>.