TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Webová aplikace pro Dětský koutek TUL

Bakalářská práce

Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie

Autor práce: **Kristián Jacik**
Vedoucí práce: Ing. Jan Kolaja, Ph.D.
Ústav nových technologií a aplikované informatiky

Liberec 2021

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Zadání bakalářské práce

Webová aplikace pro Dětský koutek TUL

Jméno a příjmení: **Kristián Jacik**
Osobní číslo: M17000078
Studijní program: B2646 Informační technologie
Studijní obor: Informační technologie
Zadávající katedra: Ústav nových technologií a aplikované informatiky
Akademický rok: **2020/2021**

Zásady pro vypracování:

Vytvořte webovou aplikaci, která bude umožňovat správu rezervací pro Dětský koutek TUL.

1. Analyzujte současný stav běžící aplikace koutek.tul.cz společně se správci koutku a navrhněte funkčnost aplikace nové.
2. Na přiděleném virtuálním serveru zprovozněte aplikaci, která bude odpovídat požadavkům pracovníků Dětského koutku TUL a bude splňovat následující kritéria:
   a) Autentizace uživatele sítě LIANE, ověření přes SHIBBOLETH.
   b) Uživatelské role Rodič, Chůva, Správce.
   c) Základní možnosti uživatele Rodič: Přidávání a správa dětí, výběr volných termínů.
   d) Základní možnosti uživatele Chůva: Zadávání volných časů, editace vybraných vlastností dítěte, potvrzování požadovaných termínů.
   e) Základní možnosti uživatele Správce: Nejvyšší práva, editace chův, dětí, termínu a podobně.
3. Aplikaci otestujte ve zkušebním provozu. V případě úspěšného testovacího provozu bude aplikace nasazena pro reálný provoz.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30 – 40 stran
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Čeština

Seznam odborné literatury:

[1] ROSEBROCK, Eric a Eric FILSON. *Linux, Apache, MySQL a PHP: instalace a konfigurace prostředí pro pokročilé webové aplikace*. Praha: Grada, 2005. Průvodce. ISBN 80-247-1260-1.
[2] POKORNÝ, Jaroslav a Michal VALENTA. *Databázové systémy: vysokoškolská učebnice*. Praha: České vysoké učení technické v Praze, 2013. ISBN 978-80-01-05212-9.
[3] PAVLÍČEK, Antonín. *Nová média a sociální sítě*. Praha: Oeconomica, 2010. ISBN 978-80-245-1742-1.

Vedoucí práce:	Ing. Jan Kolaja, Ph.D. Ústav nových technologií a aplikované informatiky
Datum zadání práce:	19. října 2020
Předpokládaný termín odevzdání:	17. května 2021

L.S.

prof. Ing. Zdeněk Plíva, Ph.D.
děkan

Ing. Josef Novák, Ph.D.
vedoucí ústavu

V Liberci dne 19. října 2020

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracoval samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědom toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědom následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

16. května 2021 Kristián Jacik

Webová aplikace pro Dětský koutek TUL

Abstrakt

Předmětem této bakalářské práce je tvorba webové aplikace rezervačního systému pro Dětský koutek Technické univerzity v Liberci, umožňující vytváření žádostí o hlídání pro děti, potvrzování rezervací přidělením k asistentům a generování výpisů nadcházejících a provedených hlídání pro asistenty a správce. V rešeršní části práce je analyzovaný stav původního řešení rezervačního systému. Práce se dále zabývá volbou použitých technologií, popisem technického řešení, návrhem databázového modelu a popisem uživatelského prostředí.

Klíčová slova: Dětský koutek, rezervační systém, rezervace, Javascript, Node.js, Express.js, EJS, MariaDB, MySQL, databáze, MVC, webová aplikace

Web Application for University's Nursery

Abstract

The subject of this bachelor thesis is the creation of the reservation system web application for the Technical university of Liberec Day-care nursery. The web application lets users to create requests for babysitting their children, administrators to confirm created reservations by assigning them to assistants and generate summaries of upcoming and past reservations. The research part analyses the current reservation system solution. The thesis describes used technologies, technical solution, database model design and describes the user interface of the web application.

Keywords: Day-care nursery, reservation system, reservation, Javascript, Node.js, Express.js, EJS, MariaDB, MySQL, database, MVC, web application

Poděkování

Děkuji svému vedoucímu práce Ing. Janu Kolajovi, Ph.D. za cenné rady, odborné vedení této práce a pomoc při jejím vypracování. Zároveň bych rád poděkoval Mgr. Iloně Sovové, za příležitost tvorby této webové aplikace, která může dále fungovat a skutečně sloužit studentům a zaměstnancům Technické univerzity v Liberci. V poslední řadě bych velmi rád poděkoval své rodině a svým přátelům za jejich podporu.

Obsah

Seznam zkratek 9

1 Úvod **10**
1.1 Dětský koutek TUL 10

2 Rešerše existujících řešení **12**
2.1 Existující rezervační systémy 12
2.1.1 Easy!Appointments 12
2.2 Rezervační systém pro antigenní testování zaměstnanců TUL 13
2.3 Aktuální řešení Dětského koutku 14

3 Návrh webové aplikace **16**
3.1 Požadavky na systém 16
3.2 Použité technologie 17
3.2.1 Client-side technologie 17
3.2.2 Server-side technologie 18
3.3 Databázový model 21
3.4 Role uživatele 23
3.4.1 Rodič 24
3.4.2 Asistent 25
3.4.3 Správce 26
3.5 Funkce aplikace 27
3.5.1 Potvrzení rezervace a přidělení rezervace asistentovi 28
3.5.2 Odesílání potvrzujících emailů 28
3.5.3 Generování CSV výpisu 29
3.5.4 Zobrazení informačních polí na hlavní stránce 29

4 Implementace webové aplikace **31**
4.1 Routování požadavků 31
4.1.1 Provádění uživatele vytvářením rezervace 32
4.1.2 Sessions 34
4.2 Templatovací engine 34
4.3 Public resources 34
4.4 AJAX requesty 35
4.5 Příprava serveru 36
4.5.1 Instalace MariaDB 36

4.5.2 Instalace Let's Encrypt SSL certifikátu 37
4.5.3 Instalace Node.js a spuštění rezervačního koutku 37
4.6 Ukázka implementace 38

5 Testování aplikace **40**
5.1 Automatizované testy 40
5.2 Validita vstupních dat 41
5.3 Test kompatibility prohlížečů 41

6 Závěr **43**

Použitá literatura **44**

Seznam obrázků **47**

Příloha 1: Adresářová struktura aplikace **48**

Příloha 2: Zdrojový kód **49**

Seznam zkratek

AJAX	Asynchronous JavaScript and XML
BOM	Byte Order Mark
CRUD	Create, Read, Update, Delete
CSP	Content Security Policy
CSS	Cascading Style Sheets
CSV	Comma-separated values
DDL	Data Definition Language
DK	Dětský koutek TUL
DQL	Data Query Language
EJS	Embedded Javascript
ER diagram	Entitně-relační diagram
HTML	Hyper Text Markup Language
JSON	Javascript Object Notation
LIANE	Liberec Academic Network
MVC	Model-View-Controller
NF	Normální forma
REST	Representational State Transfer
SMTP	Simple Mail Transfer Protocol
SSO	Single Sign-On
SŘBD	Systém řízení báze date
SQL	Structured Query Language
TUL	Technická univerzita v Liberci
URL	Uniform Resource Locator
WYSIWYG	What You See Is What You Get
XSS	Cross Site Scripting

1 Úvod

Téma této bakalářské práce vzniklo z potřeby aktualizace existujícího rezervačního systému Dětského koutku Technické univerzity v Liberci, jehož funkcionalita, vzhledem k požadavkům vedení a uživatelů Dětského koutku, přestávala být dostačující. Cílem této práce tak bylo vytvořit aktualizovanou webovou aplikaci, která splňuje požadavky dnešních standardů, bude uživatelsky jednoduchá a usnadní uživatelům a především vedení správu rezervací na hlídání dětí v Dětském koutku.

Existuje velký počet rezervačních systémů, které častokrát nabízí širokou funkcionalitu, ale i přesto nemusí jejich funkčnost odpovídat požadavkům zadavatele. Toto byl impulz k vytvoření vlastního rezervačního systému, který bude přesně splňovat požadavky Dětského koutku. Umožní třem druhům uživatelů (Rodič / Chůva / Správce) spravovat a zobrazovat rezervace hlídání dětí ve svých rozhraních, která zobrazí relevantní informace, usnadní především rodičům vytváření rezervací dětí a do budoucna bude případně snadno rozšiřitelná a spravovatelná, vzhledem k použitému návrhu aplikace a použitým nástrojům.

Celé řešení rezervačního systému tak používá volně dostupných technologií, jako Node.js s knihovnou Express.js pro serverovou část nebo snadno upravitelný framework Bootstrap pro uživatelskou část, který zajišťuje responzivitu webového prostředí pro všechny velikosti zařízení od desktopových prohlížečů, po mobilní telefony.

První část této práce se zaměřuje na existující řešení rezervačních systémů a jejich srovnání s potřebami rezervačního systému Dětského koutku. Druhá část práce se věnuje samotnému návrhu a implementaci řešení webové aplikace, od použitých technologií, přes požadavky na systém, po použitý databázový model. Součástí je i popis přípravy serveru a instalace webové aplikace pro provoz. Třetí část se věnuje testování aplikace.

1.1 Dětský koutek TUL

Dětský koutek na Fakultě přírodovědně-humanitní a pedagogické Technické univerzity v Liberci, taktéž nazývaný jako Dětský koutek TUL, je místem, kde si mohou rodiče z řad zaměstnanců Technické univerzity v Liberci nebo rodiče z řad studentů TUL, nechat pohlídat své děti chůvami (v této práci a aplikaci jsou nazývané taktéž jako asistentky / asistenti).[1]

Cílem projektu je umožnit rodičům a zaměstnancům sladit výkon profese nebo studia společně s rodičovstvím. Klíčovou aktivitou dětského koutku je zajištění

krátkodobé péče o děti a výchovně vzdělávací aktivity.[2] Nejde však o předškolní zařízení, či jiné obdobné zařízení, které provozuje teoretické, či praktické vyučování nebo výchovu dítěte.[1] K těmto účelům provozuje univerzita zaměstnaneckou mateřskou školu s prvky waldorfské pedagogiky ŠkaTULka.[3]

2 Rešerše existujících řešení

Tato kapitola se zaměřuje na existující rezervační systémy, které by mohly částečně vyhovovat požadavkům Dětského koutku, protože nabízejí podobnou funkcionalitu nebo jsou předchůdcem nově vytvořeného řešení.

2.1 Existující rezervační systémy

Na trhu se vyskytuje nespočet řešení rezervačních systémů, od placených, po bezplatné nebo open source. Některá řešení jsou provozovaná v cloudu, tedy u poskytovatelů jednotlivých systémů, nebo si lze celé řešení hostovat na vlastním hostingovém řešení. Velké množství těchto systémů je však zaměřeno na booking hotelových pokojů nebo apartmánů, či rezervace zájezdů, aut, míst v restauraci nebo fungují jako objednávkový systém pro lékaře nebo úředníky.[4] Mezi tyto systémy patří například *Cloudbeds*, *Reservanto* nebo český *Reservatic*, který je především známý pro využití jako Centrální rezervační systém - Očkování proti COVID-19. Zároveň slouží pro další rezervace do lékařských služeb.[5]

Všechny uvedené příklady spojuje fakt, že rezervační systémy pracují s objektem (místo v restauraci / hotelový pokoj / předmět inventáře), který byl předem vytvořen ze strany poskytovatele služby. Uživatel tak nevytváří vlastní objekt, pro který by následně rezervaci vytvořil. Rezervační systém dětského koutku navíc počítá s průnikem dvou různých časů, který je nutný pro správce znát.

2.1.1 Easy!Appointments

Pro srovnání se službou, která je veřejně k dispozici k použití, jsem zvolil open source aplikaci Easy!Appointments[1].

Důvodem volby je subjektivně největší podobnost se systémem pro Dětský koutek TUL. Kritérii pro výběr byla ucelenost celého balíku jako jedna funkční aplikace, jednoduchost uživatelského prostředí, popularita daného řešení a dostupnost zdrojového kódu pro bezplatné využití aplikace.

Aplikace Easy!Appointments je postavená v programovacím jazyce PHP, v uživatelském prostředí využívá framework Bootstrap a knihovnu FullCalendar pro zobrazení interaktivního kalendáře. Z uživatelského hlediska by však použití pro rodiče nezjednodušovala, neboť podobně jako v původním rezervačním systému DK, nutí uživatele opětovně, při každé nové žádosti o rezervaci, zadávat informace o sobě

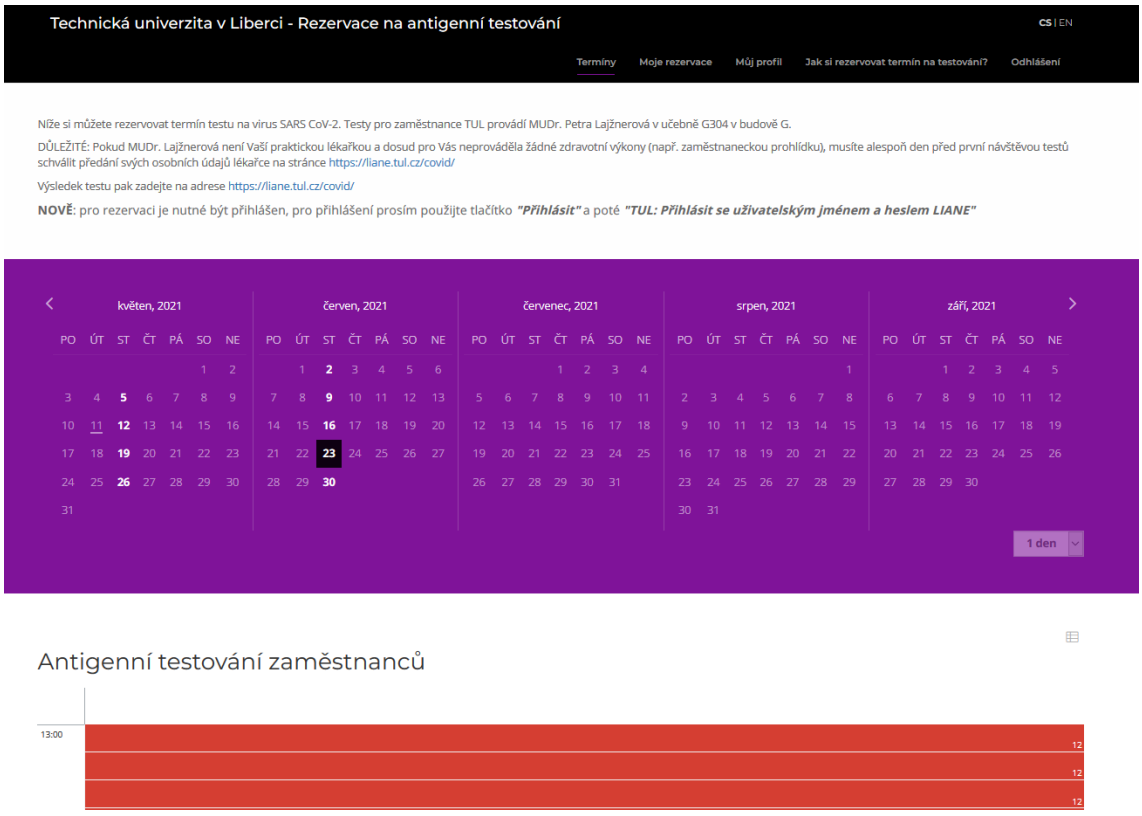
[1] Dostupné na https://easyappointments.org/

nebo dítěti. Pro asistenty chybí možnost naplánování vlastního kalendáře volných časů, stejně jako správci nenabízí protnutí časů asistentů s časem žádosti o hlídání nebo export do CSV.[6]

Oproti novému řešení Dětského koutku zajišťuje aplikace Easy!Appointments více doplňkových funkcí, které jsou pro využití Dětským koutkem zbytné. Mezi tyto funkce patří například uvádění přesnějších detailů bydlišť, rozdělení rezervací na různé služby v kategoriích nebo export do Google Kalendáře.[7]

2.2 Rezervační systém pro antigenní testování zaměstnanců TUL

Důležitým rezervačním systémem v posledních měsících vzniku této práce, kterého TUL využívá, je rezervační systém pro antigenní testování zaměstnanců TUL. Rezervační systém pracuje na podobném principu, jako výše zmíněný rezervační systém. Dovoluje uživatelům zarezervovat jeden interval z předem nastaveného seznamu termínů, k provedení antigenního testu. Systém využívá komerčního řešení *Reenio*[2], který je přizpůsobený pro využití k rezervacím na lékařské testy.



Obrázek 2.1: Výběr termínu rezervace z kalendáře volných časů

[2]Dostupné na https://reenio.cz/

Systém vyžaduje pro vytvoření rezervace přihlášení přes SSO nástroj *Shibboleth* sítě LIANE. Dochází tak k ověření uživatele, zdali je členem akademické obce TUL.[8] Toto omezení předchází vytváření rezervací cizími osobami a zároveň dovoluje aplikaci získat uživatelem schválené informace pro další vytvoření rezervace. Mezi tyto informace patří například jméno, příjmení a emailová adresa.

Po přihlášení má zaměstnanec TUL k dispozici kalendář s volnými termíny a seznam časových intervalů s přehledem obsazenosti. Po výběru volného termínu je uživateli zobrazen potvrzovací formulář s časem, informací o testovacím místě a jeho pozicí. Potvrzovací formulář zároveň dovoluje uživateli přidat poznámku. Po vytvoření rezervace času je v menu „Moje rezervace“ k dispozici seznam nadcházejících rezervací a historie rezervací. Systém dále dovoluje uživateli si prohlédnout své kontaktní údaje a informace.[9]

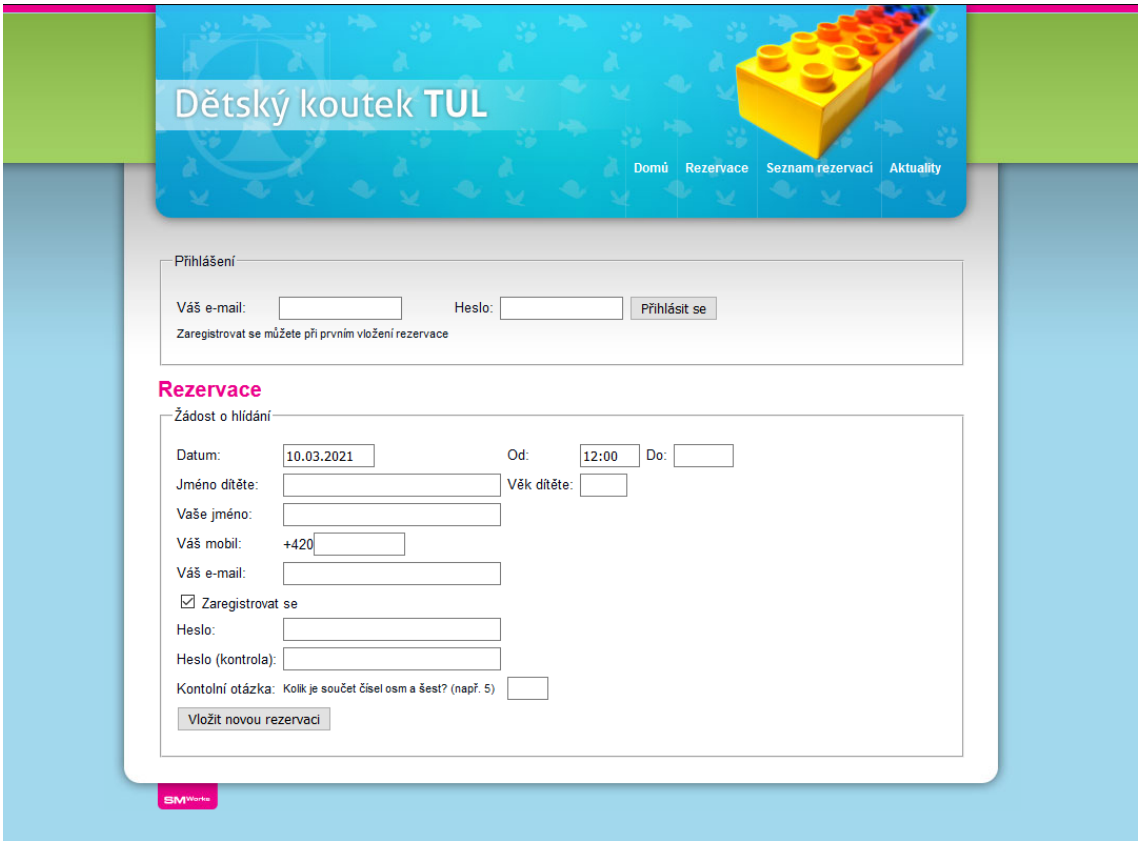
Oproti rezervačnímu systému pro DK si tento rezervační systém nemusí evidovat ke každému uživateli seznam dětí a nemusí řešit další role uživatele. Vzhledem k povaze aplikace není zapotřebí, aby aplikace řešila průniky dvou různých časových akcí, neboť uživatel vybírá z již předdefinovaného seznamu volných termínů. Nedochází zde tedy k žádnému přidělování asistentů k žádostem o termín. Systém by se tak dalo připodobnit k „plnění fronty“.

2.3 Aktuální řešení Dětského koutku

Původní webová aplikace rezervačního systému DK (viz 2.2) je vytvořená za použití programovacího jazyka PHP. Jde o webovou aplikaci s vlastním uživatelským prostředím bez použitého frameworku jako např. Bootstrap. Zdrojový kód webové stránky prozrazuje, že webová stránka je psaná s využitím staršího značkovacího jazyka XHTML 1.0, místo dnešního standardu HTML5 [10], což poukazuje na stáří webové aplikace. Vzhledem ke stáří webu není aplikace přizpůsobena pro pohodlné zobrazení na mobilních zařízeních.

Původní rezervační systém využívá pro přihlášení rodičů a administrátorů vlastního registračního a přihlašovacího formuláře. Aplikace si tak musí v backendu sama spravovat vlastní databázi uživatelských jmen a hesel. Aplikace žádným způsobem neověřuje, zdali je dotyčná osoba studentem nebo zaměstnancem TUL. Toto ověřování tak spadá na samotné vedení DK. Aplikace zároveň oproti novému řešení nenabízí roli asistent. Asistenti, kteří mají na starosti hlídání dětí, tak nemají přehled o chystaných hlídáních a přidělených dětech nebo nemají možnost si naplánovat časy, kdy budou v dětském koutku dostupní. Toto funkční omezení s sebou nese nemožnost kontroly průniku těchto časů.

Proces rezervace do DK v původním systému probíhá následovně: Uživatel po příchodu na webovou stránku vyplní ve formuláři „Žádost o hlídání“ datum hlídání, požadovaný začátek a čas hlídání, jméno a věk dítěte, jméno, email, telefonní kontakt rodiče a kontrolní otázku. Pokud se chce rodič v systému zaregistrovat, vyplní a potvrdí heslo. V případě, že chce opětovně rezervovat dítě na další termín, je nutné všechny tyto údaje znovu vyplnit. [11]



Obrázek 2.2: Formulář žádosti o rezervaci původní aplikace Dětského koutku TUL

Po přihlášení uživatele do systému zadaným emailem a heslem uvedeným v registraci, získá uživatel přístup k seznamu nadcházejících rezervací a má k dispozici předvyplněný formulář pro vytvoření nové rezervace dítěte.

3 Návrh webové aplikace

Následující kapitola je věnována samotnému návrhu webové aplikace rezervačního systému, konzultovaným požadavkům na nové nahrazující řešení a požadavky na systém, které vyplývají z předchozí rešerše. Konzultované požadavky vychází ze schůzek s konzultantkou práce a organizátorkou DK TUL Mgr. Ilonou Sovovou.

3.1 Požadavky na systém

Požadavkem vedení Dětského koutku bylo vytvořit zcela novou aplikaci, která by více odpovídala procesům přidělování hlídání asistentům.

Jak bylo výše uvedeno (viz 2.3), aktuální systém nijak nezohledňuje roli chůvy (asistenta) a veškeré zjišťování volných časů asistentů muselo probíhat interní komunikací s asistenty. Stejně tak aktuální systém žádným způsobem neověřuje, zdali je zadavatel hlídání studentem nebo zaměstnancem TUL. Dále ze schůzek vyplynulo, že aktuální procesy výpočtů pohledávek za provedená hlídání nejsou pro osoby zabývající se účetnictvím optimální.

Žádostí vedení zároveň bylo, aby k databázovému serveru byl snadný přístup pro případ řešení konfliktů dat nebo oprav údajů. Výsledkem konzultace byla shoda na využití SŘBD MySQL, respektive odnože MariaDB, která je s MySQL kompatibilní.[12]

- Aplikace by měla definovat tři odlišné role uživatelů, kteří s rezervačním systémem budou pracovat. Role *Rodič* / *Chůva* / *Správce*. Jde o role vycházející z praxe.
- Asistent by měl mít možnost označit v kalendáři své volné časy, podle kterých by mělo být možné dítě k asistentovi přidělit.
- Systém by měl dovolit skladovat interní poznámku správců o hlídaném dítěti. Uložená poznámka by měla být skryta před rodiči, jelikož může jít čistě o organizační záležitost.
- Evidence jednotlivých rezervací. Každá rezervace musí obsahovat datum, čas a konec hlídání.
- Zjištění věku dítěte a nouzového kontaktu od rodiče.

- Zajištění přihlašování uživatele. Služeb Dětského koutku mohou využívat pouze zaměstnanci nebo studenti TUL. Přihlašování k aplikaci bude řešeno pomocí SSO nástroje Shibboleth.
- Možnost jednoduchého přidělení rolí *Chůva* / *Asistent*, aniž by bylo nutné kontaktovat správce serveru nebo databáze.
- Umožnit přehledný výpis pohledávek za provedená hlídání a umožnit export dat pro účetní.

Další požadavky na aplikace vychází z konzultací s vedoucím práce Ing. Janem Kolajou, Ph.D. a z rešerše předchozího řešení rezervačního systému Dětského koutku.

Původní aplikace nebyla uživatelsky přívětivá. Pravidelně nutila uživatele zadávat při opakovaných hlídáních stejné údaje o dítěti a rodiči. Vyplňování věku dítěte jako čísla aktuálního věku, byl pro nový systém zcela nevhodný, jelikož při uložených informacích o dítěti může systém věk vypočítat z data narození.

- Aplikace musí umožnit rodičům uložit si své děti, pro které bude následně možné vytvářet rezervace.
- Věk dítěte bude zadáván datem narození.
- Rodič by měl mít možnost přidat ke svým dětem poznámku, která se následně bude zobrazovat správcům a asistentům.
- Rodič by měl mít přehled o provedených hlídáních. Vzhledem k nastavené ceně je možné zobrazit rodiči celkovou sumu za provedená hlídání.

3.2 Použité technologie

Pro realizaci nového rezervačního systému pro DK je využito několik webových a serverových technologií. V následujících podkapitolách jsou tyto použité technologie popsány a je zdůvodněno jejich použití.

3.2.1 Client-side technologie

Následující technologie jsou využívány na straně uživatele při vykreslování jednotlivých webových stránek a formulářů.

HTML, CSS

HTML a CSS jsou základními technologiemi potřebnými k vytvoření statické webové stránky v prohlížeči. HTML je zkratkou pro „HyperText Markup Language“, v překladu „Hypertextový značkovací jazyk“. Jak název vypovídá, nejde o programovací jazyk, ale o jazyk, který popisuje význam textu, podle kterého prohlížeč přiděluje příslušné vizuální styly dle CSS.[13] Lze si tak představit, že textu přidáváme hlubší význam a druhý rozměr, ve kterém specifikujeme další parametry.

CSS je zkratkou pro „Cascading Style Sheets“ nebo také „Kaskádové styly“. Jde o kolekci metod pro grafickou úpravu webových stránek. Nastavením stylů odpovídajícím třídám HTML značek, vytváříme vzhled finální webové stránky v prohlížeči uživatele.[14]

Velké množství CSS stylů je do stránky implementováno díky využití frameworku Bootstrap. Aplikace využívá však i vlastních dodatečných stylů, které slouží pro zlepšení vzhledu několika částí webu, jako například úvodní stránka, barvy tlačítek nebo zápatí stránky.

Bootstrap

Bootstrap je framework (softwarová struktura) pro vývoj responzivních webových stránek a aplikací, s použitím HTML, CSS a JavaScriptových pluginů.[15] Aplikace Dětského koutku využívá tento framework ve verzi 4.5.2. Díky využití tohoto frameworku, je celá webová aplikace snadno zobrazitelná i na malých zařízeních, jako jsou mobilní telefony.

Uživatelské prostředí ve formulářích využívá pro zadávání kalendářního data doplňkovou knihovnu *bootstrap-datepicker* ve verzi 1.9.0, která po kliknutí na zadávací pole vykreslí miniaturní kalendář se zaklikávacím datem.[16] Dalším rozšířením je bezplatná verze knihovny *FullCalendar* využitá pro vykreslení velkého kalendáře a manipulaci s akcemi v daném kalendáři.[17] Obě rozšíření využívají prostředí Bootstrap a skriptů v jazyce JavaScript.

JavaScript, jQuery

JavaScript je interpretovaný programovací jazyk využívaný webovými stránkami k jejich skriptování.[18] JavaScript je v aplikaci Dětského koutku využívaný k dynamické manipulaci s jinak statickými webovými stránkami. Díky JavaScriptu je možné generovat na stránce nová okna nebo HTML elementy a pomocí technologie AJAX asynchronně komunikovat se serverem, tedy bez nutnosti opětovně načítat celou stránku.[19]

jQuery je knihovna JavaScriptu určená k usnadnění v orientaci mezi jednotlivými HTML elementy.[20] Jde o knihovnu, kterou pro svůj běh vyžaduje aplikací využívaný framework Bootstrap. Knihovna jQuery je využívaná ve verzi 3.5.1.

I přes poměrně velkou popularitu knihovny jQuery, vývojáři od jejího používání ustupují. Jako důvod uvádí, že knihovna není v roce 2021 potřebná, vzhledem k tomu, že velké množství funkcí, které knihovna poskytuje, lze naprogramovat nativně použitím standardizovaných příkazů prohlížeče.[21] Důvodem, proč je knihovna jQuery v této práci použita, je závislost frameworku Bootstrap, který by v použité verzi nemohl správně fungovat. Nastupující verze frameworku Bootstrap od použití technologie jQuery ustupuje a není již nutné knihovnu přikládat.[15]

3.2.2 Server-side technologie

Dále popsané technologie jsou využívány na serverové straně. Jde o programovací nástroje nebo knihovny, které jsou pro funkcionalitu projektu nezbytné.

Node.js

Node.js je běhové prostředí JavaScriptového kódu, vhodné například pro tvorbu webových serverů. Node.js pro běh webových serverů využívá modelu událostí (eventů) a asynchronního programování.[22]

Výhodou využití programovacího jazyka JavaScript v prostředí Node.js, oproti konvenčnějšímu programovacímu jazyku PHP, je možnost snadné přenositelnosti mezi servery. K vytvoření webové aplikace nepotřebuje programátor nebo správce aplikace zvlášť nastavovat webový server, ať už *Apache* nebo *nginx* a zvlášť PHP. V případě Node.js je nutné instalovat pouze běhové prostředí Node.js a celou aplikaci nainstalovat pomocí přiloženého balíčkovacího systému `npm`, který automaticky připraví knihovny potřebné pro běh webového serveru. Ten je následně spuštěn na přednastaveném portu. Správce dále nemusí řešit cesty a oprávnění k jednotlivým souborům webových stránek. Výhodou pro programátora je možnost pokračování ve zpracování kódu i po odeslání vygenerované webové stránky uživateli aplikace. Této možnosti je využito například při potvrzení rezervace, kdy po odeslání odpovědi uživateli, dochází k přípravě a odeslání emailu s potvrzením.

Možnou nevýhodou využití Node.js oproti PHP je nutnost přístupu k serveru, na kterém bude aplikace spuštěna. Webovou aplikaci tak není obvykle možné provozovat na komerčních sdíleních hostingových službách, které využívají společného webového serveru jako *Apache* nebo *nginx*, pro běh více různých webových aplikací svých zákazníků.

Express

Express je framework pro Node.js určený k tvorbě webových serverů a aplikací. Umožňuje tvorbu webové aplikace podle návrhu MVC (Model-View-Controller), dovoluje využití templatovacího enginu (viz 3.2.2) ke generování výsledných HTML stránek a umožňuje při zpracování používání dalších prostředníků (tzv. middleware).[23]

Aplikace Dětského koutku využívá tohoto frameworku ve verzi 4.17.1. Společně s tímto frameworkem je použitý middleware *express-session* ve verzi 1.17.1, který slouží pro udržování informací o sezení uživatele. Webový server díky tomu může udržovat informace o tom, který webový prohlížeč je přihlášený a bude mít přístup k funkcím aplikace.[24] Dalším middlewarem je *Helmet* ve verzi 4.6.0, který zajišťuje základní nastavení bezpečnostních hlaviček pro prohlížeč.[25]

EJS

Templatovací engine EJS, neboli „Embedded JavaScript Templating“ je nástroj pro generování HTML značek za použití čistého JavaScriptového kódu z předem připravených šablon.[26] Funguje na podobném principu jako PHP, kdy do připravené HTML stránky, je možné doplnit generující kód (viz Listing 3.1). Soubory určené pro EJS engine jsou označeny příponou `.ejs`, místo klasického `.html`.

```
<%- include ("templates/header") %>
<%
if (typeof currentInfo != 'undefined') {
    %>
    <div class="col">
        <h2>Aktuální informace</h2>
        <%- currentInfo %>
    </div>
    <%
}
%>
```

Listing 3.1: Ukázka šablony pro EJS dosazující header stránky a aktuálních informací o koutku, pokud jsou k dispozici

Shibboleth

Shibboleth je systém pro SSO (Single Sign-On). Jde o technologii, jejíž hlavním účelem je prokázání identity uživatele poskytovatelem identity a předání požadovaných a schválených údajů aplikaci. Uživatel provádí přihlašování přímo na stránce spravované univerzitou, na doméně `shibbo.tul.cz`. Shibboleth tak řeší soukromí a univerzita dovoluje uvolnění jen nezbytných informací.[27]

Pro aplikaci Dětského koutku je využití Shibbolethu pro autentizaci uživatele výrazným usnadněním, jelikož samotná aplikace nemusí řešit skladování hesel, platnost osobních údajů nebo platnost emailové adresy. Zároveň aplikace nemusí ověřovat, zdali je rodič ve vztahu s TUL. Všechny tyto informace jsou aplikaci předávané poskytovatelem identity v hlavičkách požadavků na zobrazení stránky. Výhodou pro uživatele je fakt, že nemusí skladovat další uživatelské jméno a heslo.

MariaDB

Pro systém řízení báze dat byl zvolen relační databázový server MariaDB. MariaDB je open source řešení, které původně vzniklo jako odnož systému MySQL.[28] Jak bylo uvedeno v zadání vycházejícího z konzultace, systém je při použití databázového enginu *InnoDB* kompatibilní s databázovým systémem MySQL, a stejně jako MySQL, je spravovatelný pomocí standardních aplikací jako *phpMyAdmin* nebo *Adminer*.[12][29]

Důvodem volby MySQL je výhoda relačních databází, neboť v databázovém modelu jsou tabulky propojeny na základě primárních a vedlejších klíčů. Lze tak například snadno vypsat děti patřící rodiči nebo rezervace, které byly vytvořeny jedním rodičem.

Git

Git je nástroj pro správu verzí kódu. Zaznamenává změny souborů tak, aby bylo možné se vrátit k předchozím změnám v kódu. Dovoluje sledovat kdo jaké změny provádí, dovoluje navrhovat změny a při použití vzdáleného repozitáře, zjednodušuje správu kódu na více zařízeních.[30]

Využití Gitu umožnilo uložení zdrojového kódu ve vzdáleném repozitáři *GitHub.com*, a díky tomu bylo možné pracovat na jednom kódu z více zařízení. GitHub zároveň dovoluje integraci dalších rozšíření. Mezi takové rozšíření se řadí i *Dependabot*. Jde o automatizovaný nástroj, který na základě použitých knihoven a balíčků pravidelně kontroluje dostupnost aktualizace. Jakmile je aktualizace používaného balíčku k dispozici, připraví v Gitu tzv. Pull Request, neboli žádost o přidání změny do hlavního kódu, kterou správce repozitáře může přijmout, pokud nedochází k problémům s kompatibilitou.[31] Během vývoje přišel Dependabot s několika aktualizacemi, které přinášely především bezpečnostní aktualizace balíčků.

3.3 Databázový model

Pro SŘBD byla vybrána MariaDB databáze, jednotlivé tabulky databáze využívají InnoDB engine. V programu *MySQL Workbench* společnosti Oracle, byl vytvořen následující návrh databáze (viz obr. 3.1).

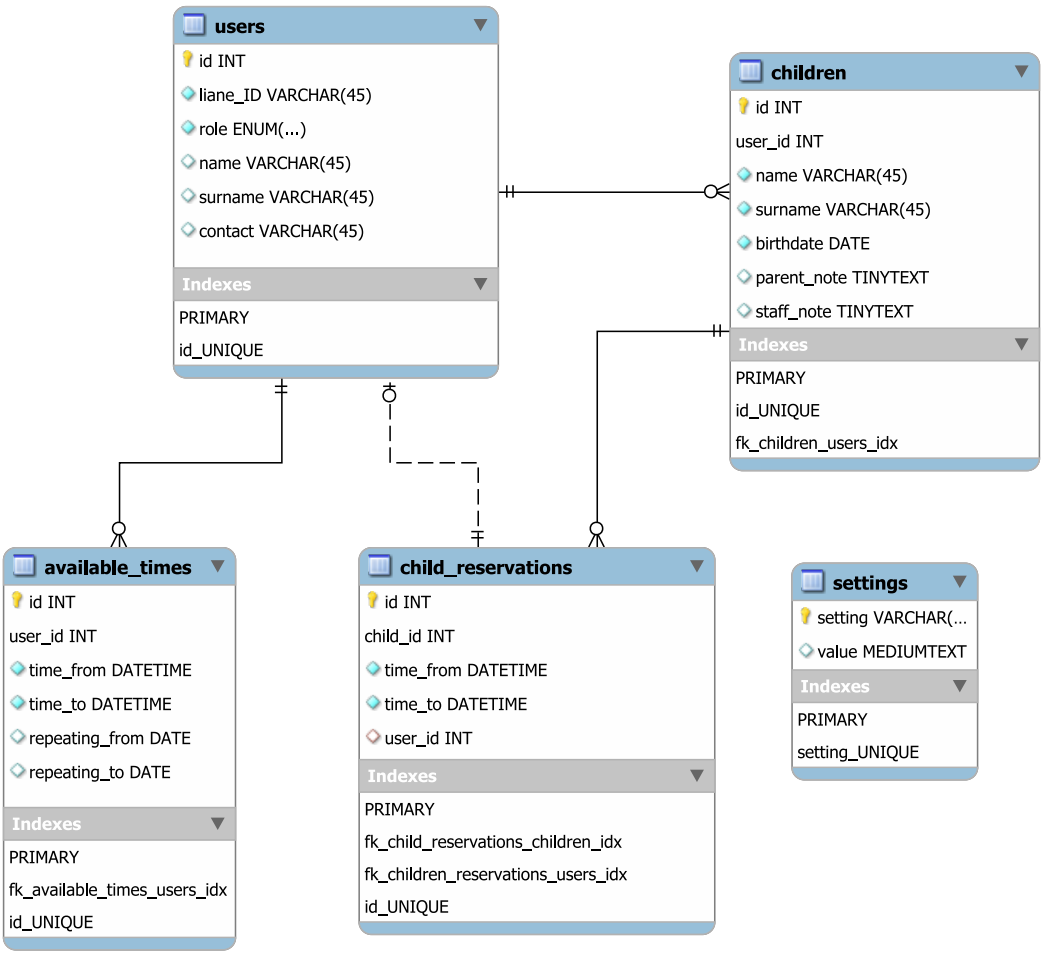
Databáze s názvem `detskykoutek` obsahuje pět tabulek. Databáze splňuje 1. normální formu (1NF), tedy všechny atributy jsou atomické, 2NF, kdy předcházíme vytváření duplicit v záznamech a 3NF, ve které tabulka neobsahuje tranzitivní závislosti.[32]

users

Tabulku `users` lze považovat jako hlavní tabulku celé databáze, neboť ostatní entity využívají sloupec `id` této tabulky jako cizí klíč. Tabulka je silnou entitou.

- Sloupec `id` je primárním klíčem, tedy je nenulový a unikátní. Sloupec má nastavený atribut *auto_increment* pro automatické navyšování hodnoty u nových řádků.
- Sloupec `liane_ID` je hodnotou získanou z Shibbolethu, z hodnoty `eppn`. Jedná se o emailovou adresu uživatele. Hodnota musí být nenulová.
- Sloupec `role` je rolí uživatele. Typ `ENUM` umožňuje zadávat pouz nastavené povolené hodnoty. Těmi jsou pouze `parent`, `assistant` a `admin`. Hodnota musí být nenulová.
- Sloupce `name` a `surname` jsou jménem a příjmením uživatele. Jde o hodnoty získané z Shibbolethu. Hodnoty mohou být nulové.
- Sloupec `contact` je kontaktem na rodiče. Jde o hodnotu zadávanou uživatelem. I přes kontrolu validity telefonního čísla regulárním výrazem, je použit `VARCHAR` místo `INT`, neboť v kontaktu mohou být i povolené znaky, jako znak `+`.

Důvod, proč mohou být hodnoty `name`, `surname` a `contact` nulové, je ten, že administrátor může předem přidat uživatele podle emailové adresy jako asistenta. Po prvním přihlášení bude tak mít asistent ihned k dispozici položky pro asistenta.



Obrázek 3.1: ER diagram databázového modelu

children

Tabulka `children` je slabou entitou. Každý uživatel může mít více dětí, ale každé dítě musí být přiřazeno právě jednomu uživateli.

- Sloupce `id` a `users` jsou primární klíče, přičemž sloupec `id` má nastavený atribut `auto_increment`. Sloupec `user_id` je cizím klíčem z tabulky `users`.
- Sloupce `name` a `surname` jsou jménem a příjmením dítěte. Hodnoty nesmí být nulové.
- Sloupec `birthdate` je datem narození dítěte. Hodnota nesmí být nulová.
- Sloupce `parent_note` a `staff_note` jsou poznámkami rodičů a vedení pro asistenty. Hodnota může být nulová.

child_reservations

Tabulka `child_reservations` je slabou entitou. Každé dítě může být přiřazeno ve více rezervacích, ale každá rezervace musí být pro právě jedno dítě. V případě, že je v entitě hodnota sloupce `user_id` nulová, je rezervace brána jako zatím nepotvrzená. Hlídajícím asistentem může být právě jeden uživatel.

- Sloupce `id` a `child_id` jsou primární klíče, přičemž sloupec `id` má nastavený atribut `auto_increment`. Sloupec `child_id` je cizím klíčem z tabulky `children`.
- Sloupce `time_from` a `time_to` jsou datem s časem začátku a konce rezervace hlídání. Hodnota nesmí být nulová.
- Sloupec `user_id` je asistentem, který bude v daný čas, dané dítě hlídat. Jde o neidentifikační vazbu s tabulkou `users`. Hodnota může být nulová.

available_times

Tabulka `available_times` je slabou entitou. Každý dostupný čas musí mít právě jednoho uživatele, ale každý uživatel může mít více časů, kdy je dostupný v koutku. V případě, že jsou v entitě hodnoty sloupce `repeating_from` a `repeating_to` nulové, je termín volného času brán jako jednorázový.

- Sloupce `id` a `user_id` jsou primární klíče, přičemž sloupec `id` má nastavený atribut `auto_increment`. Sloupec `user_id` je cizím klíčem z tabulky `users`.
- Sloupce `time_from` a `time_to` jsou datem s časem začátku a konce volného času, kdy asistent může být k dispozici v DK. Hodnoty nesmí být nulové.
- Sloupce `repeating_from` a `repeating_to` jsou datem začátku a konce opakování termínu. Hodnoty mohou být nulové.

settings

Tabulka `settings` je silnou entitou. Nemá žádný vztah se zbylými entitami. Jde o tabulku s údaji pro různé funkce rezervačního systému. Obsahuje například položku `price`, která je používaná k vypočtení celkové ceny za hlídání.

- Sloupec `setting` je primárním klíčem, proto musí být unikátní. Obsahuje název položky nastavení.
- Sloupec `value` je hodnotou daného nastavení. Hodnota může být nulová.

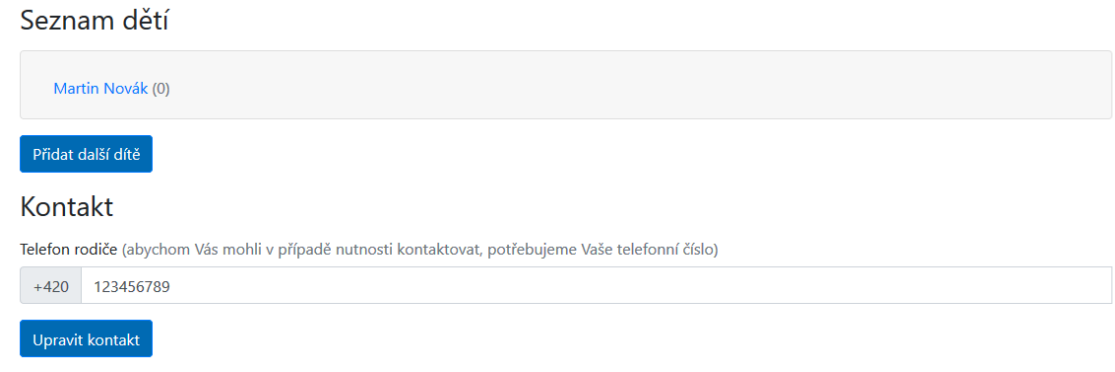
3.4 Role uživatele

Jednou z hlavních úloh Rezervačního systému je rozdělení uživatelů na tři role. *Rodič*, *Asistent* a *Správce*. Každá z těchto rolí má v systému k dispozici specifické funkce navíc, společně s funkcemi které dědí z nižší role, neboť i asistent/ka Dětského koutku může být rodičem, stejně tak jako jím může být správce DK.

3.4.1 Rodič

Role Rodič je výchozí rolí každého přihlášeného uživatele Dětského koutku. Přihlášený rodič může přidat do databáze své děti, pro které následně vytváří žádosti o rezervaci. Pro vytvoření rezervace je nutné, aby rodič měl v systému zadané své telefonní číslo, na kterém mohou asistenti v případě nouze rodiče kontaktovat.

Po prvním přihlášení do systému přes Shibboleth je rodič proveden několika málo formuláři. V případě, že rodič nemá v systému přidané dítě, je rodiči zobrazen formulář pro přidání dítěte. V tomto formuláři se zadávají o dítěti základní údaje, mezi které patří jméno, příjmení a datum narození, což je první výrazná změna oproti předchozímu systému, kdy rodič zadával věk přímo číslem. Rodič tak poté nemusí opětovně zadávat věk dítěte při každé rezervaci, neboť věk je dopočítáván automaticky. Pro asistenty a vedení je možné zároveň ke každému dítěti přidat zvlášť poznámku, která může vedení informovat o speciální vyžadované péči nebo zdravotních omezeních. Pokud rodič nemá vyplněný telefonní kontakt, je navíc zobrazen formulář pro zadání telefonního čísla. Bez této informace není možné pokračovat dále k vytváření rezervace.



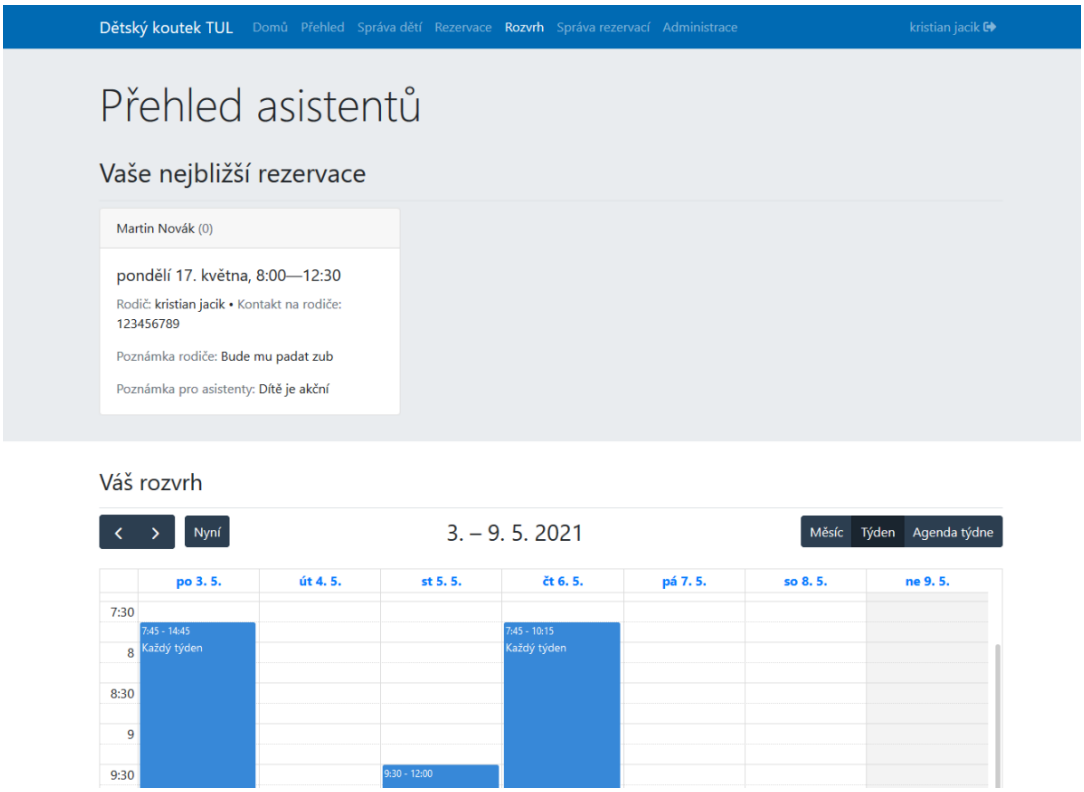
Obrázek 3.2: Rozbalovací seznam dětí a úprava telefonního kontaktu

Po přidání dětí je možné otevřít stránku s formulářem pro vytvoření rezervace. V tomto formuláři se rodiči zobrazuje výběrové menu s přidanými dětmi, pole s výběrem dne hlídání a dvě pole s časem začátku a konce hlídání. Po vytvoření rezervace se rodiči zobrazí seznam se zatím nepotvrzenými a potvrzenými rezervacemi (viz obr. 4.6).

Pro rodiče je dále k dispozici stránka s přehledem o provedených hlídáních v aktuálním měsíci. Zde má možnost nahlédnout i na předchozí měsíce, kde uvidí časy za hlídání a vypočtenou částku. Na stejné stránce má k dispozici i seznam naplánovaných nadcházejících rezervací, stejně jako na stránce s vytvářením rezervace. Důvodem je přehlednost systému a fakt, že jde o stránku, kde má mít rodič největší přehled o hlídáních.

3.4.2 Asistent

Asistenti mají v rezervačním systému vlastní roli. Stejně jako rodič, mohou přidat seznam vlastních dětí, přidat svůj kontakt a vytvořit žádosti o hlídání. Navíc mají k dispozici vlastní stránku, kde uvidí jim přidělené naplánované rezervace. Na rozdíl od rodiče, mají asistenti v seznamu nadcházejících rezervací k dispozici navíc poznámku vedení ke každému dítěti, ve kterém může být upozorněn na nějakou dodatečnou informaci, která se ke hlídání dítěte může hodit.



Obrázek 3.3: Přehled nadcházejících hlídání pro asistenty s rozvrhem časů dostupnosti v DK

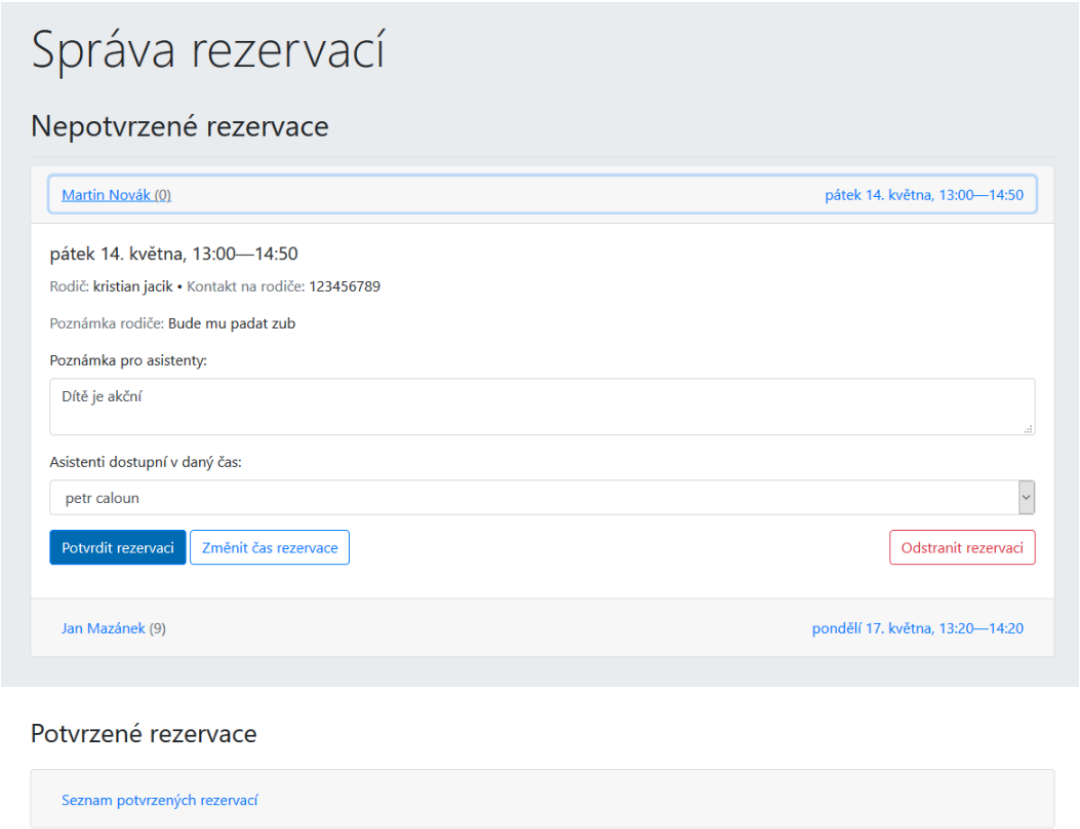
Hlavní funkcí pro asistenty je funkce rozvrhu, ve kterém si mohou naplánovat osobní rozvrh, kdy budou k dispozici pro hlídání v dětském koutku. V rozvrhu si asistenti kliknutím a tažením přidávají jednorázové časy, které vyvolají *modal* (okno v rámci stránky), kde mohou nastavit od jakého do jakého data se bude daný termín opakovat. Pokud asistenti budou dopředu vědět, že budou mít každé úterý během semestru volno na hlídání v koutku, nastaví opakování daného termínu od 1. října do 31. února. Tyto nastavené časy má potom správce k dispozici k nahlédnutí a systém z nich dále zjišťuje, zdali je na rodičem požadovaný termín, k dispozici nějaký asistent, případně pokud je k dispozici asistent, který hlídá ve stejný den.

V kalendáři má asistent možnost s jednotlivými rezervacemi manipulovat, posouvat je nebo prodlužovat a zkracovat jednoduchým tažením myši. V případě nechtěné změny je zobrazena dočasná toast notifikace v pravém dolním rohu s tlačítkem, které dovoluje změnu vrátit zpět (viz obr. 4.4).

3.4.3 Správce

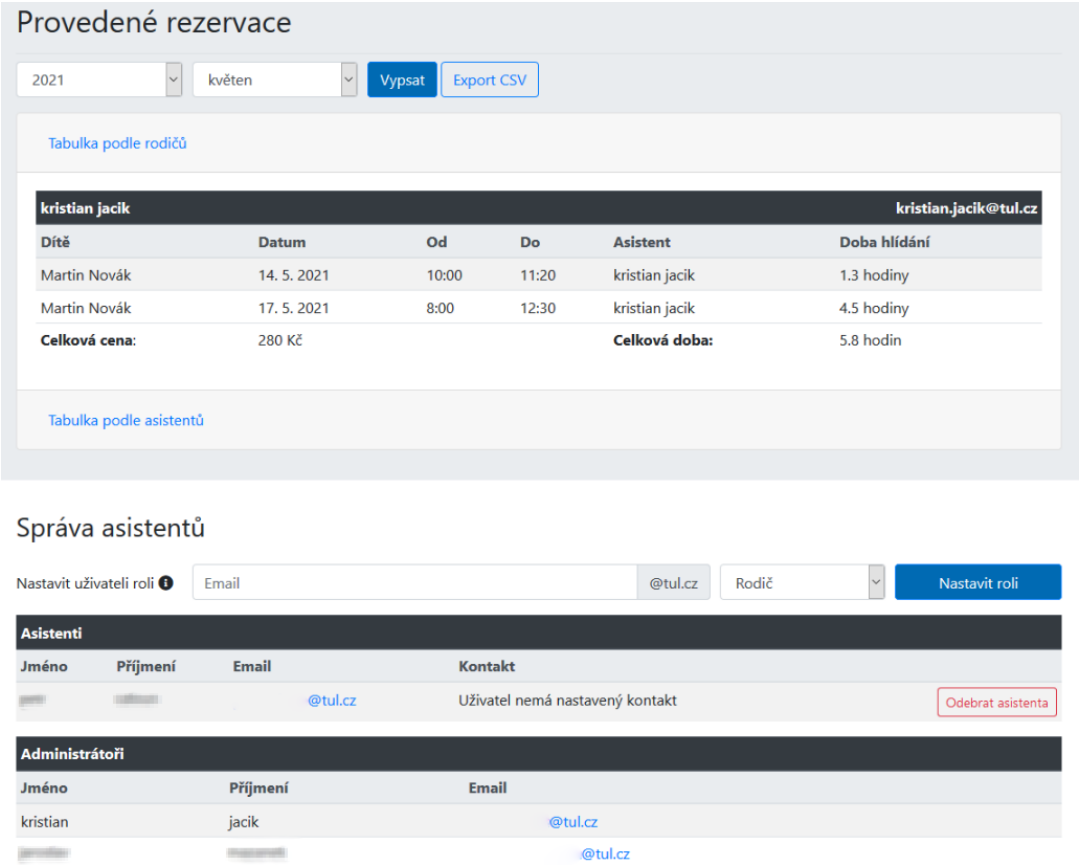
Pro správce Dětského koutku je určena role Správce. Správce má v systému nejvyšší oprávnění. Pro správce jsou k dispozici navíc dvě stránky.

První stránka „Správa rezervací“ je určená pro správu denního chodu koutku. Má k dispozici menu se seznamem nepotvrzených rezervací, ke kterým může přiřadit asistenta a přidat k dětem pro asistenty poznámku. Na stejné stránce má správce k dispozici nadcházející potvrzené rezervace, kterým může změnit asistenta v případě nutné změny, nebo zrušit potvrzení rezervace odebráním asistenta. U obou druhů rezervací, jak potvrzených, tak nepotvrzených, má správce po kliknutí na tlačítko „Změnit čas rezervace“, možnost upravit čas začátku a konce rezervace, případně ji přesunout na jiný den, což může být vhodné po žádosti rodiče nebo po předchozí komunikaci. Správce má zároveň možnost rezervaci kompletně zrušit. V takovém případě je zobrazeno potvrzovací menu, neboť taková změna je nevratná.



Obrázek 3.4: Stránka pro správce s potvrzováním rezervací a rozvrhy asistentů

Druhá stránka „Administrace“ je věnována administrativní správě Dětského koutku. Součástí této stránky jsou výpisy provedených hlídání za vybraný měsíc, které jsou rozdělené podle rodičů a podle asistentů. V tabulkách jsou vypsána provedená hlídání, jejich časy a jednotlivé ceny za započaté hodiny. V tabulce je uvedena i celková cena za všechna hlídání v daném měsíci. Součástí je i možnost exportu informací o měsíci do CSV pro následné zpracování např. v Excelu.



Obrázek 3.5: Administrace s výpisem provedených hlídání a seznamem asistentů a administrátorů

Součástí této stránky je správa uživatelů, kde je možné podle emailové adresy, nastavit uživatele jako správce nebo asistenta, případně jejich role odebírat. U asistentů jsou vypsány jejich kontakty a tlačítko k odebrání asistenta. Kvůli bezpečnosti není možné, aby správce odebral sama sebe jako správce, ale může odebrat jiného správce. Systém tak předchází situaci, kdy by nebyl v systému žádný správce a bylo by nutné ručně zasahovat do databáze.

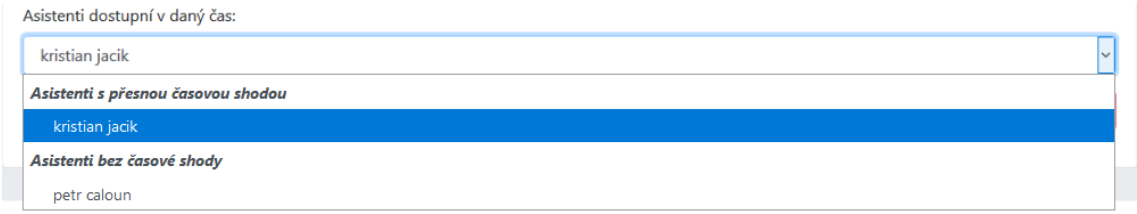
V poslední části je správci umožněno nastavit aktuální informace o dětském koutku pro rodiče nebo velké varování, například v případě omezení fungování DK. Obě tyto hodnoty se zobrazují na úvodní stránce.

3.5 Funkce aplikace

Aplikace zajišťuje několik dalších funkcí, které jsou oproti původnímu systému novinkou a nebyly zatím popsány v popisu rolí výše. Cílem těchto funkcí je usnadnění práce správcům DK, při přidělování rezervací k asistentům, nebo při výpočtů výplat a pohledávek za uplynulé měsíce.

3.5.1 Potvrzení rezervace a přidělení rezervace asistentovi

Po vytvoření požadavku na hlídání dítěte rodičem a po nastavení volných časů v kalendáři asistenty, dochází při vypisování čekajících nepotvrzených žádostí na straně správce k předvypočítání průniků těchto časů.



Obrázek 3.6: Výpis časových průniků asistentů s časem rezervace

Aplikace porovnává požadovaný čas a kontroluje, zdali zadaný interval se buď přesně shoduje s některým ze zadaných volných časů asistenta, nebo zdali se shoduje alespoň den, ve kterém bude asistent pro DK dostupný. V případě, že asistent má v systému zadaný čas, který se opakuje, kontroluje se daný čas od začátku do konce intervalu opakování. V případě, kdy asistent nemá přesný časový průnik, ale bude dostupný daný den, mají správci o situaci větší přehled a mohou daného asistenta kontaktovat, zdali by nemohl do koutku dorazit dříve, či naopak zůstat déle.

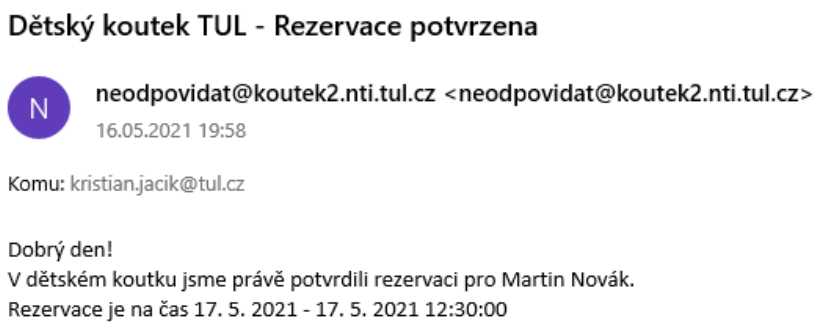
Systém dále vypíše i zbývající asistenty, kteří nemusí být v daný den vůbec dostupní, neboť mohlo dojít ke změně, kterou asistenti zatím přidáním volného času, nepotvrdili. Pro přehlednost je správci navíc zobrazován kalendář, ve kterém je pro jednotlivé asistenty, zvýrazněn jejich volný čas a dosavadní potvrzené rezervace s časy začátků a konců hlídání.

Přiřazením asistenta k dané rezervaci dochází k potvrzení platnosti rezervace. Jakmile je tedy v tabulce `child_reservations` databáze vyplněná položka `user_id`, zobrazí se rezervace pro rodiče jako potvrzená.

3.5.2 Odesílání potvrzujících emailů

Systém po potvrzení rezervace přiřazením asistenta, má za úkol informovat rodiče o platnosti dané rezervace odesláním emailové zprávy. Tato funkcionalita je zajištěna knihovnou pro Node.js *Nodemailer*[33], která vyvolává akci odeslání emailu spuštěním Bash příkazu, kde volá Linuxový program *Sendmail* využívající SMTP protokolu.[34]

Vzhledem k nastavení *Sendmail*, je emaily možné odesílat jen na emailové adresy v rámci domény `@tul.cz`. Pro využití rezervačního systému není toto omezením, vzhledem k tomu, že uživateli aplikace, mohou být pouze zaměstnanci a studenti TUL, kteří mají emailové adresy vytvořené právě pod touto doménou.



Obrázek 3.7: Ukázka potvrzujícího emailu

3.5.3 Generování CSV výpisu

Správci je umožněno vygenerovat výpis provedených hlídání za daný rok ve formátu CSV. Ten lze poté jednoduše importovat do tabulkového procesoru jako *Excel* nebo *Calc*.

Výběr provedených hlídání v daném měsíci daného roku je proveden jedním SQL příkazem. Ten vypisuje výsledek s propočítanými časy, spočtenými cenami, vypsanými jmény a kontakty v JSON formátu, který je poté převeden na CSV.

V rámci zajištění kompatibility s Excelem je nutné enkódovat daný CSV soubor s `BOM` předponou. Na začátek celého CSV výstupu je tak nutné připsat značku `\ufeff`. Výsledná tabulka po importu CSV dat může vypadat podobně, jako na obrázku 3.8.

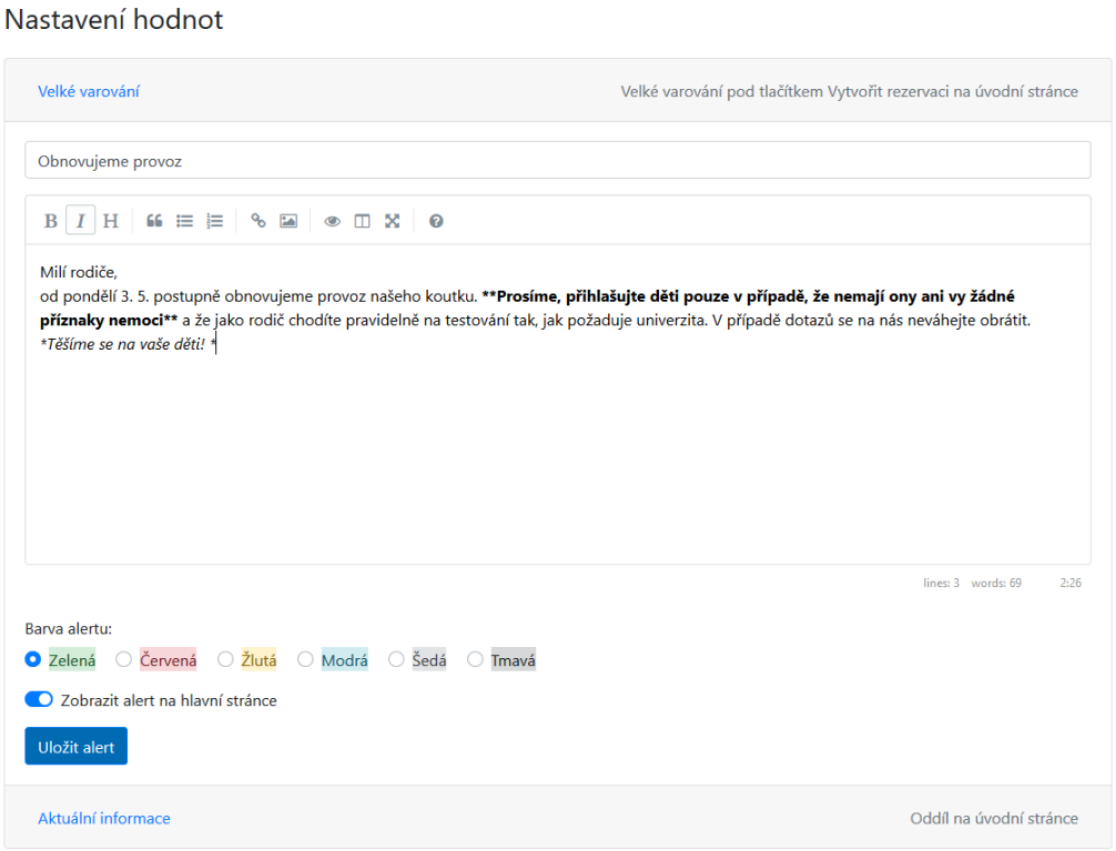
	J	K	L	M	N	O	P	Q
	Jméno asistenta	Příjmení asistenta	Email asistenta	Začátek rezervace	Konec rezervace	Odhlídané minuty	Započaté hodiny	Cena za započaté hodiny
2	Tomáš	Chválil	tomas.chvalil@tul.cz	14.05.2021 10:00	14.05.2021 11:00	60	1	40
3	Markéta	Novotná	marketa.novotna@tul.cz	07.05.2021 13:00	07.05.2021 14:20	80	2	80
4	Jana	Petrú	jana.petru@tul.cz	08.05.2021 12:00	08.05.2021 13:45	105	2	80

Obrázek 3.8: Ukázková tabulka z CSV výstupu (z důvodu šířky ukázky oříznuto jen na sloupce se smyšlenými asistenty a časy)

3.5.4 Zobrazení informačních polí na hlavní stránce

Správce DK má možnost nastavit aktuální informace o Dětském koutku, stejně tak jako přidat velký zvýrazněný box, neboli Alert.

Pro zjednodušení vytváření těchto informací je využit syntax Markdown, díky kterému lze snadno formátovat text, vytvářet seznamy, nebo přidávat odkazy, bez nutnosti znalosti HTML, pouze doplňováním předem vybraných značek. K dispozici je i WYSIWYG editor. Ten zobrazuje vizuální náhled přímo v textovém poli tak, jak bude na úvodní stránce zobrazen.[35] K zobrazení editoru se na straně klienta využívá knihovny SimpleMDE.[36]



Obrázek 3.9: WYSIWYG editor Alertu v Markdownu s možnostmi barev pozadí a přepínače zobrazování

Do tabulky `settings` je formátovaný text po vytvoření ukládán v čistém Markdown zápisu. Hlavní úvodní stránka následně tento Markdown kód převádí do HTML, díky knihovně `Showdownjs` ve verzi 1.9.1.[37] Do tabulky `settings` je zároveň ukládána informace, zdali má být na úvodní stránce Alert zobrazen, jeho nadpis a barva jeho pozadí. Červené pozadí Alertu tak může signalizovat důležité upozornění o uzavření koutku, zelené naopak o jeho opětovném otevření.

4 Implementace webové aplikace

Následující kapitola je věnována samotné implementaci celého kódu rezervačního systému DK, společně s popisem klíčových procesů a popisem struktury aplikačního kódu s několika ukázkami.

Celá aplikace využívá standardního modelu MVC (Model-view-controller), kdy řídicí logika (JavaScriptový kód v Node.js) provádí CRUD operace (Create, Read, Update, Delete) nad datovým modelem (databáze v MariaDB) a výsledky předává uživatelskému rozhraní, které je generované templatovacím enginem EJS.

Souborová struktura celé aplikace je zobrazena na obrázku v příloze 1. Soubor v adresáři `/bin/www.js` je hlavním spouštěcím souborem celého rezervačního systému. Obsahuje inicializaci knihovny Express, nastavení portů aplikace a inicializaci HTTP a HTTPS serveru. Adresář `/node_modules` je kořenovým adresářem nainstalovaných knihoven. Není součástí zabalené aplikace a je vytvořen automaticky balíčkovacím systémem `npm` při instalaci závislostí. Další adresáře jsou popsány v následujících podkapitolách.

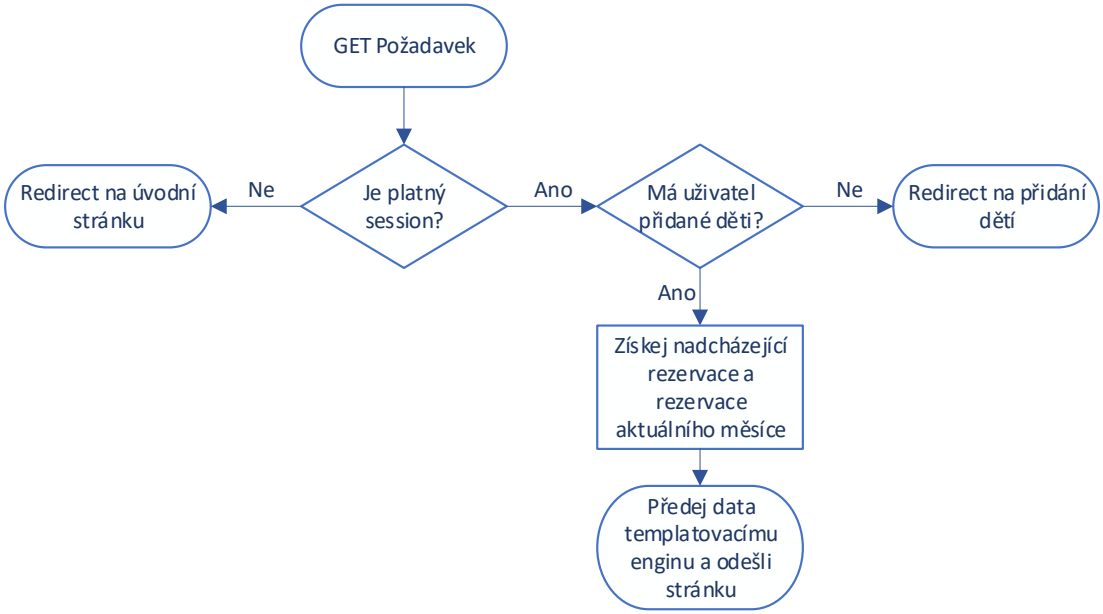
4.1 Routování požadavků

Nejdůležitější součástí kódu je routování požadavků (předávání requestů) na zobrazení webových stránek jednotlivým zpracovávajícím middlewarům (prostředníkům). O tuto logiku se stará soubor `app.js` a routovací soubory v adresáři `/routes`.

Podle URL adresy je získána cesta, která definuje, který middleware bude použit. V případě, že chceme přistoupit ke stránce pro rodiče s výpisem provedených a naplánovaných hlídání, přistupujeme k adrese `https://koutek2.nti.tul.cz/user`. Aplikace podle adresy ví, že se snažíme přistoupit k cestě `/user`, načež předává zpracovávání routeru `userRouter`, který se nachází v `/routes/user.js`.

V tomto routeru se dále kontroluje, o jaký typ požadavku jde. Aplikace rozlišuje primárně dva typy requestů `GET` a `POST`. Pokud v našem případě, jde o `GET` request, je v routeru zpracováno postupně několik procesů, které se řídí diagramem na obrázku 4.1.

Další nastavené cesty těchto routerů, zajišťují komunikaci webové stránky s daty ze serveru, čímž je vytvořená tzv. REST API. Díky AJAXu, jsou základními HTTP requesty na specifické cesty serveru, klientské webové aplikaci poskytovány odpovědi ve formátu JSON.[38] Webová aplikace tak může například změnit na stránce dynamicky obsah. Například po výběru roku, vypsat měsíce, kdy byly vytvořené rezervace, viz ukázka kódu 4.1.



Obrázek 4.1: ER diagram databázového modelu

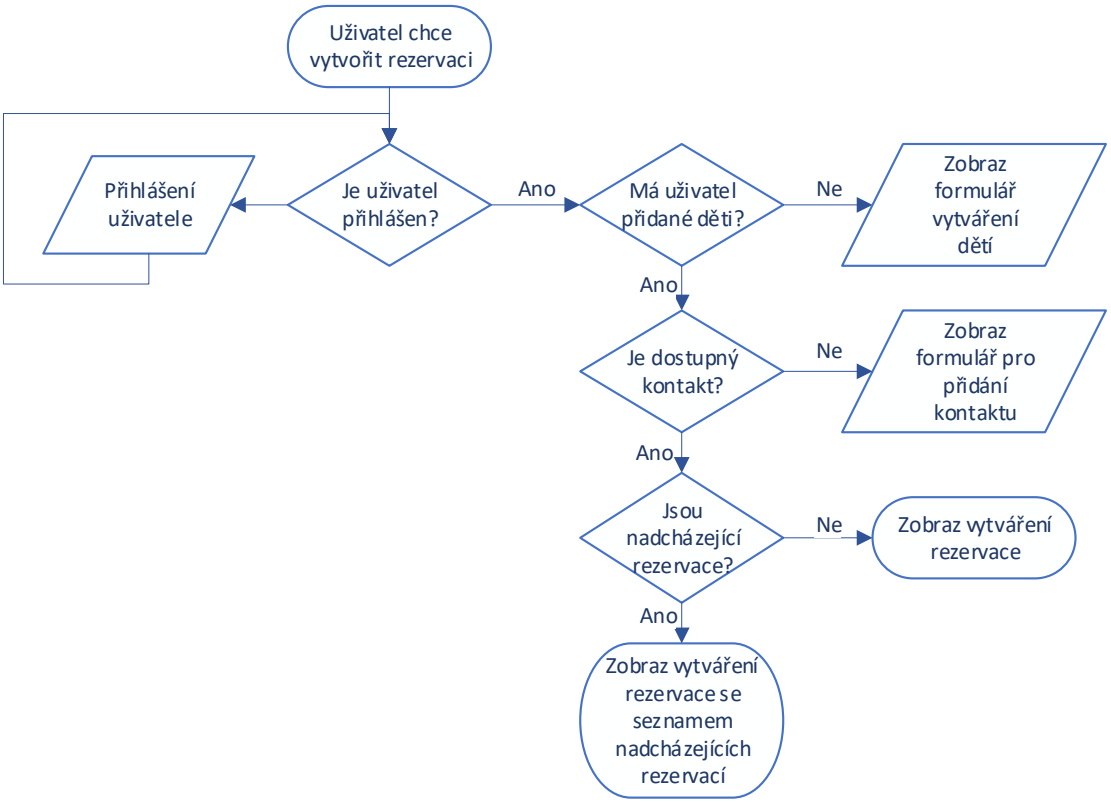
```
router.post('/getmonths', async (req, res) => {
  if (req.session.loggedin) {
    let userID = await helpers.getUserID(req.session.userid),
        year = xss(req.body.year);

    if (year !== "") {
      res.json({months: await getMonths(userID, year)});
    } else res.json({error: "Nebyla dodána potrebná data."});

  } else res.status(403).json({error: "Musíte být prihlásen."});
});
```

Listing 4.1: Ukázka routeru zpracovávající POST request na vypsání měsíců s provedenými rezervacemi ve zvoleném roce, využívající volání asynchronní funkce, čekající na odpověď databáze

4.1.1 Provádění uživatele vytvářením rezervace

Aby byl pro uživatele proces vytváření rezervace pro děti co nejjednodušší, provádí rezervační systém uživatele postupně přes několik stránek s formuláři. Celý proces se řídí diagramem na obrázku 4.2. Cílem je, aby uživatel zadal postupně všechna data požadovaná pro zarezervování času hlídání v DK.

Jednotlivé routery si po kontrole dostupnosti požadovaných informací, předávají uživatele jednoduchými redirecty, viz ukázka kódu 4.2. Ukázka kódu byla z důvodu přehlednosti zkrácena a zjednodušena.



Obrázek 4.2: Proces provádění uživatele rezervací dětí

```
if (req.session.loggedin) {
  let lianeID = req.session.userid;

  db.query("SELECT * FROM children WHERE user_id IN (" +
           "SELECT id FROM users WHERE liane_ID = ?)",
           lianeID, (err, result) => {
    if (!err && result.length > 0) { //Uzivatel má deti
      res.render('reservations', {
        title: 'Spravovat rezervované casy',
        activeNavbar: "rezervace",
        name: req.session.name,
        surname: req.session.surname
      });
    } else res.redirect('/user/children'); //Uzivatel nemá deti
  });

} else res.redirect('/auth/login'); //Uzivatel není prihlásen
```

Listing 4.2: Ukázka předávání uživatele routerům při nesplnění nutných podmínek pro vytvoření rezervace

4.1.2 Sessions

Sessions neboli sezení je proces, kterým lze v bezstavovém HTTP protokolu identifikovat uživatele a poskytovat mu data, která mohou být dostupná pouze jemu. Uživateli je odesláno cookie, která si prohlížeč skladuje a které v hlavičce každého requestu odesílá na server. Informace o sezení nejsou uživateli odesílána a server tak jen páruje identifikační kód s existujícími daty o uživateli.[39] V aplikaci se o odesílání, přijímání a přidělení dat sezení, stará rozšíření knihovny Express, `express-session`.[24]

Po přihlášení uživatele je volán router `auth.js`, který uživatele nechává autentizovat přes Shibboleth. Po získání potřebných informací z hlaviček, je stejným routerem přiděleno k sezení několik informací o uživateli. Mezi tyto informace patří `boolean`, zdali je uživatel přihlášen, jeho emailová adresa (hodnota `eppn` z hlaviček) a jméno a příjmení pro zobrazení v menu aplikace. Router dále uživatele přesměruje na stánku s rezervacemi (`/user/reservations`), kde může uživatel ihned vytvořit rezervaci pro své dítě.

Odhlášení uživatele probíhá ve stejném routeru, přičemž je volaná adresa `/auth/logout`, která všechny proměnné sezení vynuluje. Uživatel je poté přesměrován na hlavní stránku.

4.2 Templatovací engine

Po získání informací s z databáze routerem, dochází k předání informací templatovacímu enginu EJS. Ten v připravené předloze (templatu) dosadí nastavená data a výsledný vyrenderovaný dokument odesílá uživateli. Šablony pro templatovací engine se nacházejí v adresáři `/views`, kde je připravený podadresář pro často používané šablony `/views/templates` a `/views/templates/forms`.

4.3 Public resources

Veškeré soubory kódu rezervačního systému nejsou veřejně dostupné, pokud nejsou otevřené jako statické adresy. Aby se mohla uživateli stránka správně vykreslit, je nutné otevřít několik adresářů, které budou veřejně dostupné. Takovým adresářům se nastavuje statická adresa.

Knihovnám nainstalovaným balíčkovacím systémem `npm`, je nutné nastavit i cestu, pod kterou budou dostupné. Otevřeny jsou tak adresáře knihovnám `Bootstrap`, `jQuery`, `bootstrap-datepicker`, `FontAwesome` a `FullCalendar` a `SimpleMDE`. Aby bylo možné zpřístupnit další soubory uživatelům, mezi které patří například provozní řád ve formátu PDF nebo soubor s CSS styly, je otevřen adresář `/public` včetně podadresářů.

```
1 app.use('/bootstrap', express.static(__dirname + '/node_modules/
    bootstrap/dist/')); //Bootstrap
2 app.use(express.static(path.join(__dirname, 'public')));
```

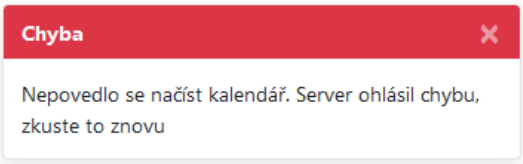
Listing 4.3: Otevření statického adresáře `/public` a souborů frameworku Bootstrap

Součástí adresáře `public` jsou adresáře `/images` a `/images/favicons` s obrázky pro rezervační systém a ikonkami, `scripts` se skripty pro funkce na straně klienta a `stylesheets` s dodatečnými CSS styly.

4.4 AJAX requesty

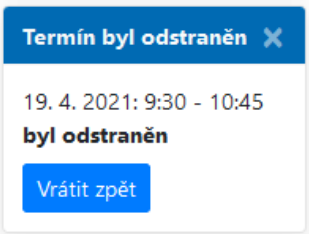
Součástí generovaného kódu klienta je i JavaScriptový kód, který primárně obsluhuje generování interaktivních modalů, formulářů pro zadávání dalších údajů, a komunikaci se serverem pomocí AJAXu.

Na stránkách, kde uživatel zadává nějaké údaje, dochází před odesláním k vytvoření nového `XMLHttpRequest` rozhraní, ve kterém je nastaven typ hlavičky a tělo zprávy, které budou serveru odeslány. Protože AJAX funguje asynchronnně a webová stránka po jeho načítání není obnovena, je v rozhraní nastaven event `onload`, který je vyvolán po obdržení odpovědi serveru s HTTP stavovým kódem 200, signalizujícím úspěšný požadavek.[40] V případě, že server ohlásí chybu, je uživatel o chybě informován buď zobrazením skrytého HTML elementu nebo zobrazením toast notifikace v pravém dolním rohu stránky, viz obr. 4.3.



Obrázek 4.3: Toast notifikace s oznámením chyby

V ukázce kódu 4.4 je znázorněno volání REST API serveru, vytvořením `POST XMLHttpRequestu`, na URL `/staff/removetime`, po kliknutí na potvrzovací tlačítko odstranění akce z kalendáře asistenta. Jakmile prohlížeč obdrží od serveru odpověď ve formátu JSON s potvrzením, dojde k odstranění časové akce z kalendáře, zavření otevřeného modalu a zobrazení potvrzovací toast notifikace, viz obr. 4.4. V případě chyby je vyvolaná toast notifikace s chybovou zprávou.



Obrázek 4.4: Toast notifikace s potvrzením odstranění času asistenta s tlačítkem pro vrácení provedené akce

```
const removeEvent = (eventID) => {
  let xhr = new XMLHttpRequest();
  xhr.open("POST", "/staff/removetime", true);
  xhr.setRequestHeader('Content-type', 'application/x-www-form-
    urlencoded');
  xhr.onload = () => {
    if (xhr.status === 200) {
      let response = JSON.parse(xhr.responseText);

      if (response.error) { //Nelze zmenu provést
        popupToast("Chyba", response.error, true);
      } else {
        calendar.getEventById(eventID).remove(); //Odebrani eventu
        $('#timeModal').modal('hide'); //Zavrení modalu
      }
    } else popupToast("Chyba", "Chyba serveru", true);
  };
  xhr.send(encodeURI(`eventID=${eventID}`));
}
```

Listing 4.4: Ukázka předávání uživatele routerům při nesplnění nutných podmínek pro vytvoření rezervace

4.5 Příprava serveru

Pro spuštění rezervačního systému DK, je nutné mít připravený server s běhovým prostředím Node.js a databázovým serverem MariaDB. Pro fungování odesílání emailových adres, je nutné mít k dispozici funkční program Sendmail. Pro pohodlnou správu a zajištění dostupnosti aplikace v případě chyby, je vhodné mít na serveru připraveného daemona, jako například PM2.

Systém rezervačního systemu DK je provozován na virtuálním počítači v rámci Virtul, s konfigurací 4 procesory, 4 GB operační paměti a 32 GB diskového prostoru. Virtuální počítač pohání operační systém Linux Mint 20 „Ulyana“.

4.5.1 Instalace MariaDB

K instalaci MariaDB byl využit balík `mariadb-server`, který byl nainstalován pomocí balíčkovacího systému `apt` systému Linux Mint. Po instalaci příkazem `apt install mariadb-server`, byl spuštěn instalační proces příkazem `mysql_secure_install`, ve kterém byl nastaven přístup pro uživatele systému `root`. Do databáze má `root` přístup pouze z lokálního počítače.

Po první instalaci byla vytvořena databáze `detskykoutek` pro rezervační systém a uživatel s přístupem do této databáze. Tomuto uživateli byla přidělena práva na základní operace nad tabulkami této databáze. Dále byly v databázi vytvořeny tabulky, pomocí DDL příkazů, vygenerovanými z návrhového softwaru *MySQL Workbench*. Tato sada příkazů je dostupná ve zdrojovém kódu aplikace v souboru `prepareDatabase.sql`.

```
CREATE DATABASE detskykoutek;
CREATE USER 'detskykoutek'@'localhost' IDENTIFIED BY 'heslo';
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, INDEX, DROP, ALTER,
    CREATE TEMPORARY TABLES, LOCK TABLES ON detskykoutek.* TO '
    detskykoutek'@'localhost';

-- Spustení souboru s DDL príkazy
SOURCE ./prepareDatabase.sql;
```

Listing 4.5: DDL příkazy k přípravě databáze a tabulek pro rezervační systém DK

4.5.2 Instalace Let's Encrypt SSL certifikátu

Pro zajištění SSL připojení k rezervačnímu systému DK, byl využit bezplatný certifikát autority *Let's Encrypt*, generovaný pomocí certifikačního nástroje *Certbot*.

K jeho spuštění bylo zapotřebí nainstalovat balíčkovací systém `snapd`, přes který je Certbot distribuován. Po instalaci `snapd` došlo k instalaci samotného Certbota příkazem `snap install --classic certbot`[41]

Protože provozujeme vlastní webový server a nepoužíváme Apache ani nginx, kterého by mohl Certbot k vytvoření certifkátu využít, a zároveň jej provozujeme na portu 80 a 443 (výchozí porty pro HTTP a HTTPS), bylo nutné přidat vlastní skript, který dočasně server rezervačního systému vypne a po certifikaci opětovně spustí. Server je spuštěný v deaomonu PM2, byl tedy využit příkaz tohoto daemona pro ukončení a opětovné spuštění služby, jak je uvedeno v kódu 4.6.

Cestu k vytvořeným klíčům `privkey.pem` a `fullchain.pem`, bylo poté nutné nastavit v konfiguračním souboru rezervačního koutku `settings.json`. V něm bylo možné nastavit i položku `https` na hodnotu `true`.

```
sudo sh -c 'printf "#!/bin/sh\npm2 stop koutek\n" > /etc/letsencrypt/
    renewal-hooks/pre/stoppm2koutek.sh'
sudo sh -c 'printf "#!/bin/sh\npm2 start koutek\n" > /etc/letsencrypt/
    renewal-hooks/post/startpm2koutek.sh'
sudo chmod 755 /etc/letsencrypt/renewal-hooks/pre/stoppm2koutek.sh
sudo chmod 755 /etc/letsencrypt/renewal-hooks/post/startpm2koutek.sh
# Provedení první certifikace
sudo certbot certonly --standalone
```

Listing 4.6: Příkazy k provedení úspěšné certifikace Certbotem

4.5.3 Instalace Node.js a spuštění rezervačního koutku

Balík `nodejs` není standardně součástí repozitářů, které jsou v základní instalaci distribuce Linux Mint, proto bylo nutné repozitář s Node.js dodat. Byla zvolena verze Node.js ve verzi 16.x. K instalaci nabízí *Nodesource* instalační bash skript, který repozitář přidá automaticky. Následně bylo možné balík nainstalovat pomocí `apt` package manageru.

Po přenesení souborů rezervačního systému do pracovního adresáře bylo nutné nainstalovat potřebné balíčky knihoven. Nainstalované prostředí Node.js obsahuje

vlastní balíčkovací systém `npm`. Příkazem `npm install` se nainstalovaly potřebné balíčky v požadovaných verzích, podle souboru `package.json`.

Aby bylo možné rezervační systém udržovat při běhu a mít nad serverem větší dohled v případě pádu, je vhodné využít daemonizujícího package manageru PM2. Ten byl nainstalován využitím `npm`. Celá aplikace byla zavedena do běhu spuštěním právě přes PM2. Všechny provedené příkazy jsou uvednené v kódu 4.7.

```
curl -fsSL https://deb.nodesource.com/setup_16.x | bash -
apt-get install nodejs

npm install pm2 -g

pm2 start npm --name "koutek" -- start
```

Listing 4.7: Příkazy použité k instalaci prostředí Node.js s daemonem PM2

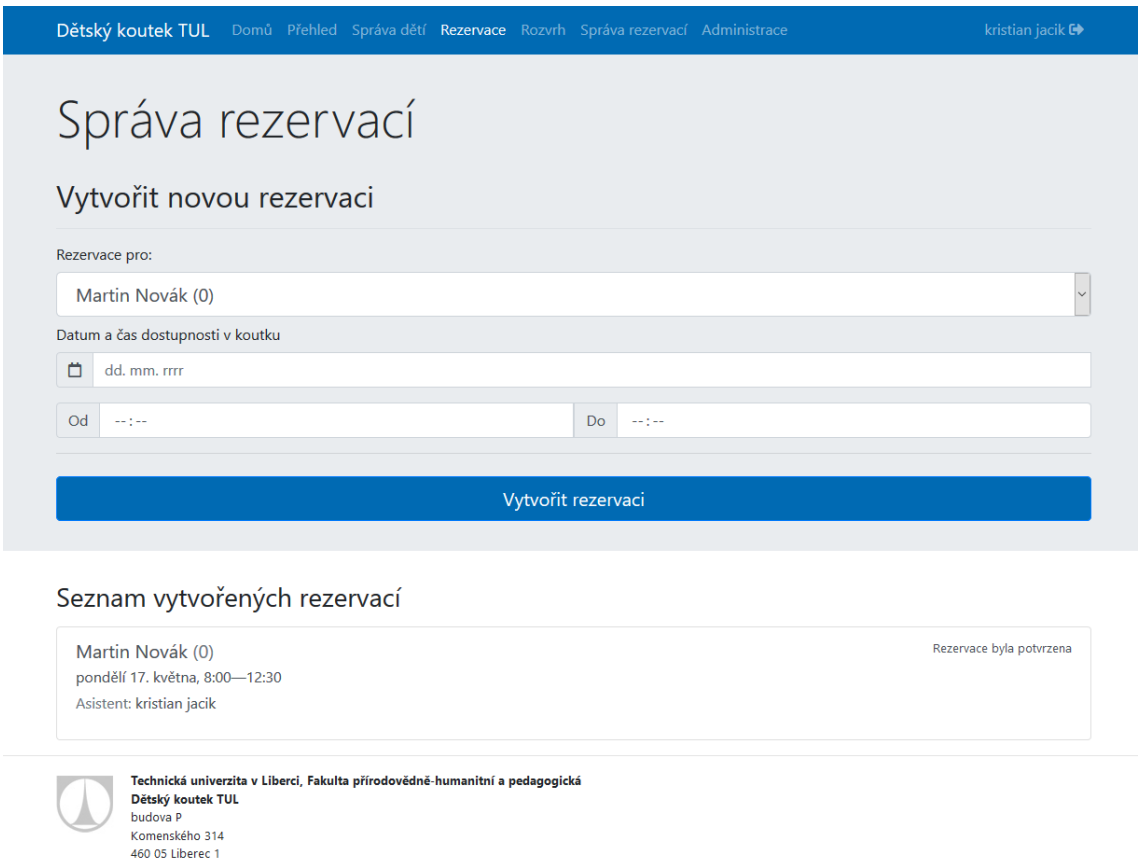
```
root@koutek2:~# pm2 list
```

id	name	namespace	version	mode	pid	uptime	↺	status	cpu	mem	user	watching
0	koutek	default	N/A	fork	133613	3h	100	online	0%	58.3mb	root	disabled

Obrázek 4.5: Běžící rezervační systém v package manageru PM2

4.6 Ukázka implementace

Webová stránka je dostupná na webové adrese https://koutek2.nti.tul.cz. Zde je možné se přihlásit pomocí Shibbolethu a jako rodič si přidat vlastní děti do systému a těm vytvořit žádosti o rezervace. Vedení dětského koutku může v aplikaci přidat správce a administrátory.



Obrázek 4.6: Vytváření a správa rezervací na stránce pro rodiče

5 Testování aplikace

Následující kapitola je věnována testování zabezpečení systému rezervačního koutku, s využitím automatizovaných veřejně dostupných testů, testů kompatibility zobrazení v různých internetových prohlížečích a způsobům zajištění očištění vstupních dat uživateli rezervačního systému.

5.1 Automatizované testy

K provedení testů byly použity nástroje *Hardenize*[1], *Mozilla Observatory*[2], *Security Headers*[3], *ImmuniWeb*[4] a *Qualsys SSL Labs*[5].

Pro dosažení vysokých skóre bylo nutné, aby na stránce byl zprovozněný HTTPS protokol, cookies měly nastavené zabezpečující parametry `same-site: strict` a `secure` a nastavené CSP hlavičky pro zabezpečení na straně prohlížeče, což vyžadovalo i specifické úpravy zdrojového kódu rezervačního systému tak, aby všechny skripty byly načítány zvlášť, z podadresáře na stejné doméně.

- Výsledky testu Hardenize se v několika málo položkách odvíjí od nastavení nadřazené domény tul.cz. Výsledky vycházející pro zabezpečení dětského koutku, vychází jako zelené, jako „Test passed“.
- Mozilla Observatory vyhodnocuje webovou stránku známkou **A+**.
- Security Headers vyhodnocuje webovou aplikaci známkou **A**.
- ImunniWeb vyhodnocuje webovou aplikaci známkou **A**.
- Qualsys SSL Labs vyhodnocuje SSL webové aplikace známkou **A+**.

V případě, že uživatel najde bezpečnostní problém, je k dispozici standardizovaný textový dokument `.well-known/security.txt` s kontaktními údaji k nahlášení detailů o problému.

[1] Dostupné na https://www.hardenize.com/report/koutek2.nti.tul.cz/1621178097
[2] Dostupné na https://observatory.mozilla.org/analyze/koutek2.nti.tul.cz
[3] Dostupné na https://securityheaders.com/?q=koutek2.nti.tul.cz&followRedirects=on
[4] Dostupné na https://www.immuniweb.com/ssl/koutek2.nti.tul.cz/VmvIcx47/
[5] Dostupné na https://www.ssllabs.com/ssltest/analyze.html?d=koutek2.nti.tul.cz

5.2 Validita vstupních dat

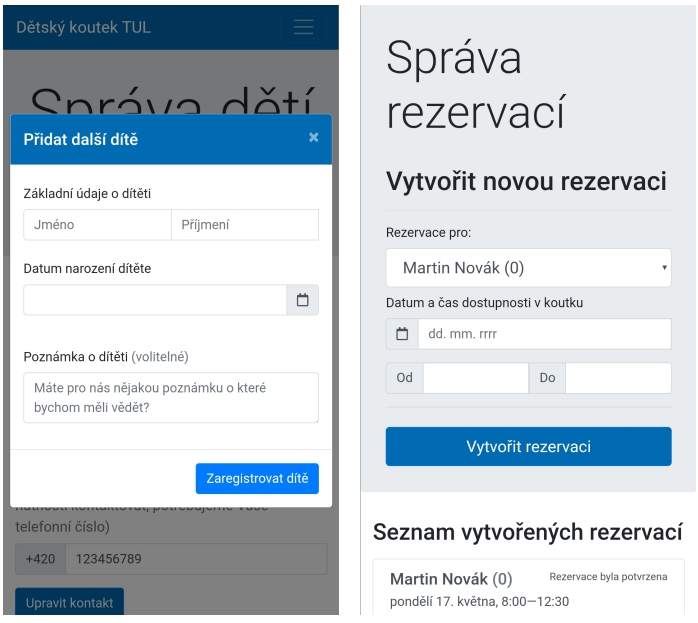
Při zadávání jakýchkoli dat, nelze důvěřovat uživateli, že zadávané údaje jsou správné. I přes kontroly, které jsou součástí formulářů (například u zadávání telefonního čísla nebo emailové adresy), je nutné data ověřit především na straně serveru. V případě, že zadávaná data nejsou platná, je uživatel v REST API informován o neplatnosti dat s chybovou hláškou.

Součástí ověření vstupních dat jsou i dva regulární výrazy pro ověření platnosti telefonních čísel a emailových adres v administraci, které musí být v doméně `@tul.cz`. Telefonní číslo může být i v mezinárodním formátu, vzhledem k tomu, že rodičem může být zahraniční student se zahraničním telefonním číslem.

Veškerá vstupní data jsou na straně serveru z bezpečnostních důvodů očišťována knihovnou `xss`, aby bylo předejito nechtěným XSS útokům, kdy by hrozilo vložení cizího kódu do aplikace. SQL příkazy využívají funkce prepared statement, kdy jsou parametry dodávány s escapovanými znaky a výrazy tak, aby nebylo možné škodlivým kódem, manipulovat s daty v databázi.

5.3 Test kompatibility prohlížečů

Požadavkem na webovou aplikaci bylo zajištění pohodlnějšího ovládání aplikace na mobilních zařízeních. Testování bylo provedeno automatizovanými, volně dostupnými nástroji *Test použitelnosti v mobilech společnosti Google*[6], *Lighthouse*, který je integrovaný v prohlížečích s jádrem Chromium ve verzi pro mobilní telefony a desktopové prohlížeče a testováním přímo na mobilním telefonu, viz obr. 5.1.



Obrázek 5.1: Přidávání dětí a vytváření nových rezervací na mobilním telefonu

[6]Dostupné na https://search.google.com/test/mobile-friendly?id=dBb2kwdIIRwwXBff-VG42g

Vzhledem k nejčastěji předpokládané situaci, kdy rodič potřebuje nutně vytvořit rezervaci pro dítě z mobilního telefonu, byla zvlášť v testovacím nástroji Lighthouse pro mobilní telefony, otestovaná stránka s vytvářením nové rezervace.

- Test použitelnosti v mobilech vypisuje výsledek „Stránka je optimalizovaná pro mobily“
- Lighthouse ve verzi pro mobilní telefony vypisuje výsledky Performance: **63/100**, Accesibility **98/100**, Best practices: **93/100**, SEO: **92/100**
- Lighthouse ve verzi pro desktopové prohlížeče vypisuje výsledky Performance: **88/100**, Accesibility **95/100**, Best practices: **93/100**, SEO: **90/100**
- Lighthouse ve verzi pro mobilní telefony na stránce s vytvořením rezervace, vypisuje výsledky Performance: **75/100**, Accesibility **99/100**, Best practices: **100/100**, SEO: **92/100**

6 Závěr

Výstupem této bakalářské práce je nový rezervační systém pro Dětský koutek Technické univerzity v Liberci, čímž bylo splněno zadání.

Práce v rešeršní části mapuje existující řešení rezervačních systémů, od veřejně dostupných, přes aktuálně univerzitou používaných rezervačních systémů, po stávající řešení. Popisuje jakou funkcionalitu nabízí a porovnává s funkcemi, které jsou pro systém dětského koutku specifické. Z provedené rešerše, vedených konzultací s vedením Dětského koutku a zadání práce, stanovuje požadavky na systém a novou aplikaci rezervačního systému.

Následující část práce je věnována popisu použitých technologií spolu s odůvodněním volby těchto technologií a knihoven. Zároveň detailně popisuje nejdůležitější části systému, mezi které patří databázový model, tři rozdílné role uživatelů s jejich specifickými pohledy a další funkce, které jsou pro tento rezervační systém specifické.

Poslední části práce jsou věnované samotné implementaci a testování zabezpečení a kompatibility rezervačního systému. Součástí implementace je i popis přípravy běhového prostředí, díky kterému je možné rezervační systém zprovoznit na jakémkoliv jiném serveru.

Při vytváření aplikace byl brán zřetel na uživatele aplikace tak, aby uživatelské prostředí bylo příjemné na používání a pro nezkušené uživatele maximálně intuitivní. Zároveň byl kladen důraz na využití dnešních moderních nástrojů pro tvorbu webových aplikací a webových serverů.

Výsledkem této práce je aplikace se serverem rezervačního systému, která může být nasazena jako nový systém pro Dětský koutek TUL. Hlavní přínosy této aplikace spočívají ve zjednodušení procesu vytváření žádostí o rezervaci času pro své děti v dětském koutku, umožnění asistentům dětského koutku sledovat nadcházející rezervace a spravovat svůj plán dostupnosti v dětském koutku, nové vedení poznámek k dětem ze strany rodiče a správce, zjednodušení procesů přiřazování asistentů k dětem pro správce dětského koutku a zpřístupnění aplikace na mobilní zařízení.

Vzhledem k použitým technologiím je možné aplikaci v budoucnu rozšiřovat o další funkce, které by mohly být pro vedení dětského koutku užitečné. Takovými může být implementace platební brány nebo automatizace přiřazování dětí asistentům. Díky použitým moderním technologiím lze aplikaci v případě potřeby transformovat i na rezervační systémy pro jiné účely.

Použitá literatura

[1] SOVOVÁ, Ilona. *Provozní řád Dětského koutku (dále jen „DK“) na Fakultě přírodovědně-humanitní a pedagogické Technické univerzity v Liberci* [online]. 2009 [cit. 2021-05-01]. Dostupné z: https://koutek.tul.cz/provozni_rad_dk.pdf.

[2] *O nás* [online]. Technická univerzita v Liberci, 2010 [cit. 2021-05-10]. Dostupné z: http://koutek.fp.tul.cz/o-nas.

[3] *Školka a dětský koutek* [online]. Technická univerzita v Liberci, 2021 [cit. 2021-05-10]. Dostupné z: https://www.tul.cz/spoluprace/sluzby/skolka-a-detsky-koutek/.

[4] ANSORG, Matthias. *List: Open source software for resource scheduling and booking* [online]. 2020 [cit. 2021-04-20]. Dostupné z: https://edgeryders.eu/t/list-open-source-software-for-resource-scheduling-and-booking/6629#heading--1-3.

[5] *Očkování Covid-19: Reservatic* [online]. Reservatic, c2015-2021 [cit. 2021-05-10]. Dostupné z: https://reservatic.com/cs/ockovani.

[6] TSELEGIDIS, Alex. *Easy!Appointments v1.4.1: Open Source Appointment Scheduler* [online]. 2021 [cit. 2021-04-23]. Dostupné z: https://demo.easyappointments.org/index.php/backend.

[7] TSELEGIDIS, Alex. *Features: An overview of Easy!Appointments available features.* [Online]. 2021 [cit. 2021-04-23]. Dostupné z: https://easyappointments.org/features/.

[8] *Shibboleth* [online]. Technická univerzita v Liberci, 2020 [cit. 2021-05-10]. Dostupné z: https://liane.tul.cz/cz/uzivatel/shibboleth.

[9] *Rezervace na antigenní testování* [online]. Technická univerzita v Liberci, 2021 [cit. 2021-05-10]. Dostupné z: https://covid.tul.cz/.

[10] *HTML: Living Standard* [online]. WHATWG (Apple, Google, Mozilla, Microsoft), 2021 [cit. 2021-04-24]. Dostupné z: https://html.spec.whatwg.org/#history-2.

[11] *Dětský koutek TUL* [online]. Technická univerzita v Liberci, [n.d.] [cit. 2021-05-10]. Dostupné z: https://koutek.tul.cz/index.php.

[12] *MariaDB versus MySQL: Compatibility* [online]. MariaDB, 2021 [cit. 2021-04-24]. Dostupné z: https://mariadb.com/kb/en/mariadb-vs-mysql-compatibility/.

[13] *MDN Web Docs*. HTML: HyperText Markup Language [online]. Mozilla a individual contributors, 2021 [cit. 2021-04-24]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/HTML.

[14] JANOVSKÝ, Dušan. *Jak psát web*. CSS styly - úvod [online]. ©1999 - 2021 [cit. 2021-04-25]. ISSN 1801-0458. Dostupné z: https://www.jakpsatweb.cz/css/css-uvod.html.

[15] OTTO, Mark a Jacob THORNTON. *Bootstrap: The most popular HTML, CSS, and JS library in the world.* [Online]. 2021 [cit. 2021-04-25]. Dostupné z: https://getbootstrap.com/.

[16] *Bootstrap-datepicker* [online]. eternicode, 2016 [cit. 2021-05-10]. Dostupné z: https://fullcalendar.io.

[17] *FullCalendar* [online]. FullCalendar, 2021 [cit. 2021-05-10]. Dostupné z: https://fullcalendar.io.

[18] *MDN Web Docs*. JavaScript [online]. Mozilla a individual contributors, 2021 [cit. 2021-04-25]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/JavaScript.

[19] ŠTRÁFELDA, Jan. *AJAX* [online]. [N.d.] [cit. 2021-05-10]. Dostupné z: https://www.strafelda.cz/ajax.

[20] *JQuery* [online]. OpenJS Foundation a jQuery contributors, 2021 [cit. 2021-05-10]. Dostupné z: https://jquery.com/.

[21] KHCHICH, Mouad. Do we still need jQuery in 2020? In: *DEV Community* [online]. DEV Community, c2016 - 2021 [cit. 2021-05-12]. Dostupné z: https://dev.to/mouadkh9/do-we-still-need-jquery-in-2020-4n48.

[22] *About Node.js®* [online]. OpenJS Foundation, 2021 [cit. 2021-05-12]. Dostupné z: https://nodejs.org/en/about/.

[23] *Express: Node.js web application framework* [online]. StrongLoop, IBM, a other expressjs.com contributors, 2021 [cit. 2021-05-12]. Dostupné z: https://expressjs.com/.

[24] WILSON, Douglas. *Express-session* [online]. 2020 [cit. 2021-05-12]. Dostupné z: https://www.npmjs.com/package/express-session.

[25] BALDWIN, Adam a Evan HAHN. *Helmet* [online]. 2021 [cit. 2021-05-13]. Dostupné z: https://www.npmjs.com/package/helmet.

[26] EERNISSE, Matthew. *EJS* [online]. 2021 [cit. 2021-05-13]. Dostupné z: https://ejs.co/.

[27] SATRAPA, Pavel. Shibboleth: Identifikujte se jen jednou. *Lupa.cz: Server o českém Internetu* [online]. c1998-2021 [cit. 2021-05-13]. ISSN 1213-0702. Dostupné z: https://www.lupa.cz/clanky/shibboleth/.

[28] *About MariaDB Software* [online]. MariaDB, 2021 [cit. 2021-05-14]. Dostupné z: https://mariadb.com/kb/en/about-mariadb-software/.

[29] WIKIPEDIE. *MariaDB — Wikipedie: Otevřená encyklopedie* [online]. 2021 [cit. 2021-05-14]. Dostupné z: https://cs.wikipedia.org/w/index.php?title=MariaDB&oldid=19805220.

[30] CHACON, Scott. In: *Pro Git*. CZ.NIC, c2009, s. 15–19. CZ.NIC. ISBN 978-80--904248-1-4. Dostupné také z: https://knihy.nic.cz/files/edice/pro_git.pdf.

[31] *Dependabot: Automated dependency updates for your Ruby, Python, JavaScript, PHP, .NET, Go, Elixir, Rust, Java and Elm.* [Online]. GitHub, 2019 [cit. 2021-05-13]. Dostupné z: https://dependabot.com.

[32] SKŘIVAN, Jaromír. Databáze a jazyk SQL. *Interval.cz* [online]. 2021 [cit. 2021-05-13]. Dostupné z: https://www.interval.cz/clanky/databaze-a-jazyk-sql/.

[33] REINMAN, Andris. *Nodemailer* [online]. 2021 [cit. 2021-05-14]. Dostupné z: https://nodemailer.com/.

[34] *Sendmail Open Source* [online]. Proofpoint, 2021 [cit. 2021-05-14]. Dostupné z: https://www.proofpoint.com/us/products/email-protection/open-source-email-solution.

[35] ŠTRÁFELDA, Jan. *WYSIWYG* [online]. [N.d.] [cit. 2021-05-10]. Dostupné z: https://www.strafelda.cz/wysiwyg.

[36] COSSICK, Wes. *SimpleMDE: Markdown Editor* [online]. Sparksuite, 2016 [cit. 2021-05-13]. Dostupné z: https://www.npmjs.com/package/simplemde.

[37] SANTOS, Estevão Soares dos. *Showdownjs: A markdown to HTML converter* [online]. showdownjs, c2017-2019 [cit. 2021-05-12]. Dostupné z: http://showdownjs.com/.

[38] MALÝ, Martin. REST: architektura pro webové API. *Zdroják: o tvorbě webových stránek a aplikací* [online]. 2021 [cit. 2021-05-12]. ISSN 1803-5620. Dostupné z: https://zdrojak.cz/clanky/rest-architektura-pro-webove-api/.

[39] ALPHARD a Michal MENCL. Sessions. *Péhápko.cz: učebnice skriptovacího jazyka PHP* [online]. [N.d.] [cit. 2021-05-15]. Dostupné z: http://www.pehapko.cz/programujeme-v-php/sessions.

[40] FIELDING, Roy T. a Julian RESCHKE. *Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. 200 OK [RFC 7231]. RFC Editor, 2014 [cit. 2021-05-16]. Request for Comments, č. 7231. Dostupné z DOI: 10.17487/RFC7231.

[41] *Certbot: An automatic client for enabling HTTPS on your website* [online]. Electronic Frontier Foundation (EFF), 2021 [cit. 2021-05-16]. Dostupné z: https://certbot.eff.org/.

Seznam obrázků

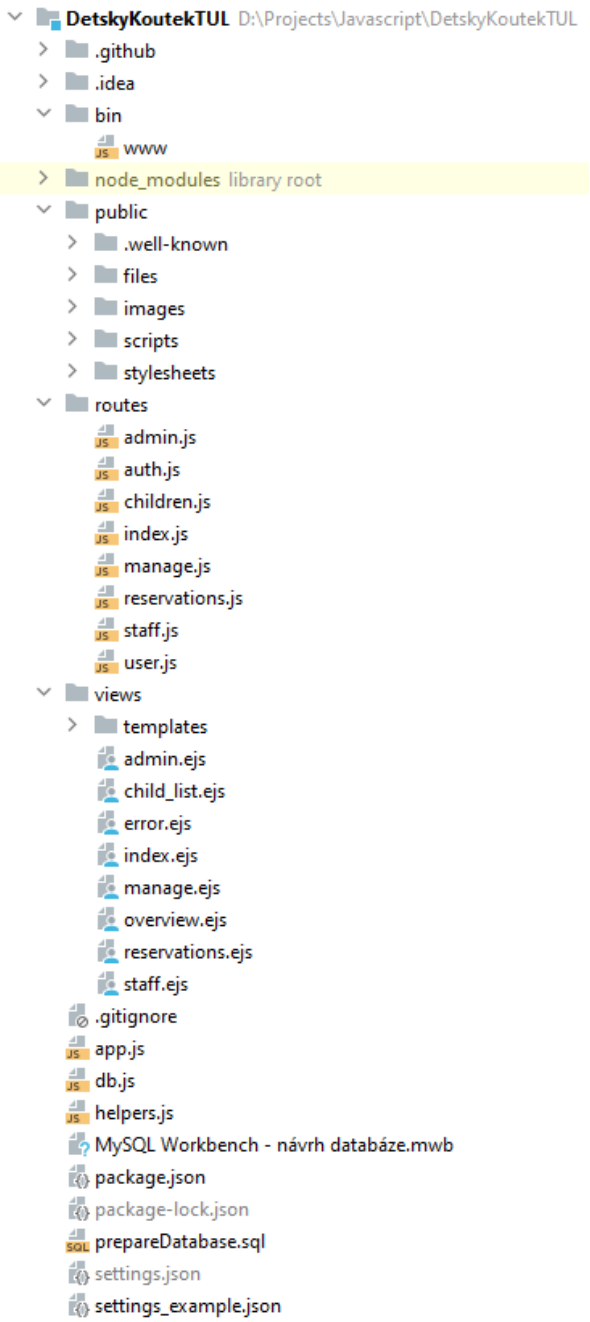
2.1 Aplikace rezervačního systému pro antigenní testování zaměstnanců TUL . . . 13
2.2 Původní aplikace rezervačního systému Dětského koutku . . . 15

3.1 ER diagram databázového modelu . . . 22
3.2 Seznam dětí a úprava telefonního kontaktu . . . 24
3.3 Přehled hlídání pro asistenty s rozvrhem dostupnosti . . . 25
3.4 Stránka pro správce s potvrzováním rezervací . . . 26
3.5 Administrace s výpisy a seznamy asistentů . . . 27
3.6 Výpis časových průniků asistentů s časem rezervace . . . 28
3.7 Ukázka potvrzujícího emailu . . . 29
3.8 Ukázková tabulka z CSV výstupu . . . 29
3.9 WYSIWYG editor Alertu v Markdownu . . . 30

4.1 ER diagram databázového modelu . . . 32
4.2 Proces provádění uživatele rezervací dětí . . . 33
4.3 Toast notifikace s oznámením chyby . . . 35
4.4 Toast notifikace s potvrzením odstranění času asistenta . . . 35
4.5 Běžící rezervační systém v package manageru PM2 . . . 38
4.6 Správa rezervací na stránce pro rodiče . . . 39

5.1 Snímky obrazovky aplikace na mobilním telefonu . . . 41

Příloha: Adresářová struktura aplikace



Příloha: Zdrojový kód

Součástí práce je zdrojový kód rezervačního systému Dětského koutku, přiložený v archivu `DetskyKoutekTUL.zip`.