



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

DEPARTMENT OF INTELLIGENT SYSTEMS

REZERVAČNÍ SYSTÉM SPORTOVIŠTĚ

SPORTS GROUND RESERVATION SYSTEM

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

JURAJ REMEŇ

VEDOUcí PRÁCE

SUPERVISOR

Ing. ALEŠ SMRČKA, Ph.D.

BRNO 2024

Zadání bakalářské práce



156442

Ústav: Ústav inteligentních systémů (UITS)
Student: **Remeň Juraj**
Program: Informační technologie
Název: **Rezervační systém sportoviště**
Kategorie: Informační systémy
Akademický rok: 2023/24

Zadání:

1. Seznamte se s dostupnými řešeními rezervačních systémů sportovišť. Nastudujte aktuální technologie pro vývoj a nasazení informačních systémů s desktopovými a mobilními klienty.
2. Analyzujte požadavky TJ Ochoz u Brna na rezervační systém jejich sportovišť. Podrobně navrhnete řešení rezervačního systému.
3. Implementujte rezervační systém. Hlavním nefunkčním kritériem výběru technologie a implementace je udržitelnost a zabezpečení.
4. Navrhnete a implementujte automatické testy a automatické nasazení systému. Podrobně zdokumentujte systém pro uživatele a administrátorskou podporu.

Literatura:

- E. Pelivani, A. Besimi and B. Cico, "A Comparative Study of UI Testing Framework," *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022, pp. 1-5, doi: 10.1109/MECO55406.2022.9797165.

Při obhajobě semestrální části projektu je požadováno:
První dva body zadání.

Podrobné závazné pokyny pro vypracování práce viz <https://www.fit.vut.cz/study/theses/>

Vedoucí práce: **Smrčka Aleš, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1.11.2023
Termín pro odevzdání: 9.5.2024
Datum schválení: 6.11.2023

Abstrakt

Cielom bakalárskej práce je digitalizácia agendy rezervácie športovísk, primárne založené na požiadavkách TJ Ochoz u Brna, ale systém je aplikovateľný aj pre iné malé alebo stredné športové kluby. Systém má za úlohu zprehľadniť časový harmonogram športovísk v obci, uľahčiť vytvorenie rezervácie na dané športoviská pre obyvateľov a odbremeniť zodpovedné osoby od udržiavania a používania rôznych systémov na správu rezervácií, ako napríklad rôzne Google či Excel tabuľky alebo zapisovanie na obyčajný papier.

Abstract

The aim of the bachelor thesis is the digitalization of the sports ground booking agenda, primarily based on the requirements of TJ Ochoz u Brna, but the system is also applicable to other small or medium-sized sports clubs. The system is intended to make the timetable of sports grounds in the municipality more transparent, to facilitate the creation of bookings for the sports grounds in question for the residents and to relieve the responsible persons from maintaining and using different systems for managing bookings, such as various Google or Excel spreadsheets or writing them down on plain paper.

Kľúčové slová

Informačný systém, rezervačný systém, šport, rezervácia, športovisko, Python, Django

Keywords

Information system, reservation system, sport, reservation, sports ground, Python, Django

Citácia

REMEŇ, Juraj. *Rezervační systém sportoviště*. Brno, 2024. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Aleš Smrčka, Ph.D.

Rezervační systém sportoviště

Prehlásenie

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing, Aleša Smrčku Ph.D. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Juraj Remeň
8. mája 2024

Podakovanie

Obrovská vďaka patrí aj Ing, Alešovi Smrčkovi Ph.D. za neustálu pomoc a konzultácie pri implementácii jednotlivých prvkov. Osobne chcem poďakovať aj môjmu bratovi Mgr. Marekovi Remeňovi, ktorý vo svojom osobnom voľnom čase vypracoval logo, ktoré je použité pre tento systém. Takisto by som chcel poďakovať ľuďom, ktorí sa podieľali na UX testovaní, za ich spätnú väzbu, ktoré mi pomohlo odstrániť nepraktické veci zo systému.

Obsah

1	Úvod	5
2	Analýza požiadaviek systému	7
2.1	aktéri a ich diagramy prípadov použitia v systéme	10
2.1.1	Diagram prípadov použitia popisujúci chovanie administrátora systému	10
2.1.2	Diagram prípadov použitia popisujúci chovanie vedúceho klubu . . .	12
2.1.3	Diagram prípadov použitia popisujúci chovanie športovca	13
2.1.4	Diagram prípadov použitia popisujúci chovanie registrovaného užívateľa	14
2.1.5	Diagram prípadov použitia popisujúci chovanie neregistrovaného užívateľa	15
2.1.6	Diagram prípadov použitia popisujúci chovanie systému ako pasívneho aktéra	16
2.2	Entity a atribúty a ich zakotvenie v systéme	17
2.2.1	Užívatelia	17
2.2.2	Športovisko	18
2.2.3	Klub	18
2.2.4	Rezervácia	19
2.2.5	Tag	19
3	Návrh systému	20
3.1	MVC	20
3.2	ER-diagram systému	22
3.3	Užívateľ	24
3.4	Klub	24
3.5	Rezervácia	25
3.6	Športovisko	25
4	Implementačné detaily rezervačného systému	26
4.1	Technológie použité v projekte	26
4.1.1	Databáza	26
4.1.2	Server	27
4.1.3	Grafické užívateľské rozhranie	27
4.1.4	Docker	28
4.2	Implementácia databázy (Models)	29
4.2.1	Užívatelia	29
4.2.2	Športovisko	29
4.2.3	Rezervácia	30
4.2.4	Klub	30

4.2.5	Vedenie klubu	30
4.2.6	Členstvo v klube	31
4.3	Implementácia serveru (Views)	31
4.3.1	Aplikácia Domov	31
4.3.2	Vyhľadávanie	33
4.3.3	Aplikácia Správy užívateľov	33
4.3.4	Aplikácia Správy rezervácií	36
4.3.5	Aplikácia Správy športovísk	39
4.3.6	Aplikácia Správy klubov	40
4.3.7	Admin	45
4.4	Implementácia grafického užívateľského rozhrania (Templates)	45
4.4.1	Domov	46
4.4.2	Vyhľadávanie	51
4.4.3	Správa užívateľov	54
4.4.4	Rezervácia	58
4.4.5	Športovisko	66
4.4.6	Klub	66
4.4.7	Admin	75
4.5	Automatické nasadenie systému	77
4.5.1	Certifikácia	78
5	Testovanie	80
5.1	Automatické testy	80
5.1.1	Backend testy jednotlivých aplikácií	81
6	Záver	86
	Literatúra	87
	A Obsah priloženého pamäťového média	89
	B Inštalácia, konfigurácia a spustenie	90
	C Diagramy	92
	D Stránky	101
	E Abstrakt adresárovej štruktúry	105

Zoznam obrázkov

2.1	Hierarchia aktérov systému	10
2.2	Diagram prípadov použitia administrátora	11
2.3	Diagram prípadov použitia vedúceho klubu	12
2.4	Diagram prípadov použitia športovca	13
2.5	Diagram prípadov použitia registrovaného užívateľa	14
2.6	Diagram prípadov použitia neregistrovaného užívateľa	15
2.7	Diagram prípadov použitia systému ako pasívneho aktéra	16
3.1	Diagram MVC	21
3.2	Diagram MVT	22
3.3	ER diagram popisujúci programovaný systém	23
4.1	Domovská stránka administrátora	47
4.2	Rozdiely na domovskej stránke pre vedúceho klubu	48
4.3	Rozdiely na domovskej stránke pre športovca	48
4.4	Rozdiely na domovskej stránke pre neregistrovaného užívateľa	49
4.5	Zmena dátumov v záhlaví	49
4.6	Stránka manuálu	50
4.7	Stránka s vyhľadávaním administrátora	52
4.8	Rozdiely na stránke vyhľadávania vedúceho klubu	53
4.9	Rozdiely na stránke vyhľadávania športovca	53
4.10	Rozdiely na stránke vyhľadávania neregistrovaného užívateľa	54
4.11	Stránka registrácie do systému	55
4.12	Aktiváciu účtu	55
4.13	Stránka prihlásenia do systému	56
4.14	Zabudnuté heslo	56
4.15	Resetovanie hesla	57
4.16	Úprava profilu	57
4.17	Vytvorenie rezervácie pre prihláseného užívateľa	58
4.18	Vytvorenie klubovej rezervácie pre vedúceho klubu alebo administrátora	59
4.19	Vytvorenie rezervácie pre neregistrovaného užívateľa	59
4.20	Vytvorenie účtu spolu s rezerváciou pre neregistrovaného užívateľa	60
4.21	Rozdiely v stránke s informáciami o cudzej rezervácii	61
4.22	Úprava a stornovanie vlastnej rezervácie	62
4.23	Úprava a stornovanie klubovej rezervácie	63
4.24	Tabuľky so všetkými mojimi rezerváciami	64
4.25	Tabuľky so všetkými rezerváciami	65
4.26	Info o športovisku	66
4.27	Stránka klubov pre registrovaného užívateľa	67

4.28	Stránka klubov pre administrátora	68
4.29	Rozdiely pri vytváraní nového klubu	69
4.30	Schválenie nového klubu	70
4.31	Poslanie alebo zrušenie žiadosti o pridanie do klubu	70
4.32	Schválenie žiadosti o pridanie do klubu	71
4.33	Poslanie alebo zrušenie žiadosti o pridanie do klubu	71
4.34	Schválenie žiadosti o pridanie do klubu	72
4.35	Opustenie klubu	72
4.36	Opustenie alebo odstránenie klubu vedúcim klubu	72
4.37	Rozdiely pri info o klube	75
4.38	Administrátorská správa užívateľov	76
4.39	Pridanie nového športoviska	76
4.40	Úprava alebo vymazanie športoviska	77
4.41	Samopodpísaný certifikát použitý v systéme	78
C.1	Životný cyklus vytvorenia klubu administrátorom	92
C.2	Životný cyklus vytvorenia klubu neregistrovaným užívateľom	93
C.3	Životný cyklus vytvorenia klubu registrovaným užívateľom	93
C.4	Sekvenčný diagram vytvorenia klubu registrovaným užívateľom	94
C.5	Sekvenčný diagram vytvorenia klubu neregistrovaným užívateľom	95
C.6	Diagram prípadov použitia celého systému	96
C.7	Diagram prípadov použitia registrovaných užívateľov	97
C.8	Vývojový diagram vytvorenia rezervácie	98
C.9	Vývojový diagram vytvorenia klubovej rezervácie	99
C.10	Vývojový diagram vyhľadávania	100
D.1	Domovská stránka vedúceho klubu	102
D.2	Domovská stránka športovca	103
D.3	Domovská stránka neregistrovaného užívateľa	104

Kapitola 1

Úvod

V dnešnej dobe je mnoho spôsobov ako športovať, takisto je mnoho ľudí, ktorí radi športujú a zároveň chcú športovať minimálne v dvojici. Na to, aby si človek mohol zahrať so svojim priateľom napríklad tenis potrebuje tenisový kurt, ten tenisový kurt musí mať nejakého správcu, alebo osobu, ktorá je zaň zodpovedná a vie aký časový harmonogram má dané ihrisko.

Ak je to niekto menej počítačovo zdatný, je viac než isté, že bude mať niekde na nástenke vyvesený papier s časmi, kedy je ihrisko obsadené a kedy voľné a hráči sa buď napíšu sami na daný papier, alebo správcovi zavolajú, opýtajú sa kedy je voľno a objednávajú si daný termín. V lepšom prípade má správca spravený zdieľaný tabuľkový dokument, kde vidí kto sa prihlásil, alebo koho prihlásil on.

No v ideálnom prípade, je jeho jedinou starostou aby si skontroloval informačný systém, v ktorom vidí kto a kedy príde hrať, koľko má zaplatiť, či je členom daného klubu, v tomto prípade nejakého spolku tenistov, alebo či si daný kurt objednal na celú sezónu dopredu. Takisto to nemusí byť, len s tenisovým kurtom, prvotný nápad k tejto práci bol rezervačný systém, ktorý budú využívať primárne členovia TJ Ochoz u Brna na prenájom multifunčného ihriska s umelou trávou.

Vzhľadom na to, že v obci vyrástla komunita športovcov aj v iných odvetviach a rozrástli sa takisto aj spolky, ktoré v obci pôsobia, sa tento systém rozšíril aj na iné športoviská či plochy na rôzne športové akcie, či už na spoločné kondičné posilňovanie, futbal, tenis, basketbal, atď.

Daný systém nebude na mieru urobený len pre danú obec, po konzultácii s vedúcim práce a poslancom môjho rodného mesta sme sa rozhodli daný systém ponúknuť na využívanie aj obyvateľom Hnúšte.

Práca je logicky usporiadaná do kapitôl ako postupovalo jeho vyvíjanie. V prvej kapitole je podrobná analýza požiadaviek, v ktorej sú konkrétne popísaní aktéri a takisto entity celého systému a jej súčasťou je jednak tabuľka požiadaviek a aj diagram prípadov použitia, ktoré detailne popisujú požadované chovanie systému.

V druhej kapitole sa čitateľ dozvedá o výsledom návrhu celého systému, ktorého súčasťou je ER diagram, popisujúci entity a vzťahy medzi nimi v systéme.

Tretia kapitola sa zaoberá implementáciou systému, na začiatku sú spomenuté použité technológie, ktoré boli použité pri implementácii, s ktorými sa čitateľ tejto práce stretne. Jednotlivé sekcie tejto kapitoly sa zaoberajú implementáciou častí celého systému, čiže implementácia databázy, serveru a užívateľského rozhrania.

Štvrtá kapitola sa venuje testovaniu celého systému, ktoré je aj v zadaní práce. Konkrétne je jedným z cieľom práce navrhnuť a implementovať automatické testy daného systému.

Kapitola 2

Analýza požiadaviek systému

Za účelom vytvorenia Rezervačného systému športovísk bolo potrebné oboznámiť sa s konkrétnymi požiadavkami zákazníka, ktorým bol v tomto prípade TJ Ochoz u Brna, v ktorej sa nachádzajú športoviská, ktoré sú využívané rôznymi skupinami ľudí. Občas teda môže nastať, že nemajú prehľad o využiteľnosti športoviska a tí čo prídu neohlásene, alebo pozdejšie majú teda smolu.

V súčasnosti aktuálny správca používa proprietárny systém na stránke [TenisOchoz](#) od spoločnosti SuperSaaS . Jedná sa o všeobecný systém, ktorý je upravený podľa požiadaviek a aktuálne ho na rezerváciu využívajú len tenisti. Chcel by ale rozšíriť svoje možnosti a preto hľadá iný spôsob systému, ktorý by bol využívaný všetkými skupinami. Takisto súčasný systém neumožňuje vytvorenie rezervácie pre neregistrovaných užívateľov a teda obmedzuje istú skupinu.

Vzhľadom na to že obec už má svoju doménu pre svoje športové aktivity, dá sa predpokladať, že by tento ich existujúci web zakomponoval prepojenie teraz vyvíjaného rezervačného systému na svoju stránku. Slúžila by ako informatívna a po kliknutí na tlačidlo rezervačného systému by bol užívateľ presmerovaný na stránku, ktorá bude mať na starosti zaobchádzanie s rezerváciami.

Pre tieto účely sme sa teda rozhodli vytvoriť daný systém na rezervovanie jednotlivých športovísk. Tento systém má takisto za úlohu pomôcť ľuďom s podobnými záujmami, ľahšie organizovať spoločné športové aktivity. Vzhľadom na to, že si viem predstaviť, že podobné situácie nemusia nastávať len v obci Ochoz u Brna, ale takisto po celej Českej republike, alebo napríklad aj na Slovensku. Sme sa spolu s vedúcim práce, hneď na prvom stretnutí, rozhodli vytvoriť systém, ktorý bude ľahko využiteľný aj pre iných zákazníkov, ktorí by podobné systémy chceli pre svoju obec získať. Ale finančná situácia im to neumožňuje z rôznych dôvodov, alebo proste doposiaľ podobný systém nepovažovali za potrebný pre svoju prácu a chod obce.

Jednou s takých obcí by v budúcnosti mohlo byť napríklad moje rodné mesto Hnúšťa, ktoré má mnoho športovísk, ktoré samo spravuje, alebo sú na území mesta, ale potenciálni zákazníci často nemajú prehľad ako je to s tým a tým športoviskom. Majú problémy napríklad so zistením otváracích hodín, cenou za prenájom, kontaktovaním zodpovednej osoby pre dané športovisko.

Tento informačný systém sa snaží tieto problémy eliminovať. Hlavnou úlohou systému je ukázať akémukoľvek užívateľovi všetky potrebné informácie o športoviskách v danej obci, od obsadenosti a online rezerváciu až po údaje o danom športovisku či kontakt na jeho správcu, alebo zodpovednú osobu.

Aktéri, entity a atribúty systému majú rôzne práva, ktoré sú presne popísané týmito tabuľkami:

Tabuľka 2.1: Tabuľka užívateľských práv

Aktér	Vyhľadanie	Vytvorenie	Schválenie	Odstránenie
Admin	✓	✓	✓	✓
Vedúci klubu	✓	✓		✓
Športovec	✓	✓		✓
Registrovaný	✓	✓		✓
Neregistrovaný	✓ - OVU	✓		

Skratka **OVU** znamená **okrem vyhľadávania užívateľov**.

Tabuľka 2.2: Tabuľka práv entít a atribútov

Entita	Vyhľadanie	Vytvorenie	Schválenie	Odstránenie
Užívateľ	✓ - ARU	✓		✓ - ONU
Športovisko	✓	✓		✓ - A
Klub	✓	✓	✓	✓ - VK/A
Rezervácia	✓	✓		✓ - ONU
Tag (atribút)	✓	✓		

Skratka **ARU** znamená **aspoň registrovaný užívateľ**, **ONU** znamená **okrem neregistrovaného užívateľa**, **A** znamená **administrátor**, **VK/A** znamená **vedúci klubu alebo administrátor**.

Celá analýza je zhrnutá v týchto tabuľkách:

Kategória	Požiadavka
Vyhľadávanie	Môžem vyhľadať len časť slova (primárne vyhľadávanie), alebo môžem zadať konkrétne celé slovo. Jedná sa o filter, takže formou checkboxou si môžem filtrovať, či chcem vidieť tabuľky s užívateľmi, klubmi, rezerváciami, športoviskami alebo tagmi.
Vedúci klubu	Môže vytvoriť alebo odstrániť klub. Môže vytvoriť klubové rezervácie, o ktorých sú emailom informovaní všetci členovia aj ostatní vedúci. Môže upraviť alebo odstrániť klubové rezervácie. Vidí kto všetko je členom, a môže schváliť užívateľov za členov alebo ďalšieho člena za ďalšieho vedúceho.
História	Užívateľ bude mať možnosť vidieť svoju históriu rezervácií. Vedúci klubu a admin bude mať možnosť vidieť históriu všetkých rezervácií.

Tabuľka 2.3: 1. tabuľka požiadaviek systému

Kategória	Požiadavka
Športovisko	Po kliknutí na tlačidlo športoviska sa objaví základné info o športovisku, , každé má svoj tag, ktorý je možné vyhľadať.
RU	Registrovaní užívatelia budú mať možnosť upraviť svoje profily a zmeniť si svoje osobné údaje.
Rezervácie	Pri novej rezervácii sa otvorí modálne okno s rezervačným formulárom. Ak je užívateľ prihlásený, jeho osobné údaje sú v danom formulári už prednastavené, ak nie je prihlásený/registrovaný, tak je prednastavený len dátum a čas podľa toho, na ktoré políčko v rozvrhu užívateľ klikol. Pri rezervácii sa zobrazí upozornenie, či je rezervácia úspešná alebo nie. Podobne ako v kalendárových aplikáciách je možné vytvoriť v rovnakom časovom okienku aj druhú rezerváciu, avšak na iné športovisko. Po kliknutí na už existujúcu rezerváciu sa zobrazí info o nej. Ak sa jedná o užívateľovu rezerváciu, je schopný si po kliknutí na ňu ju upraviť alebo kompletne stornovať. Po odoslaní formuláru bude na užívateľovu emailovú adresu odoslaný potvrdzovací email o rezervácii, v ktorom je info o jeho rezervácii a takisto iCal súbor na pridanie do svojho kalendára.
Admin	Vie pridať a odstrániť užívateľov. Vie pridať, upraviť a odstrániť kluby. Vie pridať, upraviť a odstrániť športoviská. Vie pridať, upraviť, stornovať a odstrániť rezervácie. Ak sa nejaké športovisko odstráni, tak sa všetky rezervácie na dané športovisko stornujú a odošle sa email o stornovaní užívateľom, ktorí majú športovisko rezervované. Ak sa nejaký klub odstráni, tak sa všetky rezervácie na športoviská, ktoré boli vytvorené pre daný klub stornujú a odošle sa email o odstránení klubu všetkým jeho členom a spolu s ním aj jeho rezervácie. Ak sa nejaký užívateľ odstráni, tak sa všetky jeho rezervácie stornujú a odošle sa email užívateľovi o tom, že bol jeho účet odstránený a spolu s ním aj jeho rezervácie.
Email	Pri registrácii bude na užívateľov email odoslaný aktivačný či potvrdzovací email. Pri rezervácii bude na užívateľov email odoslaný potvrdzovací email. Pri zmene rezervácie bude na užívateľov email odoslaný potvrdzovací email. Pri akejkoľvek zmene stavu klubu, alebo jeho nových členov alebo vedúcich je odoslaný email.
Kalendár	Užívateľ bude mať možnosť pridať si rezerváciu do svojho kalendára.
Tagy	Užívateľ ich môže vyhľadať, jedná sa o atribút každého modelu v systéme, slúžiaci na personalizáciu.
Kluby	Členstvo sa schvaľuje vedúcim, takisto aj vedúci klubu je schválený už existujúcim. Môže byť viacero klubov pre rovnaký šport, tag ale musí byť vždy jedinečný.

Tabuľka 2.4: 2. tabuľka požiadaviek systému

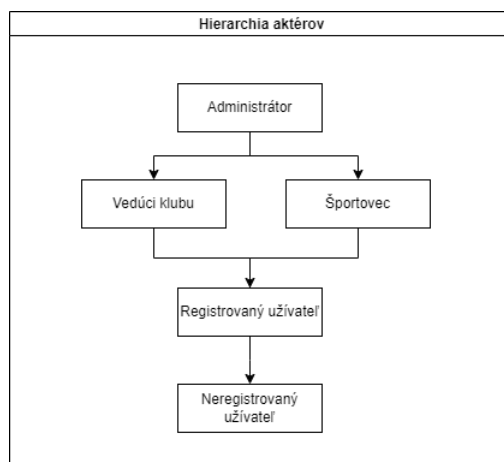
Kategória	Požiadavka
NU	Môžu si spraviť rezerváciu. Nemôže si ju upraviť alebo stornovať. Pri rezervácii si môže s danými údajmi z formuláru automaticky vytvoriť účet v systéme. Má možnosť si pozrieť športoviská a rezervácie druhých, ktoré neobsahujú žiadne osobné údaje.
Prenositelnosť	Systém bude bežať ako kontajner cez DOCKER čím sa zaručí, že sa systém bude môcť použiť aj v iných menších alebo stredných športových kluboch.
Udržateľnosť	Systémové požiadavky budú riadne odôvodnené a zabezpečené, aby sa systém mohol udržiavať. Systém bude mať možnosť pridávať nové funkcionality a rozširovať sa. Systém bude mať v sebe zabudovaný set testovania, ktorý bude testovať jeho funkcionality a takisto funkcionality jednotlivých komponentov.
Bezpečnosť	Systém bude mať zabudovanú certifikáciu Let's Encrypt. Na zabezpečenie bezpečnej registrácie a prihlásenie cez email je vytvorený samostatný backend email autentifikátor.

Tabuľka 2.5: 3. tabuľka požiadaviek systému
Skratka NU znamená **neregistrovaní užívatelia**, RU znamená **registrovaní užívatelia**.

2.1 aktéri a ich diagramy prípadov použitia v systéme

Jedným z výsledkov analýzy požiadaviek na systém je tento konkrétny Use Case Diagram (diagram prípadov použitia) C.6, ktorého celková podoba sa nachádza v prílohách. Pre sprehľadnenie bol celý diagram rozdelený na viacero častí po jednotlivých aktéroch.

Z daného diagramu C.6 a tabuľky užívateľských práv 2.1 a tabuľky práv entít 2.2 je možné zostaviť nasledujúcu hierarchiu aktérov v systéme:



Obr. 2.1: Hierarchia aktérov systému

2.1.1 Diagram prípadov použitia popisujúci chovanie administrátora systému

Administrátor môže v systéme vykonávať nasledujúce akcie a dedí všetky akcie aj od hierarchicky podriadených aktérov systému:

Obr. 2.2: Diagram prípadov použitia administrátora

- **Pridať užívateľa** - administrátor môže na základe komunikácie s potenciálnym užívateľom mimo systém vytvoriť pre daného užívateľa užívateľský profil, ku ktorému potom daný užívateľ dostane prístup.

Súčasťou tohto prípadu použitia je aj ďalší prípad, ktorý priamo súvisí so správou užívateľov a tým je:

- *Odstrániť užívateľa* - administrátor môže svojvoľne odstrániť akékoľvek užívateľa zo systému, v prípade, že porušuje niektoré z pravidiel systému. Pri odstránení užívateľa, sú aj všetky jeho minulé či budúce rezervácie vymazané zo systému.
- *Odstrániť užívateľa* - administrátor môže aktivovať účet novoregistrovaného užívateľa, pokiaľ mu z nejakého dôvodu neprišiel aktivačný email.

- **Schváliť vytvorenie klubu** - po vytvorení klubu je potrebné ho schváliť, aby tie kluby zbytočne nezahľcovali systém a aby tie kluby mali nejaký zmysel.

Súčasťou tohto prípadu použitia je aj ďalší prípad, ktorý priamo súvisí so správou klubov a tým je:

- *Pridať vedúceho klubu* - okrem schválenia klubu je potrebné schváliť aj jeho prvého vedúceho a toto je práve v kompetenciách len administrátora.

- **Pridať športovisko** - na športovisko je priamo ukotvená rezervácia, čo znamená, že bez športoviska by žiadna rezervácia nemohla byť. Avšak len administrátor ako užívateľ systému môže jednotlivé športoviská pridať do systému.

Súčasťou tohto prípadu použitia sú aj ďalšie prípady, ktoré priamo súvisia so správou športovísk a tými sú:

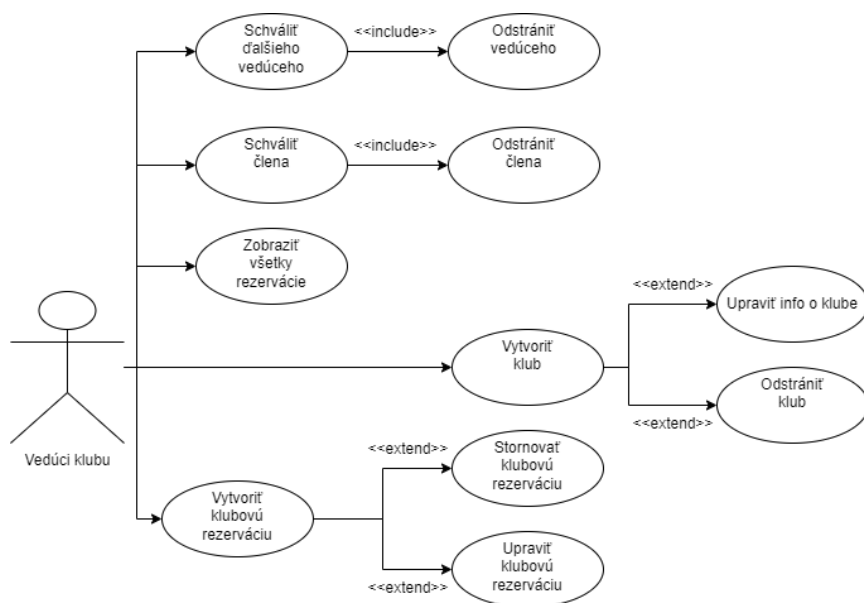
- *Odstrániť športovisko* - všetky športoviská je možné samozrejme aj odstrániť, táto možnosť pre bezpečnosť a bezproblémový chod systému prináleží len administrátorovi. Keďže každá rezervácia, je priamo vedená k danému športovisku, tak pri odstránení toho športoviska automaticky zaníka, na základe toho ako je daný systém implementovaný v databázovej časti.
- *Upraviť športovisko* - úprava športoviska, ako aj všetko ostatné čo sa týka športoviska, prináleží len administrátorovi, ktorý môže v danom športovisku zmeniť čokoľvek mu je ponúknuté. V prípade, že by k takejto zmene nastalo, tak všetci užívatelia, ktorí majú na danom športovisku nejakú budúcu rezerváciu, dostanú od systému mail, že sa niečo pri ich rezervácii mohlo zmeniť.

Ďalšími samostatnými prípadmi použitia sú:

- **Zmena role užívateľa** - zmena role súvisí hlavne s vytváraním klubov, kedy aj administrátor môže klub vytvoriť, pretože všetky prípady použitia aj ostatných užívateľov dedí. Tak automaticky pri vytvorení klubu, určí jedného z užívateľov za jeho vedúceho a tým zmení jeho rolu v rámci systému.
- **Stornovať užívateľovu rezerváciu** - v prípade, že dôjde k neočakávanej zmene, tak administrátor môže stornovať akúkoľvek užívateľskú rezerváciu.

2.1.2 Diagram prípadov použitia popisujúci chovanie vedúceho klubu

Vedúci klubu môže v systéme vykonávať nasledujúce akcie a dedí všetky akcie aj od hierarchicky podriadených aktérov systému:



Obr. 2.3: Diagram prípadov použitia vedúceho klubu

- **Schváliť člena** - každý užívateľ, ktorý sa chce pridať do tímu, musí byť schválený terajším vedúcim klubu aby sa tým členom mohol stať

Súčasťou tohto prípadu užitia je aj ďalší prípad, ktorý priamo súvisí so správou užívateľov a tým je:

- *Odstrániť člena* - v prípade, že člen je napr. neaktívny, alebo porušil nejaké pravidlá môže byť z klubu odstránený

- **Schváliť ďalšieho vedúceho** - klub môže mať viacero vedúcich a preto je potrebné aby existujúci členovia klubu po podaní žiadosti o to aby mohli byť ďalším vedúcim mohli byť už existujúcimi vedúcimi schválený na túto pozíciu.

Súčasťou tohto prípadu užitia je aj ďalší prípad, ktorý priamo súvisí so správou užívateľov a tým je:

- *Odstrániť vedúceho* - v prípade, že vedúci je napr. neaktívny, alebo porušil nejaké pravidlá môže byť z klubu odstránený

- **Zobraziť všetky rezervácie** - vedúci klubu vidí zoznam všetkých rezervácií nielen v rozvrhu na tento týždeň, ale vie si otvoriť aj zoznam úplne všetkých rezervácií v systéme, aj minulé aj budúce.

- **Vytvoriť klub** - tak ako klub môže mať viacero vedúcich, tak takisto môže aj už vedúci nejakého klubu, vytvoriť ďalší klub a stať sa jeho prvým vedúcim.

Súčasťou tohto prípadu užitia sú aj ďalšie prípady, ktoré priamo súvisia so správou klubov a tými sú:

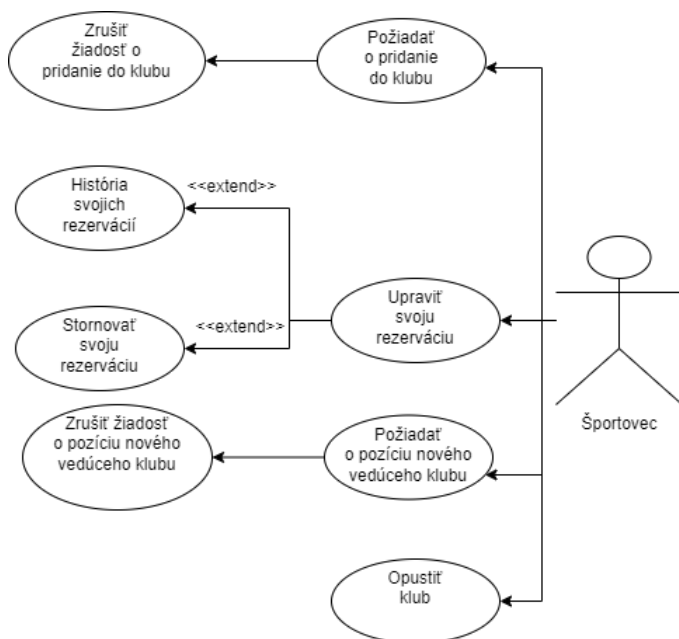
- *Odstrániť klub* - pokiaľ sa vedúci klubu rozhodne klub odstrániť, nie je na to potrebné žiadne schválenie, spolu s týmto sa však rozošle všetkým členom klubu mail o tom, že klub sa zrušil a už nie sú jeho súčasťou.
- *Upraviť info o klube* - vedúci môže upraviť info o klube, pokiaľ sa naskytne nejaká nová situácia, po ktorej to je potrebné.

Pri týchto prípadoch užitia môžu nastať 2 situácie, ktorými sú:

- Neregistrovaný užívateľ vytvorí klub C.5
- Registrovaný užívateľ vytvorí klub C.4
- **Vytvoriť klubovú rezerváciu** - je to vytvorenie rezervácie, ako každé iné avšak, je po vytvorení takejto rezervácie odoslaný email o rezervácii všetkým členom klubu a vidia túto rezerváciu medzi svojimi rezerváciami.
 - *Stornovať klubovú rezerváciu* - v prípade, že je klubová rezervácia stornovaná, systém opäť odošle email všetkým členom.
 - *Upraviť klubovú rezerváciu* - v prípade, že je klubová rezervácia upravená, systém opäť odošle email všetkým členom.

2.1.3 Diagram prípadov použitia popisujúci chovanie športovca

Športovec môže v systéme vykonávať nasledujúce akcie a dedí všetky akcie aj od hierarchicky podriadených aktérov systému:



Obr. 2.4: Diagram prípadov použitia športovca

- **Požiadat o pridanie do klubu** - ak sa chce športovec stať členom nejakého klubu, musí požiadať o pridanie, ak je schválený stane sa jeho členom a vidí v systéme klubové rezervácie medzi svojimi.

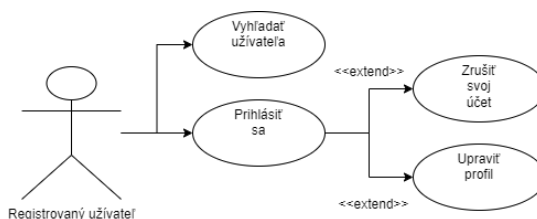
- **Zrušiť žiadosť o pridanie do klubu** - v prípade, že športovec odoslal žiadosť o pridanie do klubu, môže ju aj zrušiť
- **Upraviť svoju rezerváciu** - ak si užívateľ vytvoril rezerváciu, stal sa z neho automaticky v očiach systému športovec, čo je znázornené v pridanom sekvenčnom diagrame C.4, tak túto rezerváciu vie upraviť podľa svojej potreby, následne mu o tejto zmene príde zo systému potvrdzovací email.

Súčasťou tohto prípadu použitia sú aj ďalšie prípady, ktoré priamo súvisia so správou svojich rezervácií a tými sú:

- *História svojich rezervácií* - užívateľ si môže pozrieť všetky svoje rezervácie ako zoznam či už minulých alebo aj budúcich, v prípade, že nechce hľadať v rozvrhu.
- *Stornovať svoju rezerváciu* - ak sa užívateľ rozhodne z akýchkoľvek dôvodov, že mu rezervácia nevyhovuje tak ju môže stornovať a vymaže sa zo systému, v takomto prípade mu opäť príde potvrdzovací email o stornovaní.
- **Požiadat' o pozíciu nového vedúceho klubu** - ak sa chce člen klubu stať vedúcim svojho klubu, musí požiadať o miesto vedúceho, ak je schválený stane sa jeho vedúcim a vidí v systéme klubové rezervácie medzi svojimi.
- **Zrušiť žiadosť o pridanie do klubu** - v prípade, že člen klubu odoslal žiadosť o pridanie za vedúceho klubu, môže ju aj zrušiť
- **Opustiť klub** - člen klubu sa môže rozhodnúť ho aj opustiť

2.1.4 Diagram prípadov použitia popisujúci chovanie registrovaného užívateľa

Registrovaný užívateľ môže v systéme vykonávať nasledujúce akcie a dedí všetky akcie aj od hierarchicky podriadených aktérov systému:



Obr. 2.5: Diagram prípadov použitia registrovaného užívateľa

- **Prihlásiť sa** - ak je užívateľ už zaregistrovaný ale nie je prihlásený, tak jedným s prípadov použitia je práve táto možnosť, kedy pri spustení stránky sa môže prihlásiť ak už je zaregistrovaný.

Súčasťou tohto prípadu použitia sú aj ďalšie prípady, ktoré priamo súvisia s týmto prípadom a tým sú:

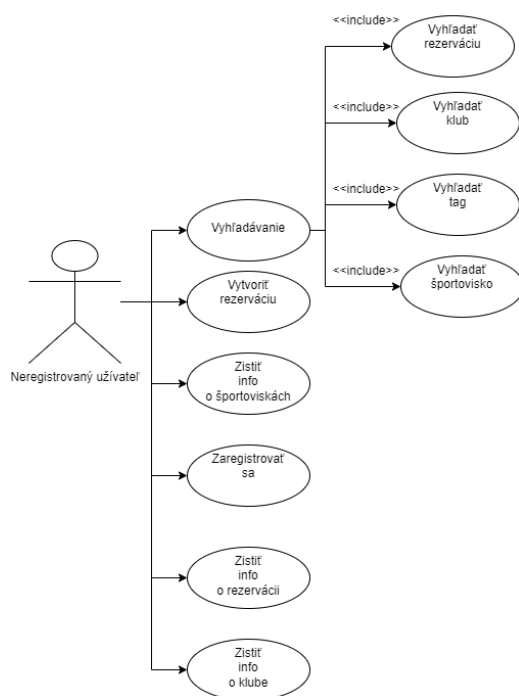
- **Upraviť profil** - každý užívateľ si vie upraviť profil a informácie z daného profilu sú uložené v databáze, pri vytvorení rezervácie sú jeho osobné údaje už predpísané vo formulári.

– *Zrušiť svoj účet* - užívateľ môže svoj účet kedykoľvek zrušiť a jeho profil a osobné údaje sa vymažú zo systému, takisto sa vymažú aj všetky jeho rezervácie.

- **Vyhľadať užívateľa** - ako registrovaný užívateľ mám možnosť vyhľadať aj užívateľa keďže aj on sám súhlasil s podmienkami systému aj v oblasti GDPR pri registrácii a má teda možnosť zistiť aký ďalší užívatelia v systéme sú, tento prístup je deklarovaný v tabuľkách 2.1 a 2.2.

2.1.5 Diagram prípadov použitia popisujúci chovanie neregistrovaného užívateľa

Neregistrovaný užívateľ je v hierarchii systému najnižšie má teda najmenšie práva a už nemá od koho dediť tak jeho a tak môže v systéme vykonávať len nasledujúce akcie:



Obr. 2.6: Diagram prípadov použitia neregistrovaného užívateľa

- **Vytvoriť rezerváciu** - vytvorenie rezervácie je základný kameň celého systému a môže ho robiť každý aktér, pokiaľ ale užívateľ je neregistrovaný v systéme vytvorí sa anonymná rezervácia, ktorú môže zrušiť len administrátor.
- **Zistiť info o športoviskách** - pri kliknutí na tlačidlo športoviska sa užívateľovi otvorí okno s infom o danom športovisku
- **Zistiť info o rezervácii** - pri kliknutí na rezerváciu v rozvrhu sa užívateľovi otvorí okno s infom o danej rezervácii
- **Zistiť info o klube** - pri kliknutí na klub sa užívateľovi otvorí okno s infom o danom klube, bez akýchkoľvek osobných údajov o členoch alebo vedúcich.

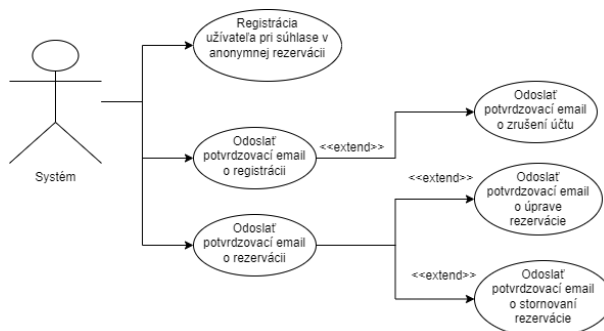
- **Vyhľadávanie** - vyhľadávanie je súčasťou systému pre ľahšiu a rýchlejšiu navigáciu, je vytvorené formou **checkboxov** a takisto je možné vyhľadať celé slovo alebo len časť slova.

Súčasťou tohto prípadu použitia sú aj ďalšie prípady, ktoré priamo súvisia s vyhľadávaním a tými sú:

- *Vyhľadať rezerváciu* - pri vyhľadní rezervácii sa neregistrovanému užívateľovi ukáže zoznam rezervácií o ktorých si po kliknutí na jednotlivú rezerváciu môže zistiť info o nej.
- *Vyhľadať klub* - pri vyhľadní klubu sa neregistrovanému užívateľovi ukáže zoznam klubov o ktorých si po kliknutí na jednotlivý klub môže zistiť info o ňom.
- *Vyhľadať tag* - pri vyhľadní tagu sa neregistrovanému užívateľovi ukáže zoznam tagov o ktorých si po kliknutí na jednotlivý tag môže zistiť info o ňom.
- *Vyhľadať športovisko* - pri vyhľadní športoviska sa neregistrovanému užívateľovi ukáže zoznam športovísk o ktorých si po kliknutí na jednotlivé športovisko môže zistiť info o nich.
- **Zaregistrovať sa** - ak má neregistrovaný užívateľ záujem sa stať členom nejakého klubu, alebo si spravovať svoje rezervácie sám, tak sa musí do systému zaregistrovať, musí poskytnúť svoje osobné údaje a súhlasiť s ich spracovaním na základe GDPR zákona, takisto musí súhlasiť s podmienkami a pravidlami systému.

2.1.6 Diagram prípadov použitia popisujúci chovanie systému ako pasívneho aktéra

Systém je pasívny aktér systému, ktorého prípady použitia sú tieto:



Obr. 2.7: Diagram prípadov použitia systému ako pasívneho aktéra

- **Registrácia užívateľa pri súhlase v anonymnej rezervácii** - pri vytvorení anonymnej rezervácii sa systém opýta užívateľa či si chce vytvoriť účet v systéme, ak áno tak sa uložia všetky osobné údaje do systému, pri takejto rezervácii ale užívateľ nemá zvolené žiadne užívateľské meno a je nutné aby si ho zmenil
- **Odoslať potvrdzovací email o registrácii** - systém pri každej registrácii odošle na poskytnutú emailovú adresu potvrdzovací email o registrácii a potvrdení emailovej adresy

Súčasťou tohto prípadu využitia sú aj ďalšie prípady, ktoré priamo súvisia s vyhľadávaním a tými sú:

- *Odoslať potvrdzovací email o zrušení účtu* - systém aj pri zrušení užívateľského účtu odošle na emailovú adresu odstráneného užívateľa email o odstránení jeho účtu zo systému
- **Odoslať potvrdzovací email o rezervácii** - pri každej rezervácii systém odošle potvrdzovací email na poskytnutú emailovú adresu užívateľa, nezáleží o akého užívateľa sa jedná, súčasťou emailu je aj iCal pozvánka na konkrétnu rezerváciu, ktorá je prílohou emailu a užívateľ si ju môže pridať do vlastného kalendára.

Súčasťou tohto prípadu využitia sú aj ďalšie prípady, ktoré priamo súvisia s vyhľadávaním a tými sú:

- *Odoslať potvrdzovací email o úprave rezervácie* - pri každej úprave rezervácie systém odošle potvrdzovací email o jej úprave na poskytnutú emailovú adresu užívateľa, nezáleží o akého užívateľa sa jedná, súčasťou emailu je aj iCal pozvánka na konkrétnu rezerváciu, ktorá je prílohou emailu a užívateľ si ju môže pridať do vlastného kalendára.
- *Odoslať potvrdzovací email o stornovaní rezervácie* - pri každom stornovaní rezervácii systém odošle potvrdzovací email o jej stornovaní na poskytnutú emailovú adresu užívateľa, nezáleží o akého užívateľa sa jedná.

2.2 Entity a atribúty a ich zakotvenie v systéme

Súčasťou systému sú takisto entity, ktoré sa uchovávajú v databáze. Všetky entity a ich práva sú definované v tabuľke práv entít [2.2](#).

V nasledujúcich sekciách si jednotlivé entity a atribút predstavíme bližšie a všetko toto je zahrnuté takisto v ER diagrame [3.3](#), ktorý bližšie popisuje vzťahy, jednotlivé atribúty či vzťahy atribútov medzi entitami.

O tom aké informácie, vo forme atribútov, sú uložené v databáze o jednotlivých entitách je bližšie popísané v sekcii o **modeloch** [4.2](#)

2.2.1 Užívatelia

Užívatelia v systéme tvoria základný kameň, pretože sú na nich naviazané aj ďalšie entity systému, bez akéhokoľvek užívateľa totiž nemôže vzniknúť žiadna iná entita. Entity ako **Rezervácia** [2.2.4](#) a **Klub** [2.2.3](#) sa v systéme bez akéhokoľvek užívateľa do aktívneho systému nedostanú.

Takisto je to aj so **Športoviskom** [2.2.2](#), ktoré dokáže vytvoriť a spravovať len administrátor a teda sa samovoľne v systéme vyskytnúť nemôže.

Sú rozdelení na 2 skupiny a dohromady dávajú 5 aktérov celého systému, ako je ukázané v kompletnom diagrame prípadov použitia [C.6](#), tabuľke práv užívateľov [2.1](#) alebo hierarchii aktérov [2.1](#). Jedná sa teda konkrétne o tieto skupiny:

- **Registrovaní užívatelia** - predpokladá sa, že budú tvoriť gró aktérov systému aby bol systém účinný a naozaj prospešný pre zákazníka.

Všetky prípady využitia registrovaných užívateľov je vidieť v diagrame [C.7](#)

Delia sa na:

- *Administrátor* - ako vidíme z diagramu C.6, tabuľky 2.1 a hierarchii 2.1, administrátor je užívateľ systému s neobmedzenými právami a môže robiť všetky dostupné operácie v systéme.

Je zodpovedný za celý chod systému. Je to úplne prvý užívateľ celého systému a pre správne fungovanie systému, nie je možné aby bol odstránený.

Konkrétne prípady použitia len pre tento typ **užívateľa** sú uvedené v diagrame 2.4.

- *Vedúci klubu* - vedúcim sa môže stať akykoľvek **registrovaný užívateľ** 2.2.1, ktorý buď sám **klub** vytvorí a následne ho *administrátor* schváli, ako je to naznačené v diagrame C.4 alebo takisto akýkoľvek člen už existujúceho **klubu**, pri podaní žiadosti aby sa stal novým vedúcim.

V prípade, že by sa chcel stať vedúcim klubu **neregistrovaný užívateľ**, je potrebné aby si buď sám vytvoril účet, alebo aby na základe vlastnej komunikácie, mimo systému, s *administrátorom* ho požiadal aby mu vytvoril účet, **klub** a spravil z neho vedúceho toho klubu ako je to naznačené v tomto diagrame C.5.

Všetky práva tohto typu **užívateľa** sú opäť vidieť v tabuľke 2.1, ktorá sa riadi hierarchiou 2.1.

Konkrétne prípady použitia len pre tento typ **užívateľa** sú uvedené v diagrame 2.3.

- *Športovec* - športovcom v očiach systému je každý **registrovaný užívateľ** pokiaľ je členom nejakého **Klubu** alebo vytvoril v systéme **rezerváciu** na nejaké **športovisko**.

Všetky práva tohto typu **užívateľa** sú opäť vidieť v tabuľke 2.1, ktorá sa riadi hierarchiou 2.1.

Konkrétne prípady použitia len pre tento typ **užívateľa** sú uvedené v diagrame 2.4.

- **Neregistrovaní užívatelia** - ako je spomínané v **análýze požiadaviek** 2, súčasný systém nepovoľuje interakciu so systémom **neregistrovanému užívateľovi**, čo je z hľadiska využiteľnosti systému značné obmedzenie. Na druhú stranu systém, ktorým sa zaoberá táto bakalárska práca umožňuje aktívne využívanie pre všetky typy **užívateľov** s malými odlišnosťami a právami, ktoré su takisto definované v tabuľke 2.1 a diagrame 2.6

Čo všetko **užívateľ** ako entita môže v systéme robiť je popísané v tabuľke 2.2.

2.2.2 Športovisko

športovisko takisto patrí k základným kameňom systému, keďže každá **rezervácia** je viazaná priamo na **užívateľa**, ktorý ju vytvoril a na **športovisko**, na ktorom sa dané podujatie bude konať.

Športovisko samotné môže do systému pridať len administrátor, ako to ukazuje aj diagram 2.2 aj tabuľka 2.1.

Čo všetko **športovisko** ako entita môže v systéme robiť je popísané v tabuľke 2.2.

2.2.3 Klub

Klub nemôže byť nikdy samostatne vytvorený bez toho aby k nemu patrilo nejaký **užívateľ**, ktorý je teda *vedúcim klubu*.

Má teda svojich vedúcich aj členov, ktorých je možno schváliť na pridanie do klubu či na pozíciu *vedúceho*, alebo ich aj odstrániť z klubu.

Klub ako entita môže mať tieto 3 životné cykly:

- Admin vytvorí klub [C.1](#)
- Neregistrovaný užívateľ chce vytvoriť klub [C.2](#)
- Registrovaný užívateľ vytvorí klub [C.3](#)

A súvisia s nimi aj tieto sekvenčné diagramy, naznačujúce priebeh vytvorenia:

- Vytvorenie klubu registrovaným užívateľom [C.4](#)
- Vytvorenie klubu neregistrovaným užívateľom [C.5](#)

Čo všetko **klub** ako entita môže v systéme robiť je popísané v tabuľke [2.2](#).

2.2.4 Rezervácia

Rezervácia je slabá entita pretože sa viaže v systéme na **užívateľa** [2.2.1](#) a **športovisko** [2.2.2](#) a teda uzatvára hlavný trojuholník celého systému.

Vzhľadom na to, že **rezerváciu** môže podľa tabuľky [2.1](#) vytvoriť každý, teda aj neregistrovaný užívateľ, naberá systém na možnosti nových užívateľov a pôsobí prívetivejšie aj pre užívateľov, ktorí sa registrovať nechceli a nemohli teda existujúci systém využívať [2](#).

Čo všetko **rezervácia** ako entita môže v systéme robiť je popísané v tabuľke [2.2](#).

2.2.5 Tag

Pri tagu sa nejedná o samostatnú entitu, ale o atribút daných entít a dopomáha k prenositeľnosti systému a k personalizácii ktorejkoľvek entity. Tag je atribút, ktorý sa môže zmeniť pri takmer všetkých entitách až na **rezerváciu**, tam je každý **tag** generovaný automaticky na základe ostatných tagov. Každý tag má na začiatku špeciálny znak **#**, ktorým sa vyznačuje práve tento atribút.

Kapitola 3

Návrh systému

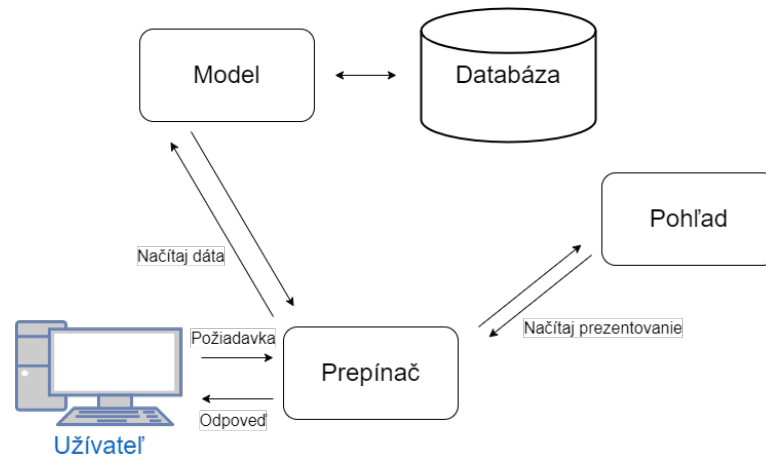
Nasledujúca kapitola konkrétne popisuje návrh systému podľa zistených požiadaviek z predošlej kapitoly. Systém je určený pre konkrétneho zákazníka s dôrazom na prenositeľnosť a udržateľnosť aj pre iných potenciálnych zákazníkov.

3.1 MVC

MVC (Model-view-controller) je tradičný softvérový architektonický vzor ktorý sa používa pri vytváraní webových alebo klasických aplikácií. Systém rozdeľuje na 3 hlavné časti, ktoré medzi sebou istým spôsobom komunikujú a každá má svoju úlohu. Typicky sú to:

- Model - Reprezentuje dáta a biznis logiku aplikácie. Zapuzdruje dáta a poskytuje metódy na manipuláciu s nimi, ako pridanie, mazanie, vyhľadávanie a editovanie.
- View - Zobrazuje dáta používateľovi a interaguje s užívateľským rozhraním. Získava dáta z modelu a posiela užívateľský vstup kontroléru.
- Controller - Pôsobí ako prostredník medzi modelom a zobrazením. Získava užívateľský vstup zo zobrazenia, spracováva ho (prípadne modifikuje model) a zodpovedajúco aktualizuje zobrazenie.

Takto navrhnutý systém je potom prehľadnejší, flexibilnejší, udržateľnejší a ľahšie sa testuje. Každá časť MVC sa však môže prispôbovať a podľa potreby zákazníka meniť. [19]



Obr. 3.1: Diagram MVC

Pri hľadaní informácií o návrhovom modeli MVC som narazil na istú odchylku pri mnou používanej technológii, zdroje uvádzajú, že DJANGO používa upravenú verziu tohoto návrhového modelu MVT. [1]

Model-Template-View (MVT) je softvérový architektonický vzor, ktorý sa často používa v webovom vývoji. Rozdeľuje webovú aplikáciu do troch hlavných komponentov:

- Model - pre Model platí to isté ako pri návrhovom vzore MVC.

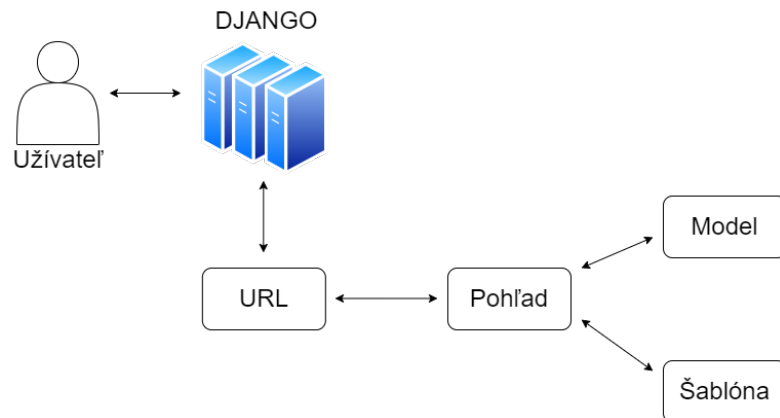
Reprezentuje dáta a biznis logiku aplikácie. Tento komponent sa stará o manipuláciu s dátami a interakciu s databázou. Model predstavuje jednotnú vrstvu, ktorá izoluje dáta od zobrazenia a šablón.

- Template (Šablóna) - táto časť návrhového vzoru nahrádza View.

Šablóny sú preddefinované HTML súbory, ktoré definujú štruktúru a vzhľad užívateľského rozhrania. Šablóny môžu obsahovať špeciálne značky alebo syntax, ktoré sa následne nahradia konkrétnymi hodnotami z modelu.

- View - táto časť zasa nahrádza Controller.

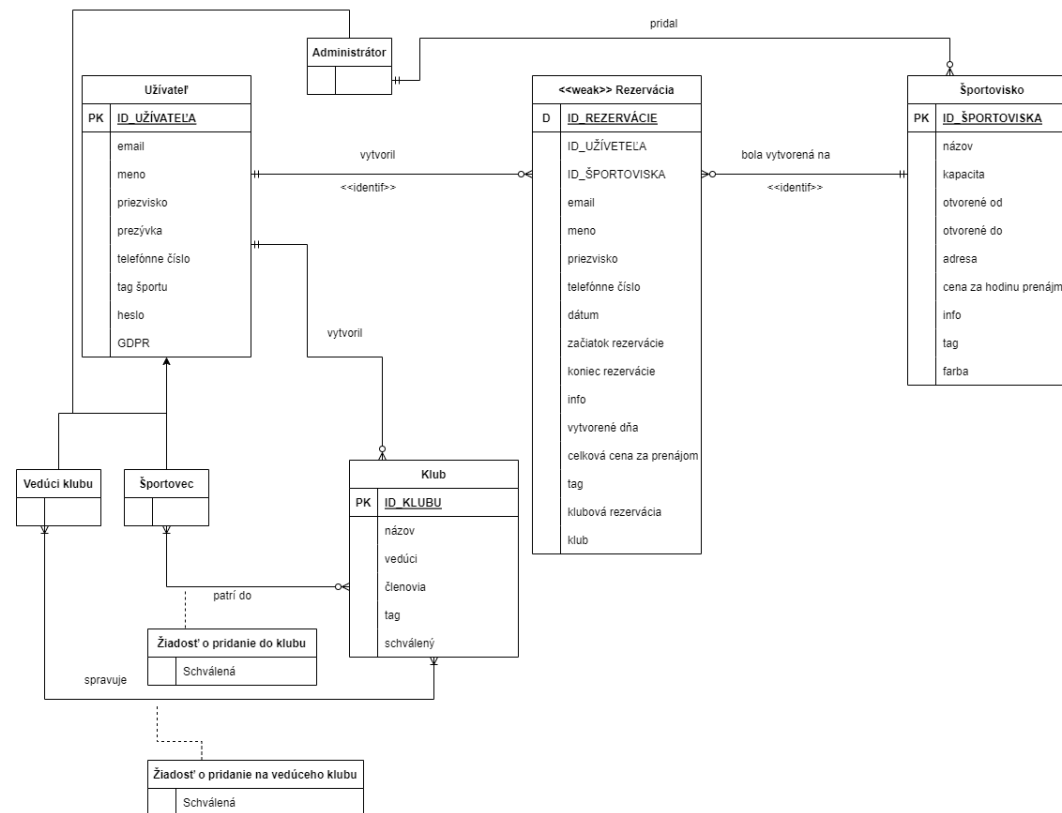
Zobrazenia sú vrstvy, ktoré interagujú s užívateľom a zobrazujú dáta na základe šablón. Zobrazenia sú zodpovedné za získanie požadovaných dát z modelu a ich prezentáciu pomocou príslušnej šablóny.



Obr. 3.2: Diagram MVT

3.2 ER-diagram systému

Entity Relationship Diagram alebo diagram vzťahov medzi entitami je diagram, ktorý sa používa k popisovaniu objektov systému - takzvaných entít, vzťahov medzi nimi, popis atribútov ktoré jednotlivé entity obsahujú alebo atribútových vzťahov systému [4]. Na základe analýzy požiadavok systému z predošlej kapitoly, bol vytvorený nasledujúci ER diagram.



Obr. 3.3: ER diagram popisujúci programovaný systém

Všetky dáta, ktoré sú ukladané do databázy o jednotlivých entitách sú popísané v sekcii 4.2.

3.3 Užívateľ

Ako je už spomínané vyššie v sekcii 2.2.1, tak v systéme sa užívateľ môže zastupovať až 3 aktérmi, ktorý sú generalizovaný do 1 entity ako **užívateľ**. Akýkoľvek **užívateľ** môže vytvoriť ľubovoľný počet rezervácií, teda kardinalita vzťahu je 0 až N pri rezervácii. Len **registrovaný užívateľ** môže vytvoriť **klub**, môže ich však opäť vytvoriť ľubovoľný počet, kardinalita vzťahu teda znova predstavuje 0 až N pri klube. Súčasťou systému je aj **neregistrovaný užívateľ**, ten ale nie je ukladaný do databázy, takže sa v ER diagrame nenachádza.

- **Vedúci klubu** - je jedným z priamych aktérov systému, priamo spravuje klub a je určený pri vytvorení klubu. Tento typ užívateľa môže byť vedúcim pre viacero klubov, minimálne ale aspoň pre jeden, teda platí u neho kardinalita 1 až N pri klube.
- **Športovec** - je v ponímaní systému, každý užívateľ, ktorý si vytvoril nejakú **rezerváciu**, alebo patrí do nejakého **klubu**. Stane sa ním teda každý užívateľ automaticky, ak splní jednu z dvoch týchto podmienok. Tento typ užívateľa môže patriť do viacerých klubov, môže sa však stať, že nemusí patriť do žiadneho, vtedy sa teda pýta kardinalita 0 až N pri klube a svoj status *športovca*, získal a drží si ho len na základe vytvorenia **rezervácie**.
- **Administrátor** - je vedúci celého systému, má na starosti všetok chod systému a disponuje všetkými právami, tak ako je to ukázané v tabuľke 2.1. Ako jediný môže pridávať do systému **športoviská** a teda kardinalita je 0 až N.

3.4 Klub

Ako je už opäť spomínané v predchádzajúcej sekcii 2.2.3, klub sám o sebe nemôže existovať a práve preto je v systéme navrhnutý tak aby každý klub mohol mať viacero *vedúcich*, avšak vytvoriť ho môže len práve jeden **užívateľ**. Preto je kardinalita práve 1 pri vzťahu k užívateľovi.

Kardinalita, určujúca počet *vedúcich klubu*, spravujúcich klub je pri vedúcom tiež 1 až N, pretože klub bez vedúceho nemôže existovať, ale zároveň, môže mať viac ako jedného vedúceho, pokiaľ ich, už existujúci vedúci, schválili.

To isté platí aj pri počte členov, pokiaľ je klub vytvorený, tak má svojho vedúceho a to je teda zároveň aj prvý člen, avšak v databáze to vidieť nie je, kvôli redundancii dát, kvôli tomu, že vedúci má isté práva, ktorými *športovec*, alebo teda len rádový člen klubu, nedisponuje. Kardinalita je teda 1 až N, pokiaľ už existujúci *vedúci klubu*, schválil jeho žiadosť o členstvo.

- **Žiadosť o pridanie do klubu** - zápis v databáze, ktorý určuje, či bol daný **užívateľ**, po odoslaní žiadosti o pridanie do klubu, schválený ako nový člen klubu.
- **Žiadosť o pridanie na vedúceho klubu** - zápis v databáze, ktorý určuje, či bol daný už existujúci člen klubu, po odoslaní žiadosti o vymenovanie za ďalšieho *vedúceho klubu*, schválený ako nový *vedúci klubu*.

3.5 Rezervácia

V predchádzajúcej sekcii 2.2.4 je spomenuté, že **rezervácia** nemôže existovať bez toho aby bola vytvorená istým **užívateľom**, môže sa jednať o registrovaného aj neregistrovaného užívateľa, a bola naviazaná, alebo vytvorená na konkrétne **športovisko**.

Preto je kardinalita pri **užívateľovi** naviazanom na **rezerváciu** 1, pretože rezerváciu vytvoril práve tento 1 **užívateľ**.

Takisto kardinalita pri **športovisku**, na ktoré bola rezervácia vytvorená, je 1 pretože jedna **rezervácia** nemôže byť vytvorená na viacero **športovísk**.

Avšak v systéme môže nastať aj situácia, že je vytvorená *klubová rezervácia*. Tento prípad nemá samostatnú entitu, skôr sa nastavuje príznak o tom či je rezervácia klubová alebo nie, ak áno musí sa k nej nastaviť aj príslušný **klub**.

- **Klubová rezervácia** - nastane v prípade, že **užívateľ**, ktorý vytvoril rezerváciu je zároveň *vedúcim* nejakého **klubu** a vo formulári na vytvorenie **rezervácie**, zaškrtnie políčko, že sa jedná o *klubovú rezerváciu* a zvolí príslušný **klub**, ktorého je *vedúcim*.

3.6 Športovisko

V predchádzajúcich sekciami 2.2.2 a 3.3 je spomenuté, že vytvorenie, alebo teda pridanie **športoviska**, do systému je výlučne prácou *administrátora*.

Preto je kardinalita pri *administrátorovi*, ktorý ho pridal do systému 1, lebo existuje len 1 *administrátor*.

Avšak pri kardinalite vzťahu s **rezerváciou** je to inak. Na jedno a to isté **športovisko** môže byť v systéme vytvorených 0 až N **rezervácií**. Pri kardinalite N sa samozrejme myslí pre iný čas a dátum rezervácie, v ten istý čas môže byť na to jedno konkrétne športovisko vytvorená len jedna rezervácia. Na druhý deň v ten istý čas môže byť rezervácia vytvorená opäť, alebo aj kedykoľvek inokedy.

Administrátor potrebuje pri vytvorení vedieť hlavne kde sa nachádza, teda jeho adresu, akú má kapacitu, koľko ľudí sa naň zmestí, či už priamo ako športovci, alebo ako diváci, môže sa totižto stať, že športovisko sa bude chcieť použiť na iné účely ako mu dovoľuje jeho kapacita. Vždy je preto dobré pri rezervácii napísať do informácie v rezervačnom formulári odhadovaný počet osôb, ktoré budú na športovisku. Bez názvu a tagu športoviska sa žiadne nedokáže vytvoriť a je preto potrebné aby malo každé športovisko jedinečný názov a tag, toto je jednou z podmienok vytvorenia. Potrebná je aj otváracia doba, je teda potrebné nastaviť čas odkedy dokedy je športovisko možné využívať. Takisto pre správne fungovanie systému je dôležité aby bola nastavená cena prenájmu za hodinu používania, ktorá pri vytvorení rezervácie slúži na výpočet celkovej ceny za prenájom.

Všetky tieto informácie majú hlavne informatívny účel pre užívateľov systému, ktorý napríklad nevedia kde sa športovisko nachádza, odkedy dokedy je otvorené, koľko ľudí môže spolu s ním prísť na športovisko, koľko budú za svoju rezerváciu platiť.

Kapitola 4

Implementačné detaily rezervačného systému

Systém beží na architektúre klient-server, užívatelia zasielajú požiadavky na server, ten ich vyhodnotí a odošle odpoveď. V úvode kapitoly sú popísané konkrétne technológie, ktoré boli použité pri implementácii systému, každá z použitých technológií je samostatne aj popísaná.

Implementácia celého systému je objektovo-orientovaná, takmer každá súčasť systému je vytvorená ako samostatná trieda, na čo je použité kľúčové slovo programovacieho jazyka `class`, obsahujúca svoje metódy, hlavne metódy `get` a `post`, ktoré patria medzi 4 základné HTTP metódy [27].

Celá aplikácia je rozdelená na jednotlivé súčasti, takzvané aplikácie, ako sa používa v DJANGO [18], kde každá aplikácia má svoju samostatnú štruktúru, ktorá je naznačená v E.

Takmer každá aplikácia predstavuje v systéme svoju entitu 2.2 okrem aplikácie **home**, ktorá predstavuje hlavný pohľad celej aplikácie, takzvaný rovrh na vytvorenie rezervácii.

Takisto sú v kapitole popísané základné implementované zložky systému, ktorými sú: databáza, server a užívateľské rozhranie.

4.1 Technológie použité v projekte

V danej sekcii vás v skratke a rýchlo prevediem použitými technológiami, nástrojmi a programovacími jazykmi, ktoré boli pri implementovaní mojej bakalárskej práce použité. Každú z týchto technológií v skratke predstavím a popíšem jej využitie v systéme. Sekcia je rozdelená do 4 podsekcí, z ktorých 3 predstavujú použité technológie pre danú implementovanú zložku systému.

Prvá sa venuje implementovaniu **databázy** 4.1.1, druhá implementácii **serveru** 4.1.2 a tretia **užívateľskému rozhraniu** 4.1.3. Posledná štvrtá sa venuje prenositeľnosti daného systému pomocou kontajnerizácie cez Docker 4.1.4

4.1.1 Databáza

Databáza je organizovaná zbierka štruktúrovaných informácií alebo údajov, zvyčajne uložených v elektronickej podobe v počítačovom systéme. Databázu zvyčajne riadi systém správy databáz (DBMS). Dáta a DBMS spolu s aplikáciami, ktoré sú s nimi spojené, sa spoločne označujú ako databázový systém, často obmedzený len na databázu.

Dáta v rámci najbežnejších typov databáz, ktoré sú dnes v prevádzke, sú zvyčajne modelované v riadkoch a stĺpcoch v sérii tabuliek, aby boli spracovanie a dopyty na dáta efektívne. Prístup k dátam možno potom jednoducho získať, spravovať, modifikovať, aktualizovať, kontrolovať a organizovať. Väčšina databáz používa na zápis a vykonávanie dopytov na dáta štruktúrovaný dopytovací jazyk (SQL). [24]

SQL

SQL je programovací jazyk používaný takmer všetkými relačnými databázami na vykonávanie dopytov, spracovanie a definovanie dát a na poskytovanie riadenia prístupu. Jazyk SQL bol prvýkrát vyvinutý v IBM v roku 1970, pričom spoločnosť Oracle bola hlavným prispievateľom. To viedlo k implementácii štandardu SQL ANSI, SQL podnietil množstvo rozšírení od spoločností ako IBM, Oracle a Microsoft. Hoci sa SQL v súčasnosti stále používa v širokej miere, začínajú sa objavovať nové programovacie jazyky. [24]

SQLite

SQLite je jeden z mnohých RDBMS systémov. Rozhodol som sa preňho, pretože je bezplatný, nevyžaduje žiadnu inštaláciu alebo konfiguráciu a nemusí bežať na samostatnom serveri. Je rýchly, kompatibilný s rôznymi platformami a veľmi kompaktný. [30].

4.1.2 Server

Ako je spomenuté na začiatku kapitoly 4, systém funguje na architektúre klient-server a táto sekcia je zameraná na serverovú časť, ktorá pristupuje do databázy a má za úlohu spracovávať a vyhodnocovať požiadavky klientov.

Python

Python patrí už dlhé roky k najpoužívanejším programovacím jazykom, v priebehu posledných rokov sa drží v top 5 najpoužívanejších na svete podľa [25] aj podľa [20]. Bol vybraný aj na implementáciu tejto bakalárskej práce, hlavne kvôli osobnej skúsenosti a osobným preferenciám.

DJANGO

Serverová časť implementácia, nie je tak úplne čistý Python, server totižto beží na frameworku DJANGO [18], ktorý sa v posledných rokoch dostáva do popredia v používaní pri vytváraní webových aplikácií [22].

4.1.3 Grafické užívateľské rozhranie

Nasledujúca sekcia popisuje technológie, ktoré boli využité pri implementácii grafického užívateľského rozhrania.

HTML

HTML, alias HyperText Markup Language, je jazyk používaný na vytváranie webových stránok. Obsahuje značky, ktoré určujú rôzne prvky a tým vytvárajú základnú štruktúru webu. Vo väčšine prípadov sa HTML kombinuje s CSS pre vzhľad a JavaScriptom pre funkcionálnosť. [13]

CSS

CSS, skratka pre Cascading Style Sheets, je jazyk, ktorým sa definuje vzhľad dokumentov napísaných v značkovacom jazyku HTML. Jeho úlohou je oddeliť obsah dokumentu od jeho prezentácie, čo umožňuje aplikovať štýly a formátovanie. Termín "kaskádové" v názve odkazuje na spôsob, ako sú štýly aplikované v poradí a môžu sa kombinovať, čo zahŕňa farby, pozadia, tieňovanie, prechody a ďalšie. CSS poskytuje možnosť definovať štýly pre rôzne HTML značky, čo vedie k jednotnému vzhľadu všetkých stránok, ak je to požadované. [29]

JavaScript

JavaScript je skriptovací programovací jazyk, ktorý umožňuje definovať funkcionality webovej stránky. Jeho kód je interpretovaný priamo na zariadení užívateľa, nie na serveri. JavaScript je dynamicky typovaný, čo znamená, že pri deklarovaní premenných nie je nutné určiť ich typ. Používa sa na vytváranie dynamických webových stránok, generovanie alebo úpravu obsahu. V jeho rámci je možné definovať rôzne funkcie a udalostné manipulátory. [23]

Bootstrap

Bootstrap je bezplatný front-endový CSS framework, ktorý sa používa na štylizáciu a vytváranie responzívnych webových stránok, ktoré sú optimalizované pre mobilné zariadenia a tablety. Obsahuje bohatú sadu šablón v CSS a JavaScripte, ktoré sa týkajú rôznych častí webových stránok, ako sú tabuľky, písmo, formuláre a rozloženie. Jeho použitie je pomerne jednoduché a nevyžaduje hlboké znalosti CSS štýlov. Vďaka tomu, že šablóny sú založené iba na HTML, CSS a JavaScripte, sú kompatibilné s rôznymi zariadeniami a prehliadačmi, čo ich robí ideálnymi nielen pre profesionálne projekty, ale aj pre moju bakalársku prácu. [3]

4.1.4 Docker

Docker je open-source platforma, ktorá umožňuje vývojárom vytvárať, spravovať a nasadzovať aplikácie pomocou kontajnerizácie. Kontajnerizácia izoluje aplikácie od ich prostredia, čo zaručuje konzistentné a bezproblémové spustenie aplikácií v rôznych prostrediach. Docker umožňuje balenie aplikácií a ich závislostí do tzv. kontajnerov, ktoré sú prenositeľné a ľahko distribuovateľné. [16]

Tento prístup zjednodušuje vývoj, testovanie a nasadzovanie aplikácií v rámci rôznych prostredí, čo zvyšuje efektivitu a produktivitu vývojárskeho tímu. Práve tieto aspekty sú jednou z podmienok splnenia mojej bakalárskej práce a preto som si vybral tento spôsob automatického nasadeniu systému, ktorý je uvedený v štvrtom bode zadania práce.

Dockerfile a Docker Compose

Dockerfile je textový súbor obsahujúci sériu príkazov, ktoré definujú, ako sa má vytvoriť Dockerový obraz. Tieto príkazy zahŕňajú inštrukcie na klonovanie zdrojového kódu, inštaláciu závislostí, konfiguráciu prostredia a spustenie aplikácie vo vnútri kontajnera. Dockerfile je základným stavebným blokom pre vytvorenie Dockerových kontajnerov a umožňuje vývojárom popísať a automatizovať proces vytvorenia obrazu, čo zabezpečuje konzistentné a opakovateľné nasadzovanie aplikácií v rôznych prostrediach. [15]

Docker Compose je nástroj určený na definovanie a správu multi-kontajnerových Docker aplikácií. S Docker Compose môžete pomocou jednoduchého konfiguračného súboru YAML

definovať `services` a `volumes`, ktoré potrebuje vaša aplikácia. Tento súbor potom použivate na vytvorenie a spustenie všetkých služieb pomocou jediného príkazu. Tento prístup značne zjednodušuje proces konfigurácie a správy kontajnerizovaných aplikácií, čo umožňuje rýchlejšie a efektívnejšie nasadenie aplikácií.

4.2 Implementácia databázy (Models)

Databáza je automaticky generovaná na základe súborov `models.py`, v jednotlivých aplikáčnych priečinkoch.

V každej entite sa ukladá do databázy niečo iné, preto si teraz ukážeme aké.

4.2.1 Užívatelia

O každom registrovanom užívateľovi sú v databáze uložené tieto atribúty:

- email - slúži ako prihlasovací údaj
- meno
- priezvisko
- užívateľské meno
- telefónne číslo - uložený ako špeciálny atribút na telefónne čísla z Python knižnice `django-phonenumbers-field` [14] a overený pomocou regulárneho výrazu
- tag športu - na výber má z niekoľkých možností
- heslo - v databáze sa ukladá zašifrovaná hodnota tohto atribútu pomocou algoritmu **PBKDF2** [28] a hašovacej funkcie **SHA256** [21]
- súhlas s GDPR

4.2.2 Športovisko

O každom športovisku sú v databáze uložené tieto atribúty:

- názov
- kapacita
- otvorené od
- zatvorené od
- adresa
- cena za hodinu prenájmu
- info
- tag
- farba - slúži na odlíšenie v rezervácii v rozvrhu

4.2.3 Rezervácia

O každej rezervácii sú v databáze uložené tieto atribúty:

- športovisko
- email - užívateľa, ktorý vytvoril rezerváciu
- meno - užívateľa, ktorý vytvoril rezerváciu
- priezvisko - užívateľa, ktorý vytvoril rezerváciu
- telefónne číslo - užívateľa, ktorý vytvoril rezerváciu
- názov
- dátum
- začiatok rezervácie
- koniec rezervácie
- vytvorené dňa
- celková cena za prenájom - ktorá je dynamická vypočítaná na základe ceny športoviska a trvaní rezervácie v metóde `save` v súbore `models.py`
- klubová rezervácia - jedná sa o príznak toho či sa jedná o ľubovú alebo klasickú rezerváciu
- klub - klub, pre ktorý bola rezervácia vytvorená
- tag - dynamicky generovaná v metóde `tag` v súbore `models.py` a skladá sa z užívateľského mena, ktorý je registrovaný a prihlásený, alebo emailu, na ktorý bola rezervácia vytvorená, tagu klubu, názvu rezervácie, poradového čísla rezervácie v systéme a tagu športoviska

4.2.4 Klub

O každom klube sú v databáze uložené tieto atribúty:

- názov
- tag
- vedúci
- členovia
- schválený - príznak, toho či je klub schválený administrátorom systému a je teda oficiálne vytvorený

4.2.5 Vedenie klubu

Jedná sa o atribút vzťahu, ktorý je medzi *športovcom* a **klubom**

- schválený - príznak toho, že užívateľovi je schválená jeho žiadosť na stanie sa *vedúcim klubu*

4.2.6 Členstvo v klube

Jedná sa o atribút vzťahu, ktorý je medzi *športovcom* alebo *registrovaný užívateľom* a **klubom**

- schválený - príznak toho, že užívateľovi je schválená jeho žiadosť na členstvo v klube

4.3 Implementácia serveru (Views)

Ako je naznačené v abstrakte celej adresárovej štruktúry projektu **E**, celý projekt je rozdelený v jednotlivých aplikáciách a každá aplikácia obsahuje rôzne python súbory, ktoré majú na starosti, rôzne aspekty celého systému. Serverovú časť, teda to ako sa jednotlivé, ktorá stránka správa, je konkrétne v kompetencii súborov `views.py`, `urls.py`, `forms.py`, poprípade `utils.py`. Jednotlivé správanie servera ako takého majú za úlohu súbory `manage.py`, ktorý riadi celý chod systému a `settings.py`, v ktorom sú nadefinované nastavenia celého systému. Napríklad autorizačný systém, použitý databázový systém, email server, alebo iné.

V nasledujúcich podkapitolách sú predstavené jednotlivé časti systému, či už samotné aplikácie, alebo aj konkrétne funkcie.

4.3.1 Aplikácia Domov

Domov, domovská stránka alebo teda základný pohľad celého systému, je implementovaný v aplikácii **home**, kde sa nachádzajú všetky potrebné veci o tejto aplikácii z hľadiska backendovej časti. Všetky potrebné webové adresy, ktoré sú nadefinované pre túto aplikáciu sú definované v súbore `urls.py`. Všetky prípony pre základnú URL sú nadefinované v českom jazyku, aby každý užívateľ pochopil na akej adrese sa nachádza, k čomu dopomáha aj to že každá stránka má takmer rovnaký názov, ktorý je viditeľný v prehliadači. Týmito URL príponami sú:

- - definuje URL pre domovskú stránku s tohto týždňovými dátami, v spolupráci s pohľadom **HomepageView**.
- ? - URL pre domovskú stránku pre dáta v nasledujúcich týždňoch v spolupráci s **NavigateToNextWeekView**.
- # - URL pre domovskú stránku pre dáta v predchádzajúcich týždňoch v spolupráci s **NavigateToPreviousWeekView**.
- uzivatele/ - URL pre stránku s tabuľkou všetkých užívateľov v spolupráci s **UsersView** 4.3.7.
- uzivatele/smazat/<int:pk>/ - URL pre vymazanie užívateľa pre administrátora v spolupráci s **DeleteUserView** 4.3.7, kde **int:pk** predstavuje ID užívateľa.
- kluby/ - URL pre stránku so všetkými tabuľkami všetkých klubov v spolupráci s **ClubsView** 4.3.6
- vsechny_rezervace/ - URL pre stránku so všetkými tabuľkami rezervácií, ktoré sú v systéme v spolupráci s **ReservationsView** 4.3.4.
- moje_rezervace/ - URL pre stránku so všetkými tabuľkami mojich rezervácií, ktoré sú v systéme v spolupráci s **ReservationsView** 4.3.4.

- vyhľadavani/ - URL pre stránku s vyhľadanou frázoou, ktorá spolupracuje s **SearchView 4.3.2** a roztrieduje ich do príslušných tabuliek.
- manual/ - URL pre stránku s manuálom, ktorá spolupracuje s **ManualView 4.3.1**.

Domovská stránka predstavuje pre užívateľa časový harmonogram počas týždňa, v ktorom vidí, ktoré **športoviská 2.2.2** sú k dispozícii na rezervovanie, ktoré **rezervácie** už sú v systéme. Tieto 2 aspekty sú definované v súbore `views.py` spolu so svojou URL v súbore `urls.py`. Konkrétne sa jedná o triedu **HomepageView**, kde sú nadefinované tieto veci:

- **get** - úlohou tejto metódy je dostať potrebné dáta do toho časového harmonogramu, ktorý vidí užívateľ, jedná sa o:
 - dni v týždni
 - časové sloty pre jednotlivé rezervácie
 - tohtotýždňové dátumy - sú naformátované, aby bolo z nich možné dostať potrebný dátum pre vytvorenie rezervácie
 - tohtotýždňové rezervácie - so všetkými potrebnými informáciami, ako je napr. názov, email užívateľa, dátum a čas rezervácie, športovisko a jeho farbu či celkovú cenu za prenájom

táto metóda potom v sebe volá ešte ďalšie 2 metódy:

- **get_context** - táto metóda naplní všetky tie dáta, ktoré sa snaží dostať metóda **get**, správnymi dátami pre ten správny týždeň a vracia ich.
- **get_user_info** - vracia informácie o užívateľovi, ako je meno, priezvisko, email, tag obľúbeného športu, kluby ktorých je buď vedúcim alebo je ich členom.

Rovnako ako trieda pohľadu **HomepageView** sa správajú aj pohľady **NavigateToNextWeekView** a **NavigateToPreviousWeek**, ktoré sú volané v prípade, že užívateľ chce prezerat ostatné týždne v harmonograme za pomoci tlačidiel na hlavnej stránke.

Súčasťou hlavnej stránky sú takisto aj tlačidlá jednotlivých športovísk, ktoré sú farebne odlišené, toto je takisto súčasťou metód **get** a **get_context**, podľa toho ako tento atribút administrátor nastavil pri vytváraní daného športoviska v systéme.

Podobne je možné sa v prípade, *neregistrovaného užívateľa 2.2.1* registrovať, prezrieť manuál **4.3.1**, alebo si čokoľvek vyhľadať **4.3.2**. V prípade registrovaného sa môže užívateľ prihlásiť. Po prihlásení má k dispozícii si svoj profil upraviť, alebo zmeniť heslo. Týmto aspektom sa venuje nasledujúca podkapitola **4.3.3**.

Stránka s manuálom predstavuje kompletnú príručku ako si:

- **Vytvoriť rezerváciu ako neregistrovaný užívateľ**
- **Vytvoriť účet, prihlásiť sa a vytvoriť tak rezerváciu, ktorú je možné spravovať**
- **Poslať žiadosť na prísanie sa k existujúcemu klubu, poslať žiadosť na stanie sa jeho vedúcim a spravovať ho**
- **Vytvoriť vlastný klub a spravovať ho**

Takisto je možné pre *prihlásených užívateľov* 2.2.1, vidieť všetky **kluby** 4.3.6 a pre *vedúcich klubov* 2.2.1 aj všetky **rezervácie** 4.3.4 zoradené v tabulkách. *Administrátor* 2.2.1 sa môže okrem toho pozrieť aj na všetkých užívateľov v systéme. Pohľady, obsluhujúce túto funkcionality sú opísané v nasledujúcich podkapitolách. Všetko toto sa samozrejme opiera o tabuľky práv užívateľov 2.1 a entít 2.1.

4.3.2 Vyhľadávanie

Funkcia vyhľadávania je implementovaná ako trieda v súbore `views.py` aplikácie **home** s názvom **SearchView** a je implementovaná na základe požiadaviek z analýzy 2 a tabuľky užívateľských práv 2.1, kde je jasne deklarované, čo a kto môže vyhľadávať v systéme. Tejto triede takisto s funkcionalitou pomáhajú skripty `search.js` a `search_filters.js`, ktoré sú implementované v priečinku `static/js/` a skript `utils.py`, ktorý sa nachádza v aplikačnom priečinku.

Implementuje všetky prípady užitia, ktoré sa týkajú vyhľadávania 2.1.5

Táto trieda pre vyhľadávanie obsahuje tieto metódy:

- `get` - ktorej úlohou je z databázy vrátiť všetky dáta o jednotlivých entitách, alebo v prípade tagu, atribúte jednotlivých entít a roztriediť ich do príslušných tabuliek podľa názvu entity či atribútu, v prípade tabuľky užívateľov je tabuľka bez záznamu o prihlásenom užívateľovi, alebo teda o samom sebe.
- `setup_pagination` - má na starosti správne fungovanie stránkovania tabuliek, ktoré metóda `get` naplňuje.

Skript `search.js` vyhľadáva daný výraz a presmerováva na stránku s výsledkami a skript `search_filters.js` filtruje tabuľky, ktoré chce užívateľ vidieť na základe hodnoty **checkboxov**.

V skripte `utils.py` sa nachádza trieda, ktorá vracia všetky modely a ich atribúty celej aplikácie, ktoré sú nadefinované v súboroch `models.py`, ktoré sme spomínali v podkapitole 4.2.

4.3.3 Aplikácia Správy užívateľov

Správa užívateľov je v systéme implementovaná v aplikácii **users**, kde sa nachádzajú všetky potrebné veci o tejto aplikácii z hľadiska backendovej časti.

Všetky potrebné webové adresy, ktoré sú nadefinované pre túto aplikáciu sú definované v súbore `urls.py`. Všetky prípony pre základnú URL sú nadefinované v českom jazyku, aby každý užívateľ pochopil na akej adrese sa nachádza, k čomu dopomáha aj to že každá stránka má takmer rovnaký názov, ktorý je viditeľný v prehliadači.

Týmito URL príponami sú:

- `prihlaseni/` - definuje URL pre prihlásenie do systému v spolupráci s pohľadom **LoginView**.
- `registrace/` - URL pre registračnú stránku a vytvorenie konta v systéme v spolupráci s **SignupView**.
- `aktivovat_ucet/<uidb64>/<token>` - URL pre potvrdenie emailovej adresy a aktiváciu užívateľského konta v systéme v spolupráci s **AccountActivationView**, kde **uidb64** predstavuje zašifrované ID užívateľa a **token** predstavuje zašifrovanú emailovú adresu spolu s časovou stopou.

- `upravit_profil/` - URL pre stránku s úpravou užívateľských dát v spolupráci s **ProfileView**.
- `smazat_ucet/<int:pk>/` - URL pre vymazanie svojho konta v spolupráci s **DeleteAccountView**, kde **int:pk** predstavuje ID užívateľa.
- `odhlaseni/` - URL pre odhlásenie zo systému v spolupráci s **LogoutView**.
- `zapomenute_heslo/` - URL pre stránku na odoslanie emailu o resetovaní hesla v prípade jeho zabudnutia v spolupráci s **ForgottenPasswordView**.
- `reset_hesla/<uidb64>/<token>` - URL pre stránku na nastavenie nového hesla pre užívateľa v spolupráci s **ResetPasswordView**, kde **uidb64** predstavuje zašifrované ID užívateľa a **token** predstavuje zašifrovanú emailovú adresu spolu s časovou stopou.

Priečinkom aplikácie obsahuje takisto súbor `forms.py`, kde sú definované formuláre:

- **CustomUserCreationForm** - formulár na vytvorenie užívateľského konta, ktorý dedí od triedy `UserCreationForm` z knižnice `django.contrib.auth.forms` [12] a používa atribúty, ktoré su definované pre entitu užívateľa 4.2.1 v súbore `models.py`.
- **CustomUserChangeForm** - jedná sa o formulár na úpravu užívateľského konta, ktorý dedí od triedy `UserChangeForm` z knižnice `django.contrib.auth.forms` [12] a tiež používa atribúty entity užívateľa 4.2.1.
- **LoginForm** - formulár na prihlasovanie užívateľa do systému, ktorý používa atribúty **email** a **heslo**.
- **CustomPasswordChangeForm** - formulár na zmenu hesla, ktorý tiež dedí od triedy `SetPasswordForm` z knižnice `django.contrib.auth.forms` [12].

V tejto aplikácii je takisto definovaná aj trieda **EmailAuthBackend**, ktorá obstaráva autentifikáciu a prihlasovanie užívateľov po zadaní emailovej adresy a hesla [7] užívateľa v súbore `backends.py`, základná autentifikácia užívateľov od DJANGA je zaobstaraná pomocou zadania užívateľského mena a hesla [8].

Ako je spomenuté vyššie, isté URL súvisia s nejakým pohľadom, tak v sekcii aplikácie **users** sú tieto pohľady opäť implementované v súbore `views.py`, a ten obsahuje tieto:

- **LoginView** - tento pohľad využíva prihlasovací formulár **LoginForm**, ktorý je definovaný vyššie na prihlásenie registrovaného užívateľa, ktorý aktivoval svoj účet. Autentifikácia užívateľa používa, miesto klasického systému autentifikácie pomocou užívateľského mena a hesla, emailovú adresu a heslo. V prípade, že chce užívateľ vidieť svoje heslo ako text využíva sa skript `password_toggle.js`.

Implementuje tento prípad užitia 2.1.4.

Obsahuje 2 metódy:

- **get** - načíta stránku s prihlasovacím formulárom.
- **post** - po zadaní správnych prihlasovacích údajov je užívateľ prihlásený do systému, v opačnom prípade sa objaví vo formulári chyba označujúca, že užívateľ zadal nesprávny email alebo heslo.

- **SignupView** - pohľad na vytvorenie užívateľského konta v systéme, ktorý využíva formulár **CustomUserCreationForm**. V prípade, že chce užívateľ vidieť svoje heslo ako text využíva sa skript `password_toggle.js`.

Implementuje tento prípad užitia 2.1.5.

Obsahuje 3 metódy:

- `get` - načíta stránku s registračným formulárom.
- `post` - po zadaní všetkých údajov do formuláru, vytvorí užívateľský profil, ktorý je ale neaktívny a odošle aktivačný email na emailovú adresu, ktorú užívateľ zadal do formuláru zavolaním poslednej metódy `send_confirmation_email`.
- `send_confirmation_email` - vytvorí link pre novoregistrovaného užívateľa v tvare: *doména/aktivovat_ucet/<uidb64>/<token>*, kde **uidb64** je zakodované ID registrovaného užívateľa pomocou funkcií knižníc `django.utils.encoding` [9] a `django.utils.http` [10] a **token** je zašifrovaná emailová adresa spolu s časovou stopou algoritmom **SHA256** [21] z triedy `PasswordTokenGenerator` knižnice `django.contrib.auth.tokens`. Tento aktivačný email je aktívny 72 hodín.

S predchádzajúcou triedou súvisí aj trieda

- **AccountActivationView** - dešifruje **uidb64** z aktivačného emailu pomocou funkcie knižnice `django.utils.encoding` [9] a priradzuje ho ku konkrétnemu vytvorenému kontu v systéme a aktivuje ho v prípade, že neprepadla jeho platnosť, ktorá je 72 hodín. V prípade neplatného linku nastane chyba a musí požiadať administrátora, aby jeho účet aktivoval manuálne.
- **ProfileView** - pohľad pre úpravu profilu a zmenu hesla. V prípade, že chce užívateľ vidieť svoje heslo ako text využíva sa skript `password_toggle.js`.

Implementuje tento prípad užitia 2.1.4.

Používa tieto 3 metódy:

- `get_object` - vracia prihláseného užívateľa.
- `get` - načíta stránku s formulármi na úpravu profilu, ktorý obsahuje všetky dáta o užívateľovi a zmenu hesla.
- `post` - ak sa užívateľ rozhodne zmeniť niektorý zo svojich osobných údajov, tak sa uloží do databáza táto nová hodnota, ak je formulár validný, ak sa užívateľ rozhodne zmeniť heslo, musí najprv zadať staré heslo a potom nové a potvrdiť ho. Užívateľ v prípade úspechu uvidí na stránke správu o úspešnej zmene. V opačnom prípade musí formulár vyplniť znovu, ale vidí kde bola chyba.

- **LogoutView** - obshuje metódu `get`, ktorá má za úlohu odhlásiť užívateľa a presmerovať na domovskú stránku
- **DeleteAccountView** - pohľad, ktorý implementuje vymazanie svojho konta zo systému. Obsahuje metódu `post`, ktorá na základe ID prihláseného užívateľa zistí o ktorého užívateľa uloženého v databáze sa jedná, vymaže jeho účet, odošle mu email na jeho emailovú adresu o vymazaní jeho konta zo systému a databázy a presmeruje na domovskú stránku.

Implementuje tento prípad užitia 2.1.4.

- **ForgottenPasswordView** - v prípade, že užívateľ zabudne svoje heslo, môže požiadať systém aby mu vygeneroval link na resetovanie jeho hesla.

Tento pohľad má 2 metódy:

- **get** - načíta stránku, na ktorej musí užívateľ zadať svoju emailovú adresu, ku ktorej je v systéme vytvorené nejaké účet.
- **post** - pri zadaní neexistujúcej emailovej adresy v systéme, je užívateľ upozornený, že účet s týmto emailom neexistuje a nech skúsi znovu zadať email.
V prípade, že je emailová adresa v systéme je na túto adresu odoslaný link na resetovanie hesla v tvare opäť *doména/reset_hesla/<uidb64>/<token>*, kde **uidb64** je opäť zašifrované ID užívateľa na základe poskytnutého emailu pomocou funkcií knižnice `django.utils.encoding` [9] a **token** je znova zakódovaná poskytnutá emailová adresa užívateľa s časovou stopou algoritmom **SHA256** [21] funkciou knižnice `django.utils.http` [10]. Link na resetovanie hesla je znova platný 72 hodín.

Poslednou triedou je:

- **ResetPasswordView** - pohľad na nastavenie nového hesla v prípade, že užívateľ zabudol svoje staré heslo a požiadal o jeho resetovanie. V prípade, že chce užívateľ vidieť svoje heslo ako text využíva sa skript `password_toggle.js`.

Používa tieto 2 metódy:

- **get** - dekoduje **uidb64** a **token** a v prípade, že je odkaz platný a je správne priradený k správne mu užívateľovi načíta formulár na nastavenie nového hesla. Ak je odkaz na resetovanie hesla neplatný užívateľ je o tom upozornený a musí si teda zažiadať o nový odkaz na resetovanie hesla.
- **post** - po priradení a skontrolovaní správnosti údajov k užívateľovi je nastavené nové heslo, o čom je užívateľ aj upozorený predtým ako je presmerovaný na prihlasovaciu stránku. V prípade akejkoľvek chyby vo formulári je užívateľ upozornený. V prípade neplatného odkazu je takisto o tom informovaný.

4.3.4 Aplikácia Správa rezervácií

Celá správa rezervácií v systéme je implementovaná v aplikácii **reservation**, kde sa nachádzajú všetky potrebné veci o tejto aplikácii z hľadiska backendovej časti.

Všetky potrebné webové adresy, ktoré sú nadefinované pre túto aplikáciu sú definované v súbore `urls.py`. Všetky prípony pre základnú URL sú nadefinované v českom jazyku, aby každý užívateľ pochopil na akej adrese sa nachádza, k čomu dopomáha aj to že každá stránka má takmer rovnaký názov, ktorý je viditeľný v prehliadači.

Týmito URL príponami sú:

- `rezervace/` - definuje URL pre vytvorenie rezervácie v spolupráci s pohľadom **ReservationView**.
- `info_o_rezervaci/<int:reservation_id>/` - URL pre získanie informácií o rezervácii v spolupráci s **ReservationInfoView**, kde **int:reservation_id** predstavuje ID rezervácie.

- `upravit_rezervaci/<int:reservation_id>/` - URL pre úpravu informácií o rezervácií v spolupráci s **EditReservationInfoView**, kde `int:reservation_id` predstavuje ID rezervácie.

Priečinkom aplikácie obsahuje takisto súbor `forms.py`, kde sú definované formuláre:

- **ReservationForm** - formulár na vytvorenie rezervácie, dediaci od triedy `ModelForm` z knižnice `django.forms` [6] a používa atribúty, ktoré su definované pre entitu rezervácie 4.2.3 v súbore `models.py`.

Táto trieda formuláru má takisto 2 metódy:

- `clean` - na základe času začiatku a koncu rezervácie a ceny za rezervované športovisko, vypočíta celkovú cenu za prenájom športoviska pre vytvorenú rezerváciu.
- `save` - v prípade, že sa jedná o klubovú rezerváciu sa nastaví príznak klubovej rezervácie

- **EditReservationForm** - jedná sa o formulár na úpravu vytvorenej rezervácie, ktorý dedí od triedy `ModelForm` z knižnice `django.forms` [6] a tiež používa atribúty entity rezervácie 4.2.3.

Úpravou sa jedná o zmenu údajov, alebo stornovanie celej rezervácie.

Takisto obsahuje metódu `clean`, ktorá takisto pri úprave rezervácie vypočíta novú cenu za prenájom športoviska.

V tejto aplikácii je definovaná aj trieda **ReservationValidator**, v súbore `utils.py`, ktorá obstaráva validáciu rezervácie, alebo teda kontrolu či na daný dátum, čas a športovisko už v systéme neexistuje nejaká rezervácia a vracia hodnotu `True` alebo `False`. Nekontroluje aktuálne upravovanú rezerváciu.

Spolu s touto triedou je tam definovaná aj funkcia `send_reservation_email` na odoslanie emailu na vytvorenú rezerváciu, ktorá odošle užívateľovi na zadaný email email aj s iCal pozvánkou na čas a dátum rezervácie, ktorú si môže dať do vlastného kalednáru.

Ako je spomenuté vyššie, isté URL súvisia s nejakým pohľadom, tak v sekcii aplikácie **reservation** sú tieto pohľady opäť implementované v súbore `views.py`, a ten obsahuje tieto:

- **ReservationView** - tento pohľad využíva rezervačný formulár **ReservationForm**, ktorý je definovaný vyššie na vytvorenie rezervácie.

Implementuje tieto prípady užitia 2.1.5 a 2.1.2.

Obsahuje 3 metódy:

- `get_object` - vracia prihláseného užívateľa.
- `get` - v prípade, že je **užívateľ prihlásený** 2.2.1, načíta stránku s rezervačným formulárom už s dopredu vyplnenými informáciami o užívateľovi, ktorý vytvára rezerváciu, ako email, meno, atď. Ak je užívateľ zároveň aj vedúci nejakého klubu môže vytvoriť klubovú rezerváciu a vybrať si z ponuky klubov, ktorých je vedúcim. V tomto tejto triede napomáha skript `create_club_reservation.js`, ktorý detekuje či je v rezervačnom formulári prítomný checkbox **Klubová rezervácia** a takisto či je zaškrtnutý.

Na základe toho, na ktoré políčko na domovskej stránke užívateľ klikol sa automaticky vyplnia, vďaka skriptu `create_reservation.js` a `start_end_time.js`,

aj políčka dátumu, počiatočný čas rezervácie. Čas koncu rezervácie je na začiatku automaticky nastavený na hodinu pozdejšie od začiatku rezervácie.

Takisto sú vo formulári načítané aj všetky športoviská z databázy, počiatočnou hodnotou je ale športovisko, ktoré bolo vytvorené ako prvé v systéme.

Ak sa jedná o **neregistrovaného alebo neprihláseného užívateľa** 2.2.1, načíta sa stránka s formulár s predvyplnenými údajmi o dátume, čase a športovisku, všetko ostatné si tento typ užívateľa musí vyplniť sám.

Má však možnosť si automaticky so svojou rezerváciou aj vytvoriť účet v systéme, kde sa po zaškrtnutí checkboxu **Chcete si vytvoriť účet?** za pomoci skriptu `create_account_and_reservation.js`, objavia takisto políčka formuláru **CustomUserCreationForm** 4.3.3, ktoré je ešte na vytvorenie účtu doplniť.

V takomto prípade sa zmení text tlačidla na vytvorenie rezervácie z **Vytvoriť rezerváciu** na **Vytvoriť účet a rezerváciu**

- **post** - ak je **užívateľ prihlásený** 2.2.1 a jeho novovytvorená rezervácia sa nekríži so žiadnou už existujúcou rezerváciou v systéme, je mu odoslaný potvrdzovací email o vytvorenej rezervácii aj s iCal pozvánkou na ten dátum a čas a systém ho upozorní, že jeho rezervácia bola vytvorená a obnoví okno domovskej stránky, na základe čoho užívateľ svoju novovytvorenú rezerváciu uvidí v časovom harmonograme. Vizualizácia tohto typu rezervácie je nameraná tu C.8.

Ak je **prihlásený užívateľ vedúci nejakého klubu** 2.2.1 a vytvoril kluobvú rezerváciu a opäť sa nekríži so žiadnou inou, tak je odoslaný email s pozvánkou všetkým vedúcim a členom klubu. A platí všetko to čo je napísané vyššie. Vizualizácia tohto typu rezervácie je nameraná tu C.9.

Ak je **užívateľ neregistrovaný** 2.2.1 a vyplní všetky svoje údaje správne a systém sa ujistí, že sa jeho novovytvorená rezervácia nekríži so žiadnou inou, je mu odoslaný email na zadanú emailovú adresu z formuláru. Takisto platí všetko čo je napísané vyššie. Vizualizácia tohto typu rezervácie je naznačená opäť tu C.8.

Ak je **užívateľ neregistrovaný** 2.2.1 a chce si popri rezervácii športoviska aj vytvoriť účet v systéme 4.3.3, vyplní všetky potrebné informácie o rezervácii, zaškrtnie checkbox **Chcete si vytvoriť účet?** a doplní ešte ďalšie chýbajúce informácie a systém skontroluje, že jeho rezervácia sa nekríži so žiadnou inou, je mu odoslaný email o registrácii aj o vytvorení rezervácie a môže sa prihlásiť bez potvrdenia emailovej adresy. Znova platí všetko čo aj v prvom odstavci o metóde **post**. Vizualizácia tohto typu rezervácie je naznačená opäť tu C.8.

- **ReservationInfoView** - pohľad na prezretie informácií o vytvorenej rezervácii má len metódu `get`, ktorá načíta stránku s potrebnými informáciami o rezervácii po kliknutí na políčko s touto rezerváciou, ak užívateľ, ktorý na toto políčko klikol nie je prihlásený, tak vidí športovisko, dátum, čas a celkovú cenu za prenájom, prihlásený užívateľ vidí o cudzej rezervácii aj email užívateľa, ktorý ju vytvoril.

Implementuje tento prípad užitia 2.1.5.

Tento prístup sa riadi tabuľkou užívateľských práv 2.1, kde neregistrovaný, alebo teda neprihlásený užívateľ, nemá prístup k cudzím súkromným dátam.

- **EditReservationView** - pohľad, ktorý riadi úpravu alebo stornovanie existujúcej rezervácie. K tomuto pohľadu má plný prístup **administrátor** na základe tabuľky užívateľských práv 2.1 a takisto užívateľ, ktorý vytvoril túto rezerváciu.

Implementujte tieto prípady užitia [2.1.3](#), [2.1.3](#), [2.1.2](#) a [2.1.2](#)

Má 2 metódy:

- **get** - načíta stránku s formulárom `EditReservationForm` na úpravu rezervácie, kde sú z databáze načítané všetky údaje o tejto rezervácii.
- **post** - pokiaľ sa užívateľ rozhodne zmeniť nejaké údaje svojej rezervácie, systém sa presvedčí o tom, či sa nekríži s inou, alebo sú ostatné údaje validné, odošle email o zmene rezervácie na emailovú adresu užívateľa. Ak sa jedná o kluobvú rezerváciu, odošle email, všetkým vedúcim a členom klubu. Takisto je užívateľ systémom priamo v prehliadači upozornený o tom, že jeho rezervácia bola upravená a je presmerovaný na domovskú stránku.

Tento pohľad takisto implementuje aj storno rezervácie, kde po jej vymazaní zo systému je odoslaný email o storne na emailovú adresu užívateľa, alebo na emailové adresy všetkých vedúcich a členov, ak sa jednalo o kluobvú rezerváciu. V prípade, že sa pri úprave zistí, že sa rezervácia kríži s inou, je užívateľ upozornený a je presmerovaný na domovskú stránku.

Ako je spomenuté v sekcii o aplikácii Domov [4.3.1](#) existuje ešte jedna funkcionálna k rezerváciám a tou je:

- **ReservationsView** - stránka so všetkými rezerváciami v systéme roztriedenými do odstránených tabuliek kvôli prehľadnosti, ku ktorej má prístup *vedúci klubu* alebo *administrátor*. V prípade, že je to len užívateľ, ktorý je alebo aj nie je členom nejakého klubu, vidí len svoje rezervácie roztriedené do odstránených tabuliek kvôli prehľadnosti. Má len jednu metódu `get`, ktorá zabezpečuje túto funkcionálnu.

4.3.5 Aplikácia Správy športovísk

Celá správa športovísk v systéme je implementovaná v aplikácii **sportground**, kde sa nachádzajú všetky potrebné veci o tejto aplikácii z hľadiska backendovej časti.

Správu športovísk má na starosti *administrátor*. Na implementáciu administrátorskej funkcionality sa pozrieme v sekcii [4.3.7](#).

Všetky potrebné webové adresy, ktoré sú nadeľované pre túto aplikáciu sú definované v súbore `urls.py`. Všetky prípony pre základnú URL sú nadeľované v českom jazyku, aby každý užívateľ pochopil na akej adrese sa nachádza, k čomu dopomáha aj to že každá stránka má takmer rovnaký názov, ktorý je viditeľný v prehliadači.

Týmito URL príponami sú:

- `pridat_sportoviste/` - definuje URL pre vytvorenie športoviska v spolupráci s pohľadom **AdminAddSportsGroundView** [4.3.7](#).
- `info_o_sportovisti/<int:sports_ground_id>/` - URL pre získanie informácií o športovisku v spolupráci s **SportsGroundInfoView**, kde `int:sports_ground_id` predstavuje ID športoviska.
- `upravit_sportoviste/<int:sports_ground_id>/` - URL pre úpravu informácií o športovisku v spolupráci s **AdminEditSportsGroundView** [4.3.7](#), kde `int:sports_ground_id` predstavuje ID športoviska.

Priečinok aplikácie obsahuje takisto súbor `forms.py`, kde sú definované formuláre:

- **AddSportsGroundForm** - formulár na vytvorenie športoviska, dediaci od triedy `ModelForm` z knižnice `django.forms` [6] a používa atribúty, ktoré su definované pre entitu športoviska 4.2.2 v súbore `models.py`.
- **EditSportsGroundForm** - jedná sa o formulár na úpravu vytvorenej športoviska, ktorý dedí od triedy `ModelForm` z knižnice `django.forms` [6] a tiež používa atribúty entity športoviska 4.2.2.

Úpravou sa jedná o zmenu údajov, alebo odstránenie športoviska.

Ako je spomenuté vyššie, isté URL súvisia s nejakým pohľadom, tak v sekcii aplikácie **sportground** sú tieto pohľady opäť implementované v súbore `views.py`, a ten obsahuje tieto:

- **SportGroundInfoView** - pohľad na prezretie informácií o vytvorenom športovisku má len metódu `get`, ktorá načíta stránku s potrebnými informáciami o športovisku po kliknutí na tlačidlo s názvom športoviska. Spolupracuje na tom aj skript `sportsground_info.js`.

Akýkoľvek **užívateľ** 2.2.1, okrem *administrátora*, vidí na tejto stránke to isté, otváraciu dobu, kapacitu, tag športoviska, cenu za hodinu prenájmu a adresu.

Tento prístup sa riadi tabuľkou užívateľských práv 2.1, kde je vidieť, že len *administrátor* môže **športovisko** vytvoriť, upraviť alebo vymazať.

Implementuje tento prípad použitia 2.1.5.

V tejto sekcii je spomenutý len jeden z nich **SportGroundInfoView**, pretože ostatné sú definované až v sekcii o *Administrátorovi* 4.3.7.

4.3.6 Aplikácia Správy klubov

Celá správa klubov v systéme je implementovaná v aplikácii **club**, kde sa nachádzajú všetky potrebné veci o tejto aplikácii z hľadiska backendovej časti.

Všetky potrebné webové adresy, ktoré sú nadefinované pre túto aplikáciu sú definované v súbore `urls.py`. Všetky prípony pre základnú URL sú nadefinované v českom jazyku, aby každý užívateľ pochopil na akej adrese sa nachádza, k čomu dopomáha aj to že každá stránka má takmer rovnaký názov, ktorý je viditeľný v prehliadači.

Týmito URL príponami sú:

- `muj_klub/` - definuje URL pre všetky moje kluby v spolupráci s pohľadom **ClubView**.
- `info_o_klubu/<int:pk>/` - URL pre získanie informácií o klube v spolupráci s **ClubInfoView**, kde **int:pk** predstavuje ID klubu.
- `vytvorit_o_klubu/<int:pk>/` - URL pre vytvorenie nového klubu v spolupráci s **CreateClubView**, kde **int:pk** predstavuje ID klubu.
- `smazat_klub/<int:pk>/` - URL pre vymazanie klubu v spolupráci s **DeleteClubView**, kde **int:pk** predstavuje ID klubu.
- `upravit_klub/<int:pk>/` - URL pre úpravu informácií o klube v spolupráci s **EditClubView**, kde **int:pk** predstavuje ID klubu.

- `schvalit_klub_a_jeho_vedouciho/<int:pk>/` - URL pre schválenie vytvorenie nového klubu a aj jeho prvého vedúceho v spolupráci s **ApproveClubAndItsManagerView**, kde **int:pk** predstavuje ID klubu.
- `schvalit_noveho_vedouciho/<int:pk>/` - URL pre schválenie nového vedúceho klubu v spolupráci s **ApproveNewManagerView**, kde **int:pk** predstavuje ID klubu.
- `poslat_zadost_o_clenstvi_v_klubu/<int:pk>/` - URL pre zaslanie žiadosti o pridanie do klubu v spolupráci s **SendRequestToJoinClub**, kde **int:pk** predstavuje ID klubu.
- `zrusit_zadost_o_clenstvi_v_klubu/<int:pk>/` - URL pre zrušenie žiadosti o pridanie do klubu v spolupráci s **CancelRequestToJoinClub**, kde **int:pk** predstavuje ID klubu.
- `odejit_z_klubu/<int:pk>/` - URL pre opustenie klubu v spolupráci s **LeaveClubView**, kde **int:pk** predstavuje ID klubu.
- `poslat_zadost_o_stani_se_novym_vedoucim/<int:pk>/` - URL pre zaslanie žiadosti na nového vedúceho klubu v spolupráci s **SendRequestToBecomeNewManagerView**, kde **int:pk** predstavuje ID klubu.
- `zrusit_zadost_o_stani_se_novym_vedoucim/<int:pk>/` - URL pre zrušenie žiadosti na nového vedúceho klubu v spolupráci s **CancelRequestToBecomeNewManagerView**, kde **int:pk** predstavuje ID klubu.
- `odstranit_clena/<int:pk>/` - URL pre odstránenie člena z klubu v spolupráci s **DeleteMemberView**, kde **int:pk** predstavuje ID klubu.
- `odstranit_vedouciho/<int:pk>/` - URL pre odstránenie vedúceho z klubu v spolupráci s **DeleteManagerView**, kde **int:pk** predstavuje ID klubu.

Priečink aplikácie obsahuje takisto súbor `forms.py`, kde sú definované formuláre:

- **CreateClubForm** - formulár na vytvorenie klubu, dediaci od triedy `ModelForm` z knižnice `django.forms` [6] a používa atribúty, definované pre entitu klubu 4.2.4 v súbore `models.py`.
- **EditClubForm** - jedná sa o formulár na úpravu vytvoreného klubu, ktorý dedí od triedy `ModelForm` z knižnice `django.forms` [6] a tiež používa atribúty entity klubu 4.2.4.
- **ApproveClubForm** - jedná sa o formulár na schválenie vytvoreného klubu, ktorý dedí od triedy `ModelForm` z knižnice `django.forms` [6] a tiež používa atribút entity klubu 4.2.4.
- **ApproveNewClubManagerForm** - jedná sa o formulár na schválenie nového vedúceho klubu, ktorý dedí od triedy `ModelForm` z knižnice `django.forms` [6] a tiež používa atribút entity klubu 4.2.5.
- **ApproveNewClubMemberForm** - jedná sa o formulár na schválenie nového člena klubu, ktorý dedí od triedy `ModelForm` z knižnice `django.forms` [6] a tiež používa atribút entity klubu 4.2.6.

Ako je spomenuté vyššie, isté URL súvisia s nejakým pohľadom, tak v sekcii aplikácie **club** sú tieto pohľady opäť implementované v súbore `views.py`, a ten obsahuje tieto:

- **ClubView** - tento pohľad vracia vďaka svojej jedinej metóde `get` **prihlásenému užívateľovi** 2.2.1 tabuľku so schválenými klubmi, ktorých je buď členom, alebo môže poslať žiadosť o pridanie do klubu, *administrátor* vidí úplne všetky vytvorené kluby, aj tie ktoré ešte neboli schválené.
- **CreateClubView** - tento pohľad implementuje vytvorenie nového klubu. Spolupracuje na tom aj skript `create_club.js`.

Pre lepšiu vizualizáciu je vytvorený diagram C.3, ktorý definuje životný cyklus každého klubu vytvoreného registrovaným užívateľom. Na ilustráciu tohto prípadu použitia slúži aj sekvenčný diagram C.4, ilustrujúci celé vytvorenie klubu aj s jednotlivými aspektami, triedami a metódami zo systému.

Rovnako sú vytvorené aj vizualizačné diagramy C.2, C.1 a C.5 v prípade, že by chcel klub vytvoriť neregistrovaný užívateľ, v takomto prípade to za neho musí spraviť *administrátor*.

Implementuje tento prípad použitia 2.1.2.

Tento pohľad má tieto 2 metódy:

- `get` - **prihlásenému užívateľovi** 2.2.1, vracia stránku s formulárom na vytvorenie nového klubu **CreateClubForm** a nastaví manažéra na toho konkrétneho užívateľa. Administrátor si manažéra môže vybrať zo všetkých registrovaných užívateľov vďaka formuláru **ClubManagerFormSet**, ktorý je abstrakciou umožňujúcou prácu s viacerými formulármi na jednej stránke [11].
- `post` - v prípade, že sú zadané správne údaje klubu, je vedúcemu odoslaný email o vytvorení klubu a musí počkať, kým ho *administrátor* schváli, o tomto je takisto je upozornení o tomto fakte aj rovno v prehliadači.
- **DeleteClubView** - implementácia odstránenia klubu, naznačené v diagrame C.3, každý klub môže byť po rozhodnutí jedného z vedúcich klubov odstránený zo systému. Má len metódu `post`, ktorá zistí ID klubu a v prípade, že užívateľ je *vedúcim klubu*, ktorý má byť odstránený, alebo je *administrátor*, klub odstráni a odošle informačný mail všetkým vedúcim a čelnom klubu.

Ak chcel iný typ užívateľa odstrániť klub, je upozornený, že na to nemá oprávnenie, ako je definované v tabuľke užívateľských práv 2.1.

Implementuje tento prípad použitia 2.1.2.

- **EditClubView** - implementácia pohľadu s formulárom **EditClubForm** na upravenie informácií o klube, v ktorom je užívateľ *vedúci*, takisto má k nej prístup aj *administrátor*.

Implementuje tento prípad použitia 2.1.2.

Má tieto 2 metódy:

- `get` - vráti všetky dostupné informácie o klube, ktoré je možné upraviť, jednotlivých členov a vedúcich vypíše do tabuliek, ktoré je možno prechádzať pre lepšiu prehľadnosť.

- **post** - *vedúci klubu* alebo *administrátor* po úprave formuláru sú upozorený v prehliadači, že klub bol upravený a všetkým členom a vedúcim je odoslaný informačný email.
- **ApproveNewMemberView** - pohľad na schválenie žiadosti o členstvo v klube, má k nemu prístup buď *vedúci klubu* alebo *administrátor*.
Implementuje tento prípad užitia 2.1.2.
Má 2 metódy:
 - **get** - vráti formulár **ApproveNewClubMemberForm** ako tlačidlo na schválenie nového člena klubu.
 - **post** - po stlačení tlačidla sa odošle formulár na scvhálenie nového člena, ktorému automaticky príde mail o tejto skutočnosti.
 - **ApproveNewManagerView** - pohľad na schválenie žiadosti o členstvo v klube, má k nemu prístup buď už existujúci *vedúci klubu* alebo *administrátor*
Implementuje tento prípad užitia 2.1.2.
Má 2 metódy:
 - **get** - vráti formulár **ApproveNewClubManagerForm** ako tlačidlo na schválenie nového vedúceho klubu.
 - **post** - po stlačení tlačidla sa odošle formulár na schválenie nového vedúceho klubu, ktorému automaticky príde mail o tejto skutočnosti a vymaže ho zo zoznamu členov klubu.
 - **SendRequestToJoinClub** - pohľad na odoslanie žiadosti o členstvo v klube.
Implementuje tento prípad užitia 2.1.3.
Má 2 metódy:
 - **get** - vráti stránku s tlačidlom na odoslanie žiadosti o pridanie do klubu.
 - **post** - po stlačení tlačidla sa odošle formulár na pridanie do klubu a odošle sa email vedúcim klubu, že nejaký užívateľ sa chce pridať do ich klubu.
 - **CancelRequestToJoinClub** - pohľad na zrušenie žiadosti o členstvo v klube.
Implementuje tento prípad užitia 2.1.3.
Má 2 metódy:
 - **get** - vráti stránku s tlačidlom na zrušenie žiadosti o pridanie do klubu.
 - **post** - po stlačení tlačidla sa odošle formulár na zrušenie žiadosti o pridanie do klubu a odošle sa email vedúcim klubu, že nejaký užívateľ zrušil svoju žiadosť na pridanie do ich klubu.
 - **SendRequestToBecomeNewManagerView** - pohľad na odoslanie žiadosti o vedenie klubu.
Implementuje tento prípad užitia 2.1.3.
Má 2 metódy:

- **get** - vráti stránku s tlačidlom na odoslanie žiadosti o vedenie klubu.
 - **post** - po stlačení tlačidla sa odošle formulár na stanie sa vedúcim klubu a odošle sa email už existujúcim vedúcim klubu, že nejaký člen sa chce stať novým vedúcim klubu.
- **CancelRequestToBecomeNewManagerView** - pohľad na zrušenie žiadosti o vedenie klubu
Implementuje tento prípad užitia [2.1.3](#).
Má 2 metódy:
 - **get** - vráti stránku s tlačidlom na zrušenie žiadosti o vedenie klubu.
 - **post** - po stlačení tlačidla sa odošle formulár na zrušenie žiadosti o vedenie klubu a odošle sa email už existujúcim vedúcim klubu, že nejaký člen zrušil svoju žiadosť na vedenie ich klubu.
 - **LeaveClubView** - implementuje opustenie klubu ak je užívateľ členom alebo ak je vedúcim, tak automaticky klub vymaže.
Implementuje tento prípad užitia [2.1.3](#).
Má tieto 2 metódy:
 - **get** - vráti stránku s tlačidlom na opustenie klubu.
 - **post** - ak klub opustí jeho člen, vymaže sa z databázy záznam o členstve a odošle sa email všetkým vedúcim. Ak klub opustí vedúci, odošle sa email všetkým členom, že klub bol automaticky vymazaný.
 - **ClubInfoView** - implementuje pohľad, ktorý ukáže užívateľovi informácie o klube. Má len metódu **get**, ktorá mu zabezpečí túto funkcionálnosť. `club_info.js`.
Implementuje tento prípad užitia [2.1.5](#).
 - **DeleteMemberView** - implementuje pohľad, ktorý odstráni člena klubu. Má len metódu **post**, ktorá mu zabezpečí túto funkcionálnosť a odošle informačný email členovi, ktorý bol odstránený z klubu.
Implementuje tento prípad užitia [2.1.2](#).
 - **DeleteManagerView** - implementuje pohľad, ktorý odstráni vedúceho klubu. Má len metódu **post**, ktorá mu zabezpečí túto funkcionálnosť a odošle informačný email vedúcemu, ktorý bol odstránený z klubu.
Implementuje tento prípad užitia [2.1.2](#).

Ako je spomenuté v sekcii o aplikácii Domov [4.3.1](#) existuje ešte jedna funkcionálnosť ku klubom a tou je:

- **ClubsView** - stránka so všetkými klubmi v systéme roztriedenými do odstránkových tabuliek kvôli prehľadnosti, ku ktorej má prístup len *administrátor*. Táto stránka slúži hlavne na možnosť schválenia všetkého čo s klubom súvisí, hlavne ale **ApproveClubAndItsManagerView**, ktorý je využiteľný len *administrátorom*. Má len metódu **get**, ktorá zabezpečuje túto funkcionálnosť.

4.3.7 Admin

Ako je spomenuté v predchádzajúcich podsekciiach, niektoré funkcie alebo triedy sú priamo použiteľné len pre *administrátora*.

Tieto triedy sú súčasťou tých predchádzajúcich aplikácií v súboroch `views.py`. Sú to tieto:

- **UsersView** - vracia tabuľku so všetkými užívateľmi v systéme, ktorá je odstránená kvôli prehľadnosti. Na starosti to má jej jediná metóda `get`.

Takisto sa tu nachádza aj formulár na vytvorenie nového užívateľa, ktorý definuje tento prípad použitia 2.1.1. Spolupracuje na tom aj skript `create_user.js`.

- **DeleteUserView** - odoslanie formuláru v tvare tlačidla na odstránenie užívateľa zo systému, ktorého funkcionality zabezpečuje jej jediná metóda `post`, po vykonaní je odstránenému užívateľovi odoslaný email.

Implementuje tento prípad použitia 2.1.1.

- **ActivateUserView** - odoslanie formuláru v tvare tlačidla na aktiváciu užívateľa, ktorého funkcionality zabezpečuje jej jediná metóda `post`, po vykonaní je aktivovanému užívateľovi odoslaný email.

Implementuje tento prípad použitia 2.1.1.

- **AdminAddSportsGroundView** - pohľad na pridanie nového športoviska do systému, ktorý má 2 metódy. Metóda `get`, ktorá vracia formulár **AddSportsGroundForm** na vytvorenie nového športoviska a metódu `post`, ktorá po odoslaní validného formulára upozorní *administrátora*, že športovisko bolo vytvorené a je možné naň začať vytvárať rezervácie. V prípade nejakej chyby je vo formulári vidieť čo treba opraviť. Spolupracuje na tom aj skript `create_sportsground.js`.

Implementuje tento prípad použitia 2.1.1.

- **AdminEditSportsGroundView** - pohľad na úpravu informácií o športovisku, ktorý má 2 metódy. Metóda `get`, ktorá vracia formulár **EditSportsGroundForm** na úpravu športoviska a metódu `post`, ktorá po odoslaní validného formulára upozorní *administrátora*, že športovisko bolo upravené a užívateľom, ktorý na ňom majú vytvorenú nejakú rezerváciu je odoslaný informačný email. V prípade nejakej chyby je vo formulári vidieť čo treba opraviť.

Tento pohľad ale obstaráva aj vymazanie športoviska, v metóde `post` sa v prípade odstránenia športoviska zo systému odošle email všetkým užívateľom, ktorý mali vytvorenú rezerváciu na toto športovisko o tom, že ich rezervácia bola automaticky s vymazaním daného športoviska stornovaná. Takisto *administrátor*, je v prehliadači automaticky upozornený o tejto skutočnosti.

Spolupracuje na tom aj skript `sportsground_info.js`.

Implementuje tieto prípady použitia 2.1.1 a 2.1.1.

4.4 Implementácia grafického užívateľského rozhrania (Templates)

Ako je spomenuté vyššie v sekcii 4.1.3 na implementáciu boli použité technológie **HTML5** [13], **CSS** [29], **JavaScript** [23] a **Bootstrap** [3].

Framework DJANGO, však umožňuje implementáciu istej funkcionality aj pri písaní HTML kódu, konkrétne sa jedná o niektoré konštrukcie jazyka Python [5], napríklad cyklus **for** na prechádzanie všetkými klubmi v systéme, v podobe `{% for club in clubs %}`, na ktorého ukončenie potom použijeme `{% endfor %}`, alebo podmienka **if-elif-else** na podmienený prístup k istým stránkam v podobe `{% if request.user.is_authenticated %}`, `{% elif request.user.is_superuser %}`, `{% else %}`, na ktorého ukončenie potom použijeme `{% endif %}`.

Pre potrebu odlíšenia jednotlivých rezervácií podľa športoviska bol implementovaný jeden takýto nový tag, v aplikácii **home** v súbore **templatetags**, s názvom **get_color**, ktorý vracia farbu športoviska na základe ID športoviska.

V najbližších podsekcích budú ukázané celé stránky pre *administrátora* spolu s rozdielmi pre ostatných užívateľov, ich celé stránky sú priložené v **D**.

4.4.1 Domov

Ako je spomenuté v 4.4 DJANGO používa isté konštrukcie jazyka Python, vďaka nim na domovskej stránke vidí každý typ užívateľa niečo iné. Použité to je napríklad pri presmerovacích tlačidlách v navbare na domovskej stránke, alebo v osobných údajoch, ktoré užívateľ vidí nad časovým harmonogramom.

Súčasťou stránok, ktoré su naimplementované v tejto aplikácii je aj stránka pre všetky **kluby 1**, kde sa nachádzajú odstránkové tabuľky pre zlepšenie prehľadnosti. Spolu s ňou je tu aj stránka pre všetky **rezervácie 4.4.4**, či už svoje, alebo úplne všetky uložené v systéme. To ako vyzerajú je ukázané v nasledujúcich podkapitolách. Spolu s nimi je tu implementovaná aj stránka **manuálu 4.4.1**.

Administrátor

Administrátor vidí svoj email, tlačidlá jednotlivých športovísk, tlačidlo na pridanie nového športoviska, tlačidlá na navigáciu medzi týždňami, farebne odlíšené rezervácie v časovom harmonograme podľa športoviska, na ktorom je tá rezervácia, plus ak sa jedná o rezerváciu neregistrovaného užívateľa, vidí ju farebným lineárnym prechodom z čiernej do farby športoviska. V navbare vidí presmerovania na stránky so správou všetkých užívateľov, všetkých rezervácií a všetkých klubov, upravenie svojho profilu a odhlásenie.

The screenshot shows the administrator interface for 'Rezervuj a sportuj'. At the top left is the logo. To the right is a search bar and a navigation menu with links: Kluby, Rezervace, Uživatelé, Manuál, Upravit profil, and Odhlásit se. Below the navigation is a box displaying the email: admin@rezervujaspportuj.cz. A row of colored buttons represents different sports: Umelka (red), Tenisový kurt (orange), Posilka (yellow), Športová hala (light green), Futbalové ihrisko (purple), Basketbalové koše (teal), Piesočnaté ihrisko (pink), and Pridat sportovište (white). The main part of the page is a calendar grid for the week of May 6th to 12th, 2024. The grid shows reservations for the days from Monday to Thursday. Each reservation is represented by a colored box with text: 'email .env' (red), 'docker-compose' (red), 'ssl' (red), and 'ssl club' (red). A 'Manual' button (grey) is present for each reservation on Monday, Tuesday, and Wednesday. The time slots range from 8:00 to 11:00.

	Po 06.05.2024	Út 07.05.2024	St 08.05.2024	Čt 09.05.2024	Pá 10.05.2024	So 11.05.2024	Ne 12.05.2024
8:00							
8:30	email .env	docker-compose	ssl	ssl club	Manual		
9:00	email .env	docker-compose	ssl	ssl club	Manual		
9:30	email .env	docker-compose	ssl	ssl club	Manual		
10:00							
10:30							
11:00							

Obr. 4.1: Domovská stránka administrátora

Vedúci klubu

Vedúci klubu vidí svoj email, celé meno, tag obľúbeného športu, mená klubov, tlačidlá jednotlivých športovísk, ktorých je vedúcim, tlačidlá na navigáciu medzi týždňami, farebne odlíšené rezervácie v časovom harmonograme podľa športoviska, na ktorom je tá rezervácia, plus ak sa jedná o rezerváciu, ktorú vytvoril ten užívateľ, vidí ju inou farbou. V navbare vidí presmerovania na stránky so správou všetkých rezervácií a všetkých klubov, upravenie svojho profilu a odhlásenie. Celý pohľad vedúceho je potom tu [D.1](#).

	St 01.05.2024	Čt 02.05.2024
Email: belticek@gmail.com		
Celé jméno: Juraj Remeň	Tréning U9	Anonym v budúcnosti
Tag: #fotbal	Tréning U9	Anonym v budúcnosti
Vedoucí klubu: FC Medlánky, FK Iskra Hnúšťa	Tréning U9	Anonym v budúcnosti
Člen klubu:	Tréning U9	Anonym v budúcnosti

[Kluby](#) [Rezervace](#) [Manuál](#) [Upravit profil](#) [Odhlásit se](#)

Obr. 4.2: Rozdiely na domovskej stránke pre vedúceho klubu

Športovec

Športovec vidí svoj email, celé meno, tag obľúbeného športu, mená klubov, ktorých je členom, tlačidlá jednotlivých športovísk, tlačidlá na navigáciu medzi týždňami, farebne odlíšené rezervácie v časovom harmonograme podľa športoviska, na ktorom je tá rezervácia, plus ak sa jedná o rezerváciu, ktorú vytvoril ten užívateľ, vidí ju inou farbou. V navbare vidí presmerovania na stránky so správou svojich rezervácií a všetkých klubov, upravenie svojho profilu a odhlásenie. Celý pohľad športovca je potom tu [D.2](#). Celý pohľad neregistrovaného užívateľa je potom tu [D.3](#).

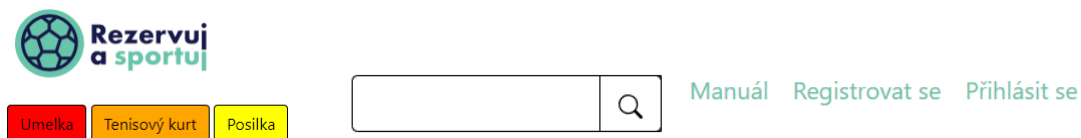
	St 01.05.2024	Čt 02.05.2024
Email: xremen02@stud.fit.vutbr.cz		
Celé jméno: Anton Berrolák	Tréning U9	Anonym v budúcnosti
Tag: #volejbal	Tréning U9	Anonym v budúcnosti
Vedoucí klubu:	Tréning U9	Anonym v budúcnosti
Člen klubu: FC Medlánky	Tréning U9	Anonym v budúcnosti

[Kluby](#) [Moje rezervace](#) [Manuál](#) [Upravit profil](#) [Odhlásit se](#)

Obr. 4.3: Rozdiely na domovskej stránke pre športovca

Neregistrovaný užívateľ

Neregistrovaný užívateľ vidí tlačidlá jednotlivých športovísk, tlačidlá na navigáciu medzi týždňami, farebne odlišené rezervácie v časovom harmonograme podľa športoviska, na ktorom je tá rezervácia. V navbare vidí presmerovania na registráciu a prihlásenie.



Obr. 4.4: Rozdiely na domovskej stránke pre neregistrovaného užívateľa

Po stlačení tlačidla na navigáciu sa na domovskej stránke zmenia dátumy v záhlaví časového harmonogramu a načítajú sa v ňom rezervácie pre tie dátumy.

<	Dnes	>	Po 06.05.2024	Út 07.05.2024	St 08.05.2024	Čt 09.05.2024	Pá 10.05.2024	So 11.05.2024	Ne 12.05.2024
---	------	---	------------------	------------------	------------------	------------------	------------------	------------------	------------------

Obr. 4.5: Zmena dátumov v záhlaví

Manuál

Pre akéhokoľvek užívateľa systému je táto stránka k dispozícii na použitie aby presne vedel ako systém funguje, vedel ho ľahšie používať a orientovať sa v ňom.



[Manuál](#) [Registrovat se](#) [Přihlásit se](#)

Manuál

Rezervace sportoviště neregistrovaným uživatelem

Calendar view for May 2024. The grid shows reserved slots (red blocks) for various sports on different days and times.

<	Dnes	>	Po 06.05.2024	Út 07.05.2024	St 08.05.2024	Čt 09.05.2024	Pá 10.05.2024	So 11.05.2024	Ne 12.05.2024
			8:00						
			8:30						
			9:00						
			9:30						
			10:00						
			10:30						
			11:00						
			11:30						
			12:00						

Obr. 4.6: Stránka manuálu

4.4.2 Vyhľadávanie

Ako je vidieť v tabulke užívateľských práv 2.1, každý užívateľ môže vyhľadať čokoľvek, až na *neregistrovaného*, ktorý nemá možnosť vyhľadať užívateľa, kvôli zachovaniu osobných údajov. Proces vyhľadávania je takisto vizualizovaný na diagrame C.10, ktorý popisuje celý proces vyhľadávania v systéme.

Pri vyhľadávaní užívateľov je ešte jedna špecifická vec, v tabulke užívateľ, ktorý niečo vyhľadal neuvidí záznam o sebe. K týmto údajom má totižto prístup cez úpravu profilu 4.4.3, alebo vidí takmer všetko potrebné aj na domovskej stránke 4.2.

Pri kliknutí na jednotlivé záznamy v tabulkách je otvorené nové okno s informáciami o tom zázname.

Administrátor

Administrátor pri vyhľadaní nejakého výrazu vidí roztriedené jednotlivé záznamy do odstránkových tabuliek, kvôli prehľadnosti. Vidí takisto checkboxy, ktoré majú za úlohu filtrovať viditeľnosť tabulky, implementovaná v skripte `search_filters.js`.

Hledaný výraz "a"

Sportoviště
 Uživatelé
 Kluby
 Rezervace
 Tagy

Sportoviště

Název sportoviště
Umelka
Tenisový kurt
Posilka
Športová hala
Futbalové ihrisko

Stránka 1 z 2. Další Poslední »

Tagy

Tag
#fotbal
#volejbal
#medlankyA
#iskraA
#umelka
#kurt
#posilka
#hala
#ihrisko
#kose

Stránka 1 z 6. Další Poslední »

Uživatelé

Jméno uživatele
Juraj Remeň
Anton Berrolák

« První Předchozí Stránka 6 z 6.

Kluby

Název klubu
FC Medlánky
FK Iskra Hnúšťa

« První Předchozí Stránka 2 z 6. Další Poslední »

Hledaný výraz "a"

Sportoviště
 Uživatelé
 Kluby
 Rezervace
 Tagy

Uživatelé

Jméno uživatele
Juraj Remeň
Anton Berrolák

Kluby

Název klubu
FC Medlánky
FK Iskra Hnúšťa

Rezervace

Název rezervace	Datum a čas	Sportoviště
Skúška	15. dubna 2024 od 8:30 do 9:30	Umelka
Všetky	20. dubna 2024 od 8:30 do 9:30	Umelka
klub	17. dubna 2024 od 8:30 do 9:30	Umelka
skuska	24. dubna 2024 od 8:30 do 9:30	Umelka
Tréning U9	26. dubna 2024 od 9:00 do 10:00	Umelka
Tréning U11	25. dubna 2024 od 11:30 do 12:30	Umelka
Zápas U9	27. dubna 2024 od 13:30 do 14:30	Umelka
Tréning Novák	25. dubna 2024 od 15:30 do 17:30	Basketbalové koše
Kondičné posilňovanie	26. dubna 2024 od 12:00 do 13:00	Posilka
Pink	28. dubna 2024 od 9:30 do 11:30	Umelka

Stránka 1 z 5. Další Poslední »

Obr. 4.7: Stránka s vyhledáváním administrátora

52

Vedúci klubu

Vedúci klubu pri vyhľadani nejakého výrazu vidí roztriedené jednotlivé záznamy do odstránkových tabuliek, kvôli prehľadnosti. Vidí takisto checkboxy, ktoré majú za úlohu filtrovať viditeľnosť tabulky, implementovaná v skripte `search_filters.js`. Jediný rozdiel je v tom, že v tabulke užívateľov nevidí záznam o sebe, ako je spomenuté v sekcii [4.4.2](#).



Uživatelia	
Jméno uživatele	
Admin Adminovič	
Anton Berrolák	

Obr. 4.8: Rozdiely na stránke vyhľadávania vedúceho klubu

Športovec

Športovec pri vyhľadani nejakého výrazu vidí roztriedené jednotlivé záznamy do odstránkových tabuliek, kvôli prehľadnosti. Vidí takisto checkboxy, ktoré majú za úlohu filtrovať viditeľnosť tabulky, implementovaná v skripte `search_filters.js`. Jediný rozdiel je v tom, že v tabulke užívateľov nevidí záznam o sebe, ako je spomenuté v podsekcii [4.4.2](#).



Uživatelia	
Jméno uživatele	
Admin Adminovič	
Juraj Remeň	

Obr. 4.9: Rozdiely na stránke vyhľadávania športovca

Neregistrovaný užívateľ

Neregistrovaný užívateľ pri vyhľadani nejakého výrazu vidí roztriedené jednotlivé záznamy do odstránkových tabuliek, kvôli prehľadnosti. Vidí takisto checkboxy, ktoré majú za úlohu filtrovať viditeľnosť tabulky, implementovaná v skripte `search_filters.js`. Nevidí však tabulku užívateľov a ani checkbox na odfiltrovanie tejto tabulky ako je spomenuté na začiatku podsekcii [4.4.2](#).

Hledaný výraz "a"

Sportoviště Kluby Rezervace Tagy

Sportoviště

Název sportoviště
Umelka
Tenisový kurt
Posilka
Športová hala
Futbalové ihrisko

Stránka 1 z 2.

Kluby

Název klubu
FC Medlánky
FK Iskra Hnúšťa

Obr. 4.10: Rozdiely na stránke vyhľadávania neregistrovaného užívateľa

4.4.3 Správa užívateľov

Pri správe užívateľov máme 6 stránok, ktoré sú prezerateľné, všetky sú dosiahnuteľné pre všetky typy užívateľov, s menšími rozdielmi. Nasledujúce stránky znázorňujú prirodzený postup akéhokoľvek užívateľa v systéme, ktorý sa najprv zaregistruje a potom postupuje s využívaním systému.

Registrácia

Registrowanie v systéme je k dispozícii pre každého. Jedinou výnimkou je registrácia nového administrátora, tá je možná len pomocou príkazu v príkazovom riadku `python3 manage.py createsuperuser` [17]. Vytvorenie ostatných typov užívateľa už potom prebieha priamo v systéme na webe cez registračný formulár.. Užívateľ musí do registračného formuláru zadať všetky potrebné údaje a po vytvorení účtu mu je zaslaný, na jeho emailovú adresu, aktivačný email 2.1.6. V prípade, že už má účet vytvorení, má možnosť sa automaticky presmerovať na prihlásenie.



Registrace

Prosím vyplňte následující formulář pro vytvoření nového účtu.

Emailová adresa*:

Jméno*:

Příjmení*:

Přezdívká*:

Telefonní číslo*:

Tag sportu:

Souhlas s GDPR a podmínkami použití*:

Heslo*: Zobrazit heslo

Potvrzení hesla*: Zobrazit heslo

Již u nás máte účet? [Přihlašte se!](#)

Obr. 4.11: Stránka registrácie do systému

Aktivácia účtu

Aktivovanie účtu funguje automaticky po kliknutí na odkaz, ktorý dostane novoregistrovaný užívateľ na jeho emailovú adresu. Po kliknutí je presmerovaný na prihlasovaciu stránku a potvrdenie o tom, že je prihlásený vidí vo svojom profile.

Potvrďte svou emailovou adresu



Od: beltricek@gmail.com dňa 2024-05-05 10:08

[Podrobnosti](#)

Děkujeme za registraci v našem systému. Potvrďte prosím vaši emailovou adresu tady
`localhost:8000/aktivovat_ucet/Mzg/c6jt3o-970bfc0f5a2018f6d4391172a54da76f/`

Upravit profil

Váš účet byl aktivován!

Obr. 4.12: Aktiváciu účtu

Prihlásenie

Prihlásenie do systému je k dispozícii pre každého. Užívateľ musí do prihlasovacie formuláru zadať všetky potrebné údaje a po prihlásení je presmerovaný na domovskú stránku. V prí-

pade, že ešte nemá účet vytvorení, má možnosť sa automaticky presmerovať na registráciu. Takisto má možnosť sa v prípade zabudnutého hesla presmerovať na túto stránku.



Přihlášení

Emailová adresa:

Heslo: Zobrazit heslo

Zapoměli jste svoje heslo? [Zapoměl jsem heslo!](#)

Jěšte u nás nemáte účet? [Registrujte se!](#)

Obr. 4.13: Stránka přihlášení do systému

Zabudnuté heslo

Ak uživatel zabudol svoje heslo, může požiadat systém aby mu vygeneroval odkaz na resetovanie jeho hesla.



Zapomenuté heslo

Email:

Email s odkazem na resetování hesla byl odeslán.

Obr. 4.14: Zabudnuté heslo

Resetovanie hesla

Po kliknutí na vygenerovaný odkaz je uživatel presmerovaný na stránku s resetovaním hesla, kde si nastaví nové heslo, ktorým sa potom prihlási, v zápätí po přihlášení sa mu v profile objaví správa o zmene hesla.

Změna hesla



Od: belticek@gmail.com dňa 2024-05-05 10:32

 [Podrobnosti](#)

Klikněte prosím na následující link: localhost:8000/reset_hesla/Mzg/c6jumr-14f8c5fbe7b16b791ee21b9f3728b59a/



Resetovat heslo

Nové heslo*: Zobrazit heslo
Potvrzení hesla*: Zobrazit heslo

Upravit profil


Vaše heslo bylo úspěšně změněno!

Obr. 4.15: Resetovanie hesla

Úprava profilu

Poslednou stránkou, zaoberajúcou sa správou užívateľov, alebo ich profilov je samotná úprava profilu, kde si užívateľ môže upraviť svoje osobné údaje zadané pri registrácii.

Upravit profil

Emailová adresa:
Jméno:
Příjmení:
Přezdívká:
Telefonní číslo:
Tag sportu: 

Upravit profil

Váš profil byl úspěšně upraven!

Staré heslo: Zobrazit heslo
Nové heslo: Zobrazit heslo
Potvrzení hesla: Zobrazit heslo

Upravit profil

Vaše heslo bylo úspěšně změněno!

Obr. 4.16: Úprava profilu

4.4.4 Rezervácia

Rezervácie majú v systéme 3 stránky, ktoré môže užívateľ vidieť.

Vytvorenie rezervácie

Po kliknutí na políčko v časovom harmonograme sa otvorí okno s rezervačným formulárom kde je potrebné vyplniť všetky údaje. Pri úspešnej rezervácii je užívateľ upozornený priamo v prehliadači, že jeho rezervácia bola vytvorená a takisto mu príde email na poskytnutú emailovú adresu s pozvánkou.

- **Prihlásený užívateľ** - vidí vo formulári už dopredu nastavené jeho osobné údaje, takisto je nastavený aj dátum spolu s časom začiatku a koncu rezervácie. Dátum a časy sú nastavené vďaka `create_reservation.js` a `start_end_time.js`. Musí tak doplniť len názov rezervácie, poprípade nejaké doplňujúce info ak chce.



Nová rezervace

Emailová adresa:

Název rezervace:

Jméno:

Příjmení:

Telefonní číslo:

Sportoviště:

Datum:

Začátek rezervace:

Konec rezervace:

Info:

Klub:

Rezervace pro klub:

Ne 05.05.2024
Tréning U11
Tréning U11
Tréning U11

Správa z webu 127.0.0.1:8000
Vaše rezervace byla vytvořena!

OK

Název rezervace:

Děkujeme za rezervaci! Těšíme sa na vás na Umelka dňa 2024-05-05 od 08:00:00 do 09:00:00 cena za vaši rezervaci je 220.00 Kč.

Jedna příloha • Skontrolované Gmailom

pozvanka.ics [Stiahnuť](#)

Obr. 4.17: Vytvorenie rezervácie pre prihláseného užívateľa

- **Klubová rezervácia** - *vedúci klubu* alebo *administrátor* môžu vytvoriť klubové rezervácie, pokiaľ zaškrtnú checkbox **Rezervácia pre klub** na to určený. Celý formulár je opäť predvyplnený s ich osobnými údajmi, musia teda len zadať názov rezervácie, zaškrtnúť checkbox a vybrať klub. Po vytvorení príde všetkým *vedúcim* a *členom klubu* email s pozvánkou.

Klub: ▾
 Rezervace pro klub:

Obr. 4.18: Vytvorenie klubovej rezervácie pre vedúceho klubu alebo administrátora

- **Neregistrovaný užívateľ** - musí do formuláru doplniť všetko potrebné, dátum spolu s časom začiatku a koncu rezervácie sú už predvyplnené. Dátum a časy sú nastavené vďaka `create_reservation.js` a `start_end_time.js`. Rezervácia je braná ako anonymná a vidí ju teda v harmonograme s farbou športoviska. To ako ju vidí *administrátor* je ukázané tu [4.4.1](#).



Nová rezervace

Emailová adresa:

Název rezervace:

Jméno:

Příjmení:

Telefonní číslo:

Sportoviště: ▾

Datum: 📅

Začátek rezervace: 🕒

Konec rezervace: 🕒

Info:

Chcete si vytvořit účet?

Ne 05.05.2024
Tréning U11
Tréning U11
Tréning U11
Tréning anonym
Tréning anonym
Tréning anonym

Obr. 4.19: Vytvorenie rezervácie pre neregistrovaného užívateľa

- **Vytvorenie účtu spolu s rezerváciou** - musí do formuláru doplniť všetko potrebné, dátum spolu s časom začiatku a koncu rezervácie sú už predvyplnené. Dátum a časy sú nastavené vďaka `create_reservation.js` a `start_end_time.js`. Ak sa rozhodne, že si chce automaticky aj vytvoriť účet, zaškrtnie checkbox **Chcete si vyvořit účet?**, ktorý mu načíta ďalší formulár na vytvorenie účtu. Po jeho doplnení je vytvorená

rezervácia, ktorú systém už neberie ako anonymnú a po prihlásení ju ten užívateľ vidí ako svoju.



Nová rezervace

Emailová adresa:

Název rezervace:

Jméno:

Příjmení:

Telefonní číslo:

Sportoviště:

Datum:

Začátek rezervace:

Konec rezervace:

Info:

Chcete si vytvorit účet?

Přezdívka*:

Tag sportu:

Souhlas s GDPR a podmínkami použití*:

Heslo*: Zobrazit heslo

Potvrzení hesla*: Zobrazit heslo

So 04.05.2024
<input type="button" value="Test aj účet"/>
<input type="button" value="Test aj účet"/>
<input type="button" value="Test aj účet"/>

Obr. 4.20: Vytvorenie účtu spolu s rezerváciou pre neregistrovaného užívateľa

Info o rezervácii

Pokiaľ si chce akýkoľvek užívateľ zistiť o cudzej vytvorenej rezervácii nejaké informácie, stačí aby klikol na ňu v harmonograme a otvorí sa mu stránka. Rozdiel však nastáva pri *neregistrovanom užívateľovi*, ktorý neuvidí osobné údaje užívateľa, ktorý tú rezerváciou vytvoril, keďže k nim nemože mať prístup podľa tabuľky 2.1.



Tréning U11

Užívateľ: belticek@gmail.com

Sportovište: Umelka

Datum a čas rezervácie: 5. května 2024: 8:00 - 9:00

Celková cena: 220,00 Kč



Tréning U11

Sportovište: Umelka

Datum a čas rezervácie: 5. května 2024: 8:00 - 9:00

Celková cena: 220,00 Kč

Obr. 4.21: Rozdiely v stránke s informáciami o cudzej rezervácii

Úprava alebo stornovanie rezervácie

Upraviť alebo stornovať svoju rezerváciu môže užívateľ, ktorý ju vytvoril a je prihlásený v systéme. Stránka ktorú užívateľ vidí je formulár s údajmi pôvodnej rezervácie. Rozdiel však nastáva pri *neregistrovanom užívateľovi*, ktorý si rezerváciou vytvoril, nemá totižto prístup k úprave ani stornovaní podľa tabuľky 2.1. Nemá teda k tejto stránke prístup. Anonymnú rezerváciu môže upravovať, alebo stornovať len *administrátor*. Po úspešnej úprave alebo stornovaní sa užívateľovi v prehliadači objaví upozornenie o ním prevedej akcii a odošle email na jeho emailovú adresu.



Tréning U11

Emailová adresa:

Název rezervace:

Křestní jméno:

Příjmení:

Telefonní číslo:

Sportoviště:

Datum:

Začátek rezervace:

Konec rezervace:

Info:

Správa z webu 127.0.0.1:8000
Vaše rezervace byla vytvořena!

Ne 05.05.2024
<input type="button" value="Tréning edit U11"/>
<input type="button" value="Tréning edit U11"/>
<input type="button" value="Tréning edit U11"/>



Tréning edit U11

Emailová adresa:

Název rezervace:

Křestní jméno:

Příjmení:

Telefonní číslo:

Sportoviště:

Datum:

Začátek rezervace:

Konec rezervace:

Info:

Správa z webu 127.0.0.1:8000
Rezervace byla stornovaná!

Ne 05.05.2024

Vaše rezervace byla upravená: Těšíme se na vás na Umelka dne 2024-05-05 od 08:00:00 do 09:00:00 cena za vaší rezervaci je 220.00 Kč.

Vaše rezervace byla stornovaná uživatelem belticek@gmail.com

Obr. 4.22: Úprava a stornovanie vlastnej rezervácie

Pri klubovej rezervácii je to rovnako, upraviť alebo zrušiť ju môže len ten kto ju vytvoril, alebo *administrátor*. Postup je rovnaký, rozdiel je v tomto:



club

Emailová adresa:

Název rezervace:

Křestní jméno:

Příjmení:

Telefonní číslo:

Sportoviště:

Datum:

Začátek rezervace:

Konec rezervace:

Info:

Klub:



edit club

Emailová adresa:

Název rezervace:

Křestní jméno:

Příjmení:

Telefonní číslo:

Sportoviště:

Datum:

Začátek rezervace:

Konec rezervace:

Info:

Klub:

Obr. 4.23: Úprava a stornovanie klubovej rezervácie

Všetky rezervácie

Táto stránka je dostupná pre všetkých užívateľov, avšak *administrátor* a *vedúci klubu* vidia všetky rezervácie v systéme roztriedené do 4 tabuliek.



[Kluby](#) [Rezervace](#) [Upravit profil](#) [Odhlásit se](#)

Mé budoucí rezervace

Název	Uživatel	Datum	Čas	Sportoviště
ide to takto?	belticek@gmail.com	9. května 2024	9:30 - 10:30	Umelka

Všechny mé rezervace

Název	Uživatel	Datum	Čas	Sportoviště
klub	belticek@gmail.com	17. dubna 2024	8:30 - 9:30	Umelka
Všetky	belticek@gmail.com	20. dubna 2024	8:30 - 9:30	Umelka
veduci viacerych	belticek@gmail.com	22. dubna 2024	13:30 - 14:30	Umelka
reg ide?	belticek@gmail.com	23. dubna 2024	17:30 - 18:30	Umelka
skuska	belticek@gmail.com	24. dubna 2024	8:30 - 9:30	Umelka

Strana 1 z 5.

[Další](#)

[Poslední »](#)

Obr. 4.24: Tabulky so všetkými mojimi rezerváciami

Budoucí rezervace

Název	Uživatel	Datum	Čas	Sportoviště
Tréning edit anonym	None	5. května 2024	10:00 - 11:00	Tenisový kurt
ide to takto?	belticek@gmail.com	9. května 2024	9:30 - 10:30	Umelka

Všechny rezervace

Název	Uživatel	Datum	Čas	Sportoviště
Skúška	admin@rezervujaspportuj.cz	15. dubna 2024	8:30 - 9:30	Umelka
klub	belticek@gmail.com	17. dubna 2024	8:30 - 9:30	Umelka
Anonym v minulosti	None	19. dubna 2024	8:30 - 11:30	Posilka
Všetky	belticek@gmail.com	20. dubna 2024	8:30 - 9:30	Umelka
lorem ipsum	xremen02@stud.fit.vutbr.cz	22. dubna 2024	8:30 - 9:30	Umelka

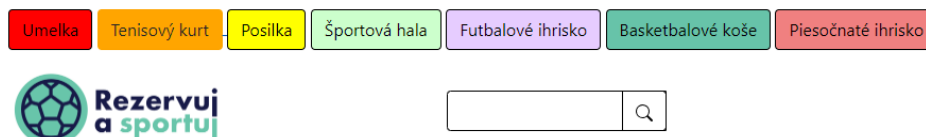
Strana 1 z 10. [Další](#) [Poslední »](#)

Obr. 4.25: Tabulky so všetkými rezerváciami

4.4.5 Športovisko

Športovisko má pre bežných užívateľov len 1 stránku, tá sa mu zobrazí po stlačení tlačidla s názvom športoviska.

Info o športovisku



Info o sportovišti - Tenisový kurt

Otvárací doba: 8:00 - 20:00

Kapacita: 4

Adresa: Kurtová 1

Cena za hodinu: 140,00

Tag sportovište: #kurt

Obr. 4.26: Info o športovisku

Ostatné 2 sú len pre *administrátora* a sú ukázané v 4.4.7.

4.4.6 Klub

Táto podsekcia sa venuje stránkam o klube, postupne si predstavíme celý postup, ktorý je znázornený v diagraoch C.4 a C.3. Existujú samozrejme aj spôsoby ako vytvoriť a spravovať klub pokiaľ ho chce vytvoriť *neregistrovaný užívateľ*, tie sú naznačené v diagramoch C.5, C.2 a C.1. Tie stránky sú ale rovnaké, líšia sa len vo vytvorení užívateľa a teda nebudú ukázané.

1. **Všetky kluby** - stránka kde **registrovaní užívatelia** vidia všetky schválené kluby, ku ktorým sa môžu pridať, rozdiel je pri *administrátorovi*, ten vidí úplne všetky kluby, aj tie, čakajúce na schválenie.

Všechny kluby

Název klubu	Tag týmu	Vedoucí týmu	Akce
FC Medlánky	#medlankyA	belticek@gmail.com	<input type="button" value="Poslat žádost o členství"/>
FK Iskra Hnúšťa	#iskraA	belticek@gmail.com	<input type="button" value="Poslat žádost o členství"/>

Noví vedoucí mého klubu

Název klubu	Tag týmu	Vedoucí na schválení	Akce
-------------	----------	----------------------	------

Noví členové mého klubu

Název klubu	Tag týmu	Členové na schválení	Akce
-------------	----------	----------------------	------

Obr. 4.27: Stránka klubov pre registrovaného užívateľa

Všechny kluby

Název klubu	Tag týmu	Vedoucí týmu	Akce
FC Medlánky	#medlankyA	belticek@gmail.com	<button>Odstranit klub</button>
FK Iskra Hnúšťa	#iskraA	belticek@gmail.com	<button>Odstranit klub</button>

Noví členové klubu

Název klubu	Tag týmu	Členové na schválení	Akce
-------------	----------	----------------------	------

Noví vedoucí klubu

Název klubu	Tag týmu	Vedoucí na schválení	Akce
-------------	----------	----------------------	------

Nové kluby

Název klubu	Tag týmu	Vedoucí týmu	Akce
-------------	----------	--------------	------

Vytvořit nový klub

Obr. 4.28: Stránka klubov pre administrátora

2. **Vytvorenie nového klubu** - stránka s formulárom na vytvorenie nového klubu, ktorý musí *administrátor* schváliť, **registrovaný užívateľ** má pri vedúcom predvyplnený svoj email, *administrátor* má na výber zo všetkých **užívateľov** v systéme.



Rezervuj a sportuj

Vytvořit nový klub

Název:

Tag tímu:

Vedoucí klubu:

Vytvořit klub

Správa z webu localhost:8000
Novej klub a jeho vedoucí byli vytvořeny a čekaj na schválení!

OK

Rezervuj a sportuj

Vytvořit nový klub

Název:

Tag tímu:

Vedoucí klubu:

Vytvořit klub

Správa z webu localhost:8000
Novej klub a jeho vedoucí byli vytvořeny a čekaj na schválení!

OK

Váš klub FK VUTBR byl vytvořen a čeká na schválení!

Váš klub FC FIT VUT byl vytvořen a čeká na schválení!

Obr. 4.29: Rozdiely pri vytváraní nového klubu

3. **Schválenie nového klubu** - tlačidlo na schválenie nového klubu a jeho prvého vedúceho je dostupné len pre *administrátora*, ako je definované aj v tabuľke 2.1.

Nové kluby

Název klubu	Tag tímu	Vedoucí tímu	Akce
FK VUTBR	#fkvutbr	xremen02@vutbr.cz	Schválit nový klub a jeho vedoucího
FC FIT VUT	#fcfitvut	xremen02@stud.fit.vutbr.cz	Schválit nový klub a jeho vedoucího

Obr. 4.30: Schválenie nového klubu

4. **Poslanie alebo zrušenie žiadostí na pridanie do klubu - registrovaný užívateľ** môže odoslať žiadosť na pridanie do akéhokoľvek schváleného klubu, takisto ju ale môže aj zrušiť. Jedná sa o tlačidlá, odosielajúce formuláre.

FK VUTBR	#fkvutbr	xremen02@vutbr.cz	Poslat žiadosť o členství
FK VUTBR	#fkvutbr	xremen02@vutbr.cz	Zrušit žiadosť na přidání do klubu

Obr. 4.31: Poslanie alebo zrušenie žiadosti o pridanie do klubu

5. **Schválenie nového vedúceho** - už existujúci *vedúci klubu* môže schváliť žiadosť na pridanie do svojho klubu. Jedná sa o tlačidlo, odosielajúce formulár.

Noví členové mého klubu

Název klubu	Tag tímu	Členové na schválení	Akce
FK VUTBR	#fkvutbr	belticek@gmail.com	<input type="button" value="Schválit nového člena"/>

Obr. 4.32: Schválení žádosti o přidání do klubu

6. **Poslání nebo zrušení žádosti na nového vedoucího klubu** - *športovec* může odeslat žádost na stání sa *novým vedoucím* svojho klubu, ktorého je členom, takisto ju ale opäť môže aj zrušiť. Jedná sa o tlačidlá, odosielajúce formuláre.

FK VUTBR	#fkvutbr	xremen02@vutbr.cz	<input type="button" value="Poslat žádost na vedoucího"/> <input type="button" value="Opustit klub"/>
FK VUTBR	#fkvutbr	xremen02@vutbr.cz	<input type="button" value="Zrušiť žádost na vedoucího"/>

Obr. 4.33: Poslání nebo zrušení žádosti o přidání do klubu

7. **Schválení nového vedoucího** - už existující *vedoucí klubu* může schválit žádost člena na stání sa *novým vedoucím klubu*. Jedná sa o tlačidlo, odosielajúce formulár.

Noví vedúci mého klubu

Název klubu	Tag tímu	Vedúci na schválení	Akce
FK VUTBR	#fkvutbr	belticek@gmail.com	Schválit nového vedoucího

Obr. 4.34: Schválení žádosti o přidání do klubu

8. **Opustenie klubu** - *športovec* môže opustiť klub a tým sa jeho členstvo vymaže z databázy. Jedná sa o tlačidlo, odosielajúce formulár.

FK VUTBR	#fkvutbr	xremen02@vutbr.cz	Poslat žádost na vedoucího	Opustiť klub
----------	----------	-------------------	----------------------------	--------------

Obr. 4.35: Opustenie klubu

9. **Opustenie alebo odstránenie klubu vedúcim klubu** - už existujúci *vedúci klubu* môže takisto opustiť klub, čím sa klub automaticky vymaže, alebo ho môže aj rovno vymazať. Jedná sa o tlačidlá, odosielajúce formuláre.

FK VUTBR	#fkvutbr	xremen02@vutbr.cz	Odstranit klub	Opustiť klub
FK VUTBR	#fkvutbr	xremen02@vutbr.cz	Odstranit klub	Opustiť klub

Obr. 4.36: Opustenie alebo odstránenie klubu vedúcim klubu

Všetky tie schválenia a odstránenia môže robiť aj *administrátor* a o každej jednej zmene stavu **klubu**, alebo žiadostí je **užívateľ** informovaný emailom na jeho emailovej adrese.

10. **Info o klube** - každý **užívateľ** má prístup k tejto stránke, ale každý typ má trochu iný pohľad.

- **administrátor** - môže upraviť info o klube, takisto môže odstrániť členov alebo vedúcich.



FK VUTBR

Název klubu:

Tag klubu:

Vedúci klubu

Email	Celé jméno	Akce
xremen02@vutbr.cz	Test Remeň	<input type="button" value="Odstranit vedoucího"/>
belticek@gmail.com	Juraj Remeň	<input type="button" value="Odstranit vedoucího"/>

Strana 1 z 1.

Členové klubu

Email	Celé jméno	Akce
xremen02@stud.fit.vutbr.cz	Anton Berrolák	<input type="button" value="Odstranit členu"/>

Strana 1 z 1.

- **vedúci klubu** - môže upraviť info o klube, takisto môže odstrániť členov alebo vedúcich, ale nemôže odstrániť samého seba.



FK VUTBR

Název klubu:

Tag klubu:

Vedoucí klubu

Email	Celé jméno	Akce
xremen02@vutbr.cz	Test Remeň	
belticek@gmail.com	Juraj Remeň	<input type="button" value="Odstranit vedoucího"/>

Strana 1 z 1.

Členové klubu

Email	Celé jméno	Akce
xremen02@stud.fit.vutbr.cz	Anton Berrolák	<input type="button" value="Odstranit člena"/>

Strana 1 z 1.

- **člen klubu** - vidí všechno info o klube vrátane jeho vedúcich a členov.



FK VUTBR

Název klubu: FK VUTBR

Tag klubu: #fkvutbr

Vedoucí klubu

Email	Celé jméno
xremen02@vutbr.cz	Test Remeň
belticek@gmail.com	Juraj Remeň

Strana 1 z 1.

Členové klubu

Email	Celé jméno
xremen02@stud.fit.vutbr.cz	Anton Berrolák

Strana 1 z 1.

- **neregistrovaný užívateľ** - vidí len názov a atag klubu.



FK VUTBR

Název klubu: FK VUTBR

Tag klubu: #fkvutbr

Obr. 4.37: Rozdiely pri info o klube

4.4.7 Admin

Niektoré stránky, ako bolo spomenuté v predchádzajúcich podsekciami, sú k dispozícii len pre *administrátora* a sú to tieto:

- **Administrátorská správa užívateľov** - jedná sa o stránku, obsahujúcu odstránenú tabuľku všetkých registrovaných užívateľov v systéme, obsahuje aj tlačidlá na odstránenie užívateľa alebo na aktiváciu účtu, v prípade, že niekomu z nejakých dôvodov neprišiel aktivačný email.

xremen02@vutbr.cz	Test Remeň	Odstranit uživatele	Aktivovat účet
-------------------	------------	---------------------	----------------

Obr. 4.38: Administrátorská správa uživatelův

- **Pridanie športoviska** - jedná sa o stránku, obsahujúcu formulár na vytvorenie nového športoviska, po vyplnení všetkých údajov sa objaví v prehliadači upozornenie o vytvorení nového športoviska.



Nové sportoviště

Název:

Otevřeno od:

Zavřeno od:

Adresa:

Tag:

Kapacita:

Cena za hodinu:

Info:


Barva:

Správa z webu 127.0.0.1:8000

Nové sportoviště bylo vytvořené!

Obr. 4.39: Pridanie nového športoviska

- **Upravenie alebo vymazanie športoviska** - jedná sa o stránku, obsahujúcu formulár na úpravu informácií o športovisku, po úprave všetkých údajov alebo vymazaní sa objaví v prehliadači upozornenie o upravení alebo vymazaní. Takisto sa automaticky odošlú informačné emaily všetkým užívateľom, ktorí mali na tom športovisku vytvorenú rezerváciu.



Info o sportovišti - Kajakáreň

Název:

Otevřeno od:

Zavřeno od:

Adresa:

Tag:

Kapacita:

Cena za hodinu:

Info:

Barva:

Info o sportovišti - TRX posilka

Název:

Otevřeno od:

Zavřeno od:

Adresa:

Tag:

Kapacita:

Cena za hodinu:

Info:

Barva:

Správa z webu 127.0.0.1:8000

Sportovište bylo upravené!

Správa z webu 127.0.0.1:8000

Sportovište bylo vymazané a s ním i všechny na něj vytvořené rezervace!

Obr. 4.40: Úprava alebo vymazanie športoviska

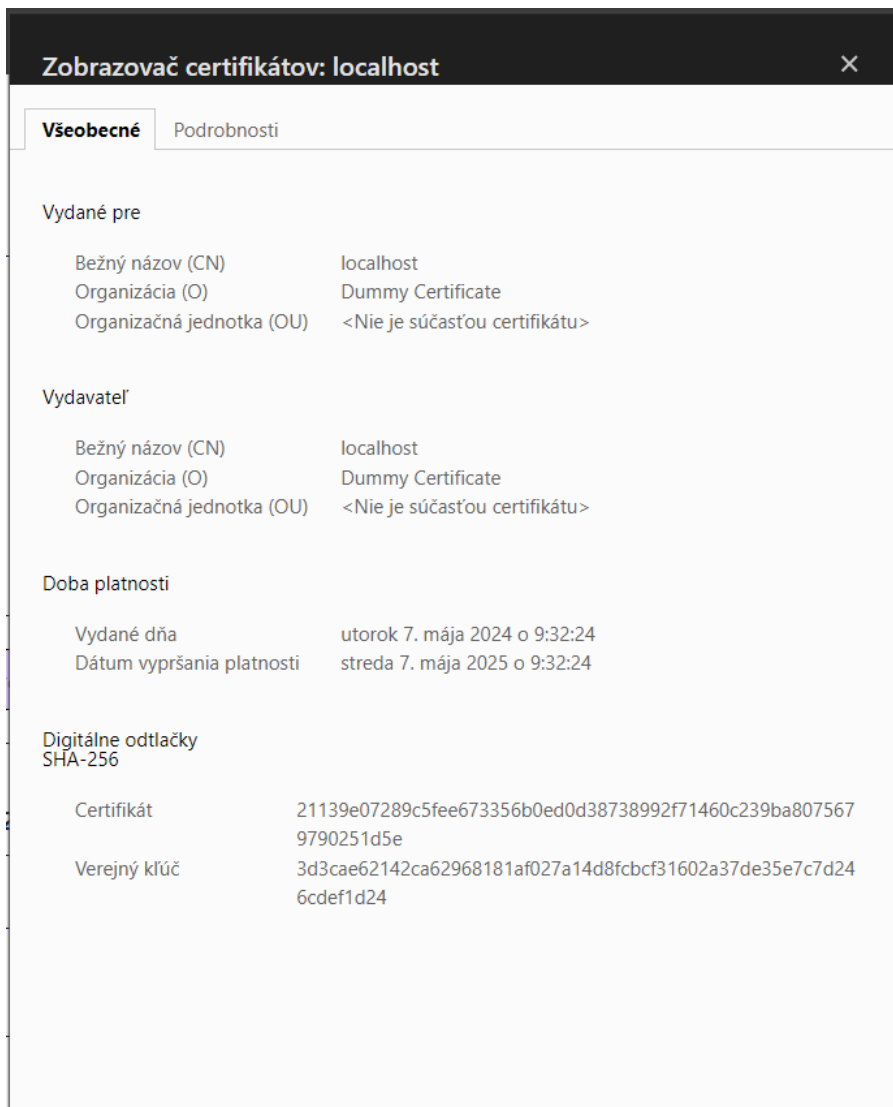
4.5 Automatické nasadenie systému

Jednou z podmienok splnenia zadania je automatické nasadenie systému. Táto podmienka vznikla hlavne na základe požiadavky k znovupoužitelnosti, prenositeľnosti a bezpečnosti. Systém vďaka tejto skutočnosti je možné použiť naozaj kdekoľvek v prípade, že sa jedná o aplikovanie systému na malé alebo stredné športové kluby, ktoré si chcú uľahčiť prácu. Vybranou technológiou pre splnenie tejto podmienky je **Docker 4.1.4**.

Na nasadenie sú použité 2 súbory `Dockerfile` a `docker-compose.yml`, ktoré sú v hlavnom priečinku celého projektu. Celý projekt je ako **Docker image** dostupný priamo na **Docker hube**, pre všetkých kto by si chcel systém vyskúšať.

4.5.1 Certifikácia

Na účely splnenia podmienky zadania o bezpečnosti je pri spustení serveru použitý samopodpísaný certifikát.



Obr. 4.41: Samopodpísaný certifikát použitý v systéme

Samopodpísané certifikáty sú užitočné pre vývojové a testovacie účely, ako aj pre interné aplikácie, kde nevyžadujete dôveru z verejného internetu. Ich hlavnou výhodou je jednoduchosť a možnosť ich rýchleho vytvorenia bez potreby zapojenia certifikačnej autority (CA). Tu sú niektoré kľúčové aspekty používania samopodpísaných certifikátov:

- **Rýchle vytvorenie** - proces vytvorenia samopodpísaného certifikátu je rýchly a jednoduchý. Bez nutnosti žiadania o certifikáciu u CA môžete okamžite vygenerovať certifikát a začať ho používať. Lokálny Vývoj a Testovanie: Samopodpísané certifikáty sú ideálne pre lokálny vývoj a testovanie webových aplikácií. Pomáhajú zabezpečiť komunikáciu medzi klientmi a serverom na vašom lokálnom počítači bez nutnosti získavania certifikátu od externých CA.

- **Nízke náklady** - používanie samopodpísaných certifikátov nemá náklady spojené s nákupom certifikátov od externých CA. To ich robí atraktívnou voľbou pre projekty s obmedzeným rozpočtom.
- **Obmedzená dôvera** - samopodpísané certifikáty nie sú dôveryhodné z pohľadu verejnosti. Preto sa neodporúčajú na použitie v produkčných prostrediach, kde je potrebná vysoká úroveň dôveryhodnosti.
- **Obmedzené bezpečnostné funkcie** - samopodpísané certifikáty neposkytujú externé overenie identity servera. Z tohto dôvodu nemajú rovnakú úroveň zabezpečenia ako certifikáty vydané renomovanými CA.

V produkčných prostrediach je odporúčané používať certifikáty od dôveryhodných CA, ako je napríklad Let's Encrypt. Let's Encrypt poskytuje bezplatné certifikáty od renomovanej certifikačnej authority, ktoré sú dôveryhodné pre verejnosť a poskytujú rovnakú úroveň zabezpečenia ako platené certifikáty. Používanie certifikátov od Let's Encrypt je tak ideálnou voľbou pre produkčné nasadenie webových aplikácií.

Kapitola 5

Testovanie

Testovanie je kľúčovým prvkom v mnohých odvetviach, od softvérového vývoja až po vedecký výskum. Jeho cieľom je overiť správnosť, spoľahlivosť a výkonnosť rôznych systémov, procesov alebo produktov. Existuje mnoho typov testovania, vrátane manuálneho a automatizovaného testovania, black-box a white-box testovania, funkčného a nefunkčného testovania atď. V zadaní mojej práce je spomenuté automatické testovanie.

Testovanie softvéru je nevyhnutné pre zabezpečenie kvality a správnej funkčnosti aplikácií a programov. To zahŕňa overenie, či softvér spĺňa špecifikácie a požiadavky, ako aj testovanie jeho odolnosti voči rôznym situáciám a podmienkam.

Celkovo je testovanie nevyhnutnou súčasťou vývoja a prevádzky systémov vo všetkých oblastiach, či už ide o softvér, zdravotnú starostlivosť, vedu alebo bezpečnosť. Efektívne testovanie pomáha zabezpečiť spoľahlivosť, bezpečnosť a výkonnosť rôznych systémov a procesov, čo má pozitívny vplyv na používateľov.

5.1 Automatické testy

Automatické testy softvéru sú základným kameňom pre zabezpečenie jeho spoľahlivosti a kvality. Tieto testy sú navrhnuté tak, aby sa vykonávali bez ľudskej interakcie, čo umožňuje opakované a konzistentné overenie funkčnosti softvéru.

Existuje niekoľko typov automatických testov, ktoré sa používajú na overenie rôznych aspektov softvéru:

- **Jednotkové testy** - overujú jednotlivé komponenty softvéru, často nazývané ako jednotky. Tieto testy sa vykonávajú na izolovaných častiach kódu a kontrolujú ich správne fungovanie.
- **Integračné testy** - overujú spoluprácu medzi viacerými komponentmi softvéru. Cieľom je zistiť, či tieto komponenty správne komunikujú a integrujú sa navzájom.
- **Funkcionálne testy** - overujú, či softvér vykonáva požadované funkcie správne. Tieto testy sa zameriavajú na koncové výsledky, ktoré používatelia očakávajú.
- **Výkonnostné testy** - merajú výkon softvéru a jeho schopnosť pracovať s rôznymi množstvami dát a záťažami. Cieľom je zistiť, ako dobre softvér zvláda požiadavky v reálnom prostredí.
- **Akceptačné testy** - overujú, či softvér spĺňa požiadavky a očakávania zákazníkov. Tieto testy sú často vykonávané zákazníkmi alebo užívateľmi softvéru.

Pri správnom nasadení automatických testov sa softvér stáva robustnejším a menej náchylným na chyby. Ich účinnosť je však závislá od kvality testovacieho scenára a správnej implementácie. Preto je dôležité, aby sa vývojári starali nielen o písanie kódu, ale aj o tvorbu a údržbu testovacieho prostredia.

Pre vizualizáciu jednotlivých fáz implementácie alebo testovania sa používajú rôzne modely, ten najzákladnejší je **V Model** [2].

5.1.1 Backend testy jednotlivých aplikácií

Automatické testy, ktoré sú súčasťou každej jednej aplikácie a nachádzajú sa v priečinkoch aplikácií sú práve **jednotkové testy**. Jedná sa o súbory `tests.py`, ktoré testujú a overujú jednotlivé komponenty systému. Za účelom sprehľadnenia pokrytia požiadaviek testami vznikli tabuľky pokrytia. Na spustenie všetkých testov naraz slúži skript `run_tests.sh`.

Test	Vyhľadávanie	Vedúci klubu	História	Športovisko	Rezervácia	Administrátor	Email	Kalendár	Tagy	Kluby	Neregistrovaní užívatelia	Prenositeľnosť	Udržateľnosť	Bezpečnosť	Registrovaní užívatelia
1.1		✓				✓			✓	✓	✓				✓
1.2		✓				✓	✓		✓	✓					✓
1.3	✓	✓				✓	✓			✓	✓				✓
1.4		✓	✓		✓	✓	✓		✓	✓					✓
1.5		✓	✓		✓	✓	✓		✓	✓					
1.6		✓	✓		✓	✓	✓		✓	✓					
1.7			✓			✓	✓		✓	✓					
1.8		✓			✓	✓	✓			✓					
1.9		✓			✓	✓	✓		✓	✓					
1.10		✓				✓				✓					✓
1.11		✓				✓				✓					✓
1.12		✓	✓		✓	✓	✓		✓	✓					✓
1.13		✓	✓		✓	✓	✓		✓	✓					✓
1.14		✓				✓	✓			✓					
1.15		✓	✓		✓	✓	✓		✓	✓					
1.16		✓				✓			✓	✓		✓	✓		
1.17	✓	✓				✓			✓	✓		✓	✓		

Tabuľka 5.1: Tabuľka pokrytia požiadaviek testami aplikácie **club**

Test	Názov
1.1	ClubViewTestCase
1.2	LeaveClubViewTestCase
1.3	ClubInfoViewTestCase
1.4	CreateClubViewTestCase
1.5	DeleteClubViewTestCase
1.6	EditClubViewTestCase
1.7	ApproveClubAndItsManagerViewTestCase
1.8	ApproveNewMemberViewTestCase
1.9	ApproveNewManagerViewTestCase
1.10	SendRequestToJoinClubTestCase
1.11	CancelRequestToJoinClubTestCase
1.12	SendRequestToBecomeNewManagerViewTestCase
1.13	CancelRequestToBecomeNewManagerViewTes
1.14	DeleteMemberViewTes
1.15	DeleteManagerViewTes
1.16	ClubFormsTests
1.17	ClubModelsTests

Tabuľka 5.2: Názvy testov aplikácie **club**

Test	Vyhľadavanie	Vedúci klubu	História	Športovisko	Rezervácia	Administrátor	Email	Kalendár	Tagy	Kluby	Neregistrovaní užívatelia	Prenositelnosť	Udržateľnosť	Bezpečnosť	Registrovaní užívatelia
2.1		✓	✓	✓	✓	✓	✓	✓		✓	✓				✓
2.2		✓	✓	✓	✓	✓	✓	✓		✓	✓				✓
2.3		✓	✓	✓	✓	✓	✓	✓		✓	✓				✓
2.4						✓							✓		
2.5					✓	✓	✓		✓	✓			✓		
2.6					✓	✓			✓	✓			✓		
2.7		✓	✓		✓	✓			✓	✓					✓
2.8		✓	✓		✓	✓	✓		✓	✓					✓
2.9	✓	✓	✓	✓	✓	✓			✓	✓	✓				✓

Tabuľka 5.3: Tabuľka pokrytia požiadaviek testami aplikácie **home**

2.1	HomepageViewTestCase
2.2	NavigateToNextWeekViewTestCase
2.3	NavigateToPreviousWeekViewTestCase
2.4	UsersViewTestCase
2.5	DeleteUserViewTestCase
2.6	ActivateUserViewTestCase
2.7	ReservationsViewTestCase
2.8	ClubsViewTestCase
2.9	SearchViewTestCase

Tabuľka 5.4: Názvy testov aplikácie **home**

Test	Vyhľadavanie	Vedúci klubu	História	Športovisko	Rezervácia	Administrátor	Email	Kalendár	Tagy	Kluby	Neregistrovaní užívatelia	Prenositelnosť	Udržateľnosť	Bezpečnosť	Registrovaní užívatelia
3.1		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
3.2		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓
3.3		✓	✓	✓	✓	✓	✓	✓	✓	✓				✓	✓
3.4				✓	✓		✓							✓	
3.5		✓				✓			✓	✓		✓	✓		
3.6	✓	✓				✓			✓	✓		✓	✓		

Tabuľka 5.5: Tabuľka pokrytia požiadaviek testami aplikácie **reservation**

3.1	ReservationViewTestCase
3.2	ReservationInfoViewTestCase
3.3	EditReservationInfoViewTestCase
3.4	TestReservationValidator
3.5	ReservationFormsTests
3.6	ReservationModelsTests

Tabuľka 5.6: Názvy testov aplikácie **reservation**

Test	Vyhľadávanie	Vedúci klubu	História	Športovisko	Rezervácia	Administrátor	Email	Kalendár	Tagy	Kluby	Neregistrovaní užívatelia	Prenositelnosť	Udržateľnosť	Bezpečnosť	Registrowaní užívatelia
5.1														✓	✓
5.2						✓	✓		✓		✓			✓	
5.3							✓							✓	
5.4									✓					✓	✓
5.5														✓	
5.6			✓		✓		✓		✓					✓	✓
5.7							✓							✓	✓
5.8							✓							✓	✓
5.9							✓							✓	✓
5.10		✓				✓			✓	✓		✓	✓		
5.11	✓	✓				✓			✓	✓		✓	✓		

Tabuľka 5.9: Tabuľka pokrytia požiadaviek testami aplikácie **users**

Test	Názov
5.1	LoginViewTestCase
5.2	SignupViewTestCase
5.3	AccountActivationViewTestCase
5.4	ProfileViewTestCase
5.5	LogoutViewTestCase
5.6	DeleteAccountViewTestCase
5.7	ForgottenPasswordViewTestCase
5.8	ResetPasswordViewTestCase
5.9	EmailAuthBackendTest
5.10	UsersFormsTests
5.11	UsersModelsTests

Tabuľka 5.10: Názvy testov aplikácie **users**

Kapitola 6

Záver

Cieľom práce bolo vytvoriť ľahko udržiavateľný, bezpečný a prenositeľný systém na rezerváciu športovísk pre TJ Ochoz u Brna. Systém, ale nie je obmedzený len na tento konkrétny športový klub, je nasaditeľný aj na podobné malé alebo stredné športové kluby po celom Česku alebo Slovensku. V prípade potreby je možné ho poupraviť aj na rezerváciu istých služieb. Systém je schopný vytvoriť rezerváciu pre akýkoľvek typ užívateľa, teda aj pre *neregistrovaného*, čo bolo jednou z hlavných požiadaviek. Súčasne používaný systém v klube to neumožňoval. Systém spĺňa všetky potrebné požiadavky a takisto podmienky zadania a je možné ho reálne teda naozaj využívať.

Testovanie prebehlo hlavne na mnou pripravených dátach, ktoré sa pridávali do systému. Počas jednej z konzultácií mi bol predstavený softvér [26], naimplementovaný práve mojím vedúcim práce, ktorý by mi tie mnou potrebné dáta pripravil podľa mojich požiadaviek a dodalo by to na vierohodnosti celého testovania a aj systému samotného. Rozhodol som sa ho, ale nevyužiť vzhľadom na to, že vďaka kompletnej funkcionalite sa jednotlivé dáta do systému vkladali jednoducho.

Vytvorenie systému nebolo vôbec ľahké vzhľadom na prvotné problémy s predstavou a následnou implementáciou, ktorá nebola ideálna ani podľa mňa a ani podľa vedúceho práce. Takisto z mojej strany občas viazla komunikácia, alebo odkomunikovanie istej funkcionality a na následných konzultáciách sa zistilo, že to v podstate ani nebolo potrebné urobiť.

Práca na systéme však určite nie je kompletne hotová, napadá mi množstvo vylepšení a rozšírení, ktoré by sa dali do systému zapracovať. Mohlo by ísť napríklad o opakujúcu sa rezerváciu, ktorá by určite zväčšila efektívnosť používania systému. Beriem tento prvotný systém za tzv. **Rezervuj a sportuj 1.0**, kde sa postupne môže pridávať funkcionalita a môžu sa teda vytvárať novšie a lepšie verzie. Práca samozrejme bude pokračovať aj vďaka spätnej väzbe po nasadení do reálneho života, kde sa samotní užívatelia budú vyjadrovať akú funkcionalitu by v novších verziách uvítali, samozrejme so zachovaním pôvodných podmienok, ktoré boli dané pri analýze 2.

Na záver by som ešte raz chcel poďakovať všetkým ľuďom, ktorí mi boli oporou pri tvorbe tejto bakalárskej práce.

Literatúra

- [1] ARORA, N. *Django Project MVT Structure*. 2021 [cit. 2024-04-24]. Dostupné z: <https://www.geeksforgeeks.org/django-project-mvt-structure/>.
- [2] AUTOMATION, S. T. T. . *ISTQB V Model*. 2018 [cit. 2024-05-06]. Dostupné z: <https://www.software-testing-tutorials-automation.com/2018/10/istqb-v-model.html>.
- [3] BOOTSTRAP CONTRIBUTORS. *Bootstrap*. [cit. 2024-04-30]. Dostupné z: <https://getbootstrap.com/>.
- [4] CHEN, P. P.-S. The Entity-Relationship Model - Toward a Unified View of Data. *ACM Transactions on Database Systems (TODS)*. 1. vyd. ACM. 1976, č. 1, s. 9–36, [cit. 2024-04-24].
- [5] CONTRIBUTORS, D. *Built-in template tags and filters*. 2023 [cit. 2024-05-04]. Dostupné z: <https://docs.djangoproject.com/en/5.0/ref/templates/builtins/>.
- [6] CONTRIBUTORS, D. *Creating forms from models*. 2023 [cit. 2024-05-03]. Dostupné z: <https://docs.djangoproject.com/en/5.0/topics/forms/modelforms/>.
- [7] CONTRIBUTORS, D. *Customizing authentication in Django*. 2023 [cit. 2024-05-01]. Dostupné z: <https://docs.djangoproject.com/en/5.0/topics/auth/customizing/#authentication-backends>.
- [8] CONTRIBUTORS, D. *Customizing authentication in Django*. 2023 [cit. 2024-05-01]. Dostupné z: <https://docs.djangoproject.com/en/5.0/topics/auth/default/#django.contrib.auth.authenticate>.
- [9] CONTRIBUTORS, D. *Django Utils*. 2023 [cit. 2024-05-03]. Dostupné z: <https://docs.djangoproject.com/en/5.0/ref/utils/#module-django.utils.encoding>.
- [10] CONTRIBUTORS, D. *Django Utils*. 2023 [cit. 2024-05-03]. Dostupné z: <https://docs.djangoproject.com/en/5.0/ref/utils/#module-django.utils.http>.
- [11] CONTRIBUTORS, D. *Formsets*. 2023 [cit. 2024-05-03]. Dostupné z: <https://docs.djangoproject.com/en/5.0/topics/forms/formsets/>.
- [12] CONTRIBUTORS, D. *Using the Django authentication system*. 2023 [cit. 2024-05-01]. Dostupné z: <https://docs.djangoproject.com/en/5.0/topics/auth/default/#django.contrib.auth.forms.SetPasswordForm>.
- [13] CONTRIBUTORS, W. *HTML Living Standard*. 2024 [cit. 2024-04-30]. Dostupné z: <https://html.spec.whatwg.org/multipage/>.

- [14] DJANGO-PHONENUMBER-FIELD CONTRIBUTORS. *Django-phonenumbers-field Documentation*. 2022 [cit. 2024-04-30]. Dostupné z: <https://django-phonenumbers-field.readthedocs.io/en/latest/>.
- [15] DOCKER. *Dockerfile reference*. 2024 [cit. 2024-04-30]. Dostupné z: <https://docs.docker.com/reference/dockerfile/>.
- [16] DOCKER CONTRIBUTORS. *Docker*. [cit. 2024-04-30]. Dostupné z: <https://www.docker.com/>.
- [17] FOUNDATION, D. S. *Writing your first Django app, part 2* [online]. 2023 [cit. 2024-04-24]. Dostupné z: <https://docs.djangoproject.com/en/5.0/intro/tutorial02/#creating-an-admin-user>.
- [18] FOUNDATION, D. S. *Django: The Web framework for perfectionists with deadlines*. No date [cit. 2024-04-30]. Dostupné z: <https://www.djangoproject.com/>.
- [19] GAMMA, E., HELM, R., JOHNSON, R. a VLISSIDES, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. 1. vyd. Addison-Wesley, 1995. ISBN 978-0201633610.
- [20] GITHUB. *The state of open source and AI*. 2023 [cit. 2024-04-30]. Dostupné z: <https://github.blog/2023-11-08-the-state-of-open-source-and-ai/>.
- [21] GRIŠKĚNAS, S. *SHA-256: What It Is and How It Works*. 2023 [cit. 2024-04-30]. Dostupné z: <https://nordvpn.com/blog/sha-256/>.
- [22] INVERITASOFT. *The Best Web Application Development Languages*. 2023 [cit. 2024-04-30]. Dostupné z: <https://inveritasoft.com/blog/the-best-web-application-development-languages>.
- [23] MOZILLA CONTRIBUTORS. *JavaScript*. 2024 [cit. 2024-04-30]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>.
- [24] ORACLE. *Čo je to databáza?* 2021 [cit. 2024-04-24]. Dostupné z: <https://www.oracle.com/cz/database/what-is-database/>.
- [25] OVERFLOW, S. *Stack Overflow Developer Survey 2023*. 2023 [cit. 2024-04-30]. Dostupné z: <https://survey.stackoverflow.co/2023/#most-popular-technologies-language-prof>.
- [26] PH.D., I. A. S. 2024 [cit. 2024-05-06]. Dostupné z: <https://dataster.testos.org/>.
- [27] POSTMAN. *What are HTTP Methods?* 2023 [cit. 2024-05-01]. Dostupné z: <https://blog.postman.com/what-are-http-methods/>.
- [28] VETTIVEL, N. *PBKDF2 Hashing Algorithm*. 2021 [cit. 2024-04-30]. Dostupné z: <https://nishothan-17.medium.com/pbkdf2-hashing-algorithm-841d5cc9178d>.
- [29] (W3C), W. W. W. C. *Cascading Style Sheets (CSS)*. 2024 [cit. 2024-04-30]. Dostupné z: <https://www.w3.org/TR/CSS/#css>.
- [30] YIGAL, A. *SQLite vs. MySQL vs. PostgreSQL: A Comparison of Relational Databases*. [cit. 2024-04-24]. Dostupné z: <https://logz.io/blog/relational-database-comparison/>.

Príloha A

Obsah priloženého pamäťového média

- reservation_system/ - priečinok so zdrojovými súbormi
- docs/ - priečinok so súbormi dokumentácie
- thesis.pdf - súbor záverečnej práce
- README.md - README súbor pre projekt
- latex/ - priečinok so zdrojovými súbormi \LaTeX

Príloha B

Inštalácia, konfigurácia a spustenie

Všetky informácie môžete nájsť aj v súbore README.md, ktorý sa nachádza v koreňovom priečinku projektu. Všetky príkazy na inštaláciu je nutné robiť v priečinku `reservation_system/`.

Požiadavky

- **Docker** - ak sa rozhodnete pre toto použitie alebo
- **Python3.10** a **pipenv** - klasické použitie

Inštalácia

- [Docker](#) - ktorý nainštaluje všetko potrebné vrátane **Python3.10** a **pipenv** a všetkých potrebných knižníc. alebo
- [Python3.10](#) a [pipenv](#)

Konfigurácia

Pre správne fungovanie celého systému je nutné vytvoriť súbor `.env`, v ktorom treba nadefinovať potrebné hodnoty, konkrétne:

- `DEBUG=on`
- `EMAIL_HOST=your_email_host`
- `EMAIL_HOST_USER=your_email`
- `EMAIL_HOST_PASSWORD=your_email_password`
- `EMAIL_PORT=your_email_port`
- `EMAIL_USE_TLS=True`

Spustenie

V prípade, že sa rozhodnete pre **Docker** stačí zadať príkaz `docker-compose up` a navštíviť [Rezervuj a sportuj](#).

V prípade klasického spustenia je nutné použiť príkaz `run python3.10 manage.py runserver_plus --cert-file certs/cert.crt` a navštíviť [Rezervuj a sportuj](#).

Prihlásenie administrátora

- **Emailová adresa** - *admin@rezervujaspportuj.cz*
- **Heslo** - *admin*

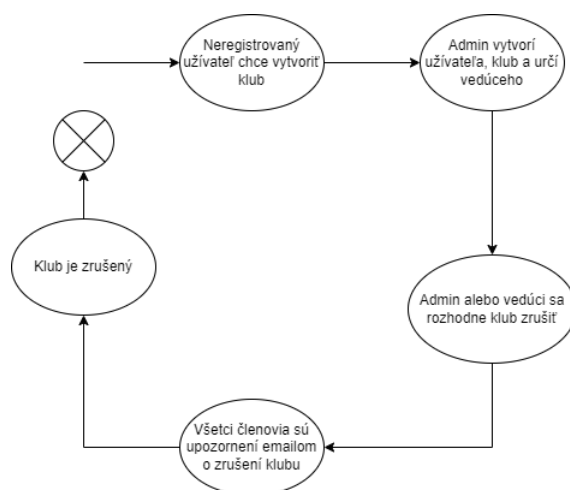
Prihlásenie užívateľa

- **Emailová adresa** - *xremen02@vutbr.cz*
- **Heslo** - *#PasSW0rD_3601*

Príloha C

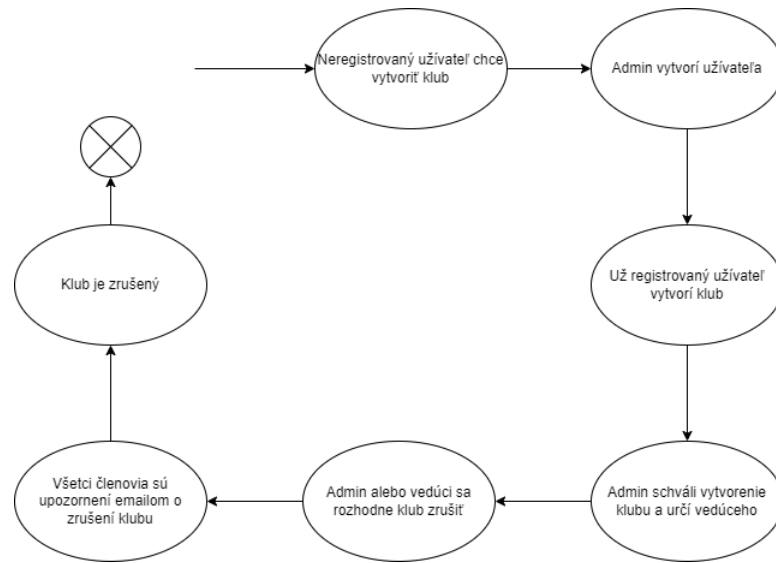
Diagramy

Životný cyklus vytvorenia klubu administrátorom



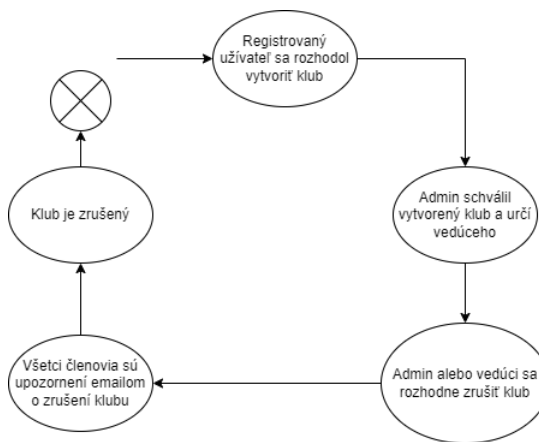
Obr. C.1: Životný cyklus vytvorenia klubu administrátorom

Životný cyklus vytvorenia klubu neregistrovaným užívateľom



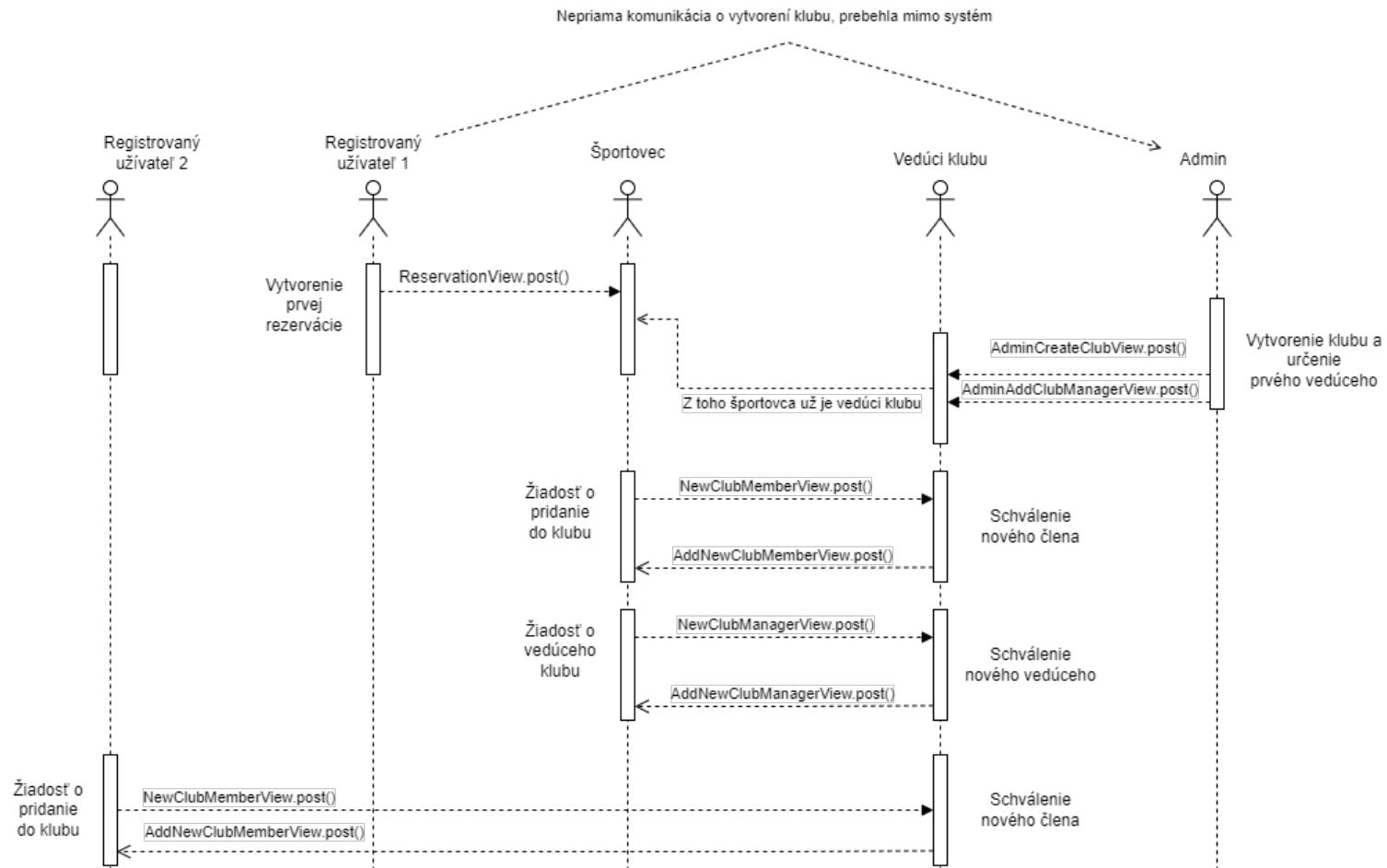
Obr. C.2: Životný cyklus vytvorenia klubu neregistrovaným užívateľom

Životný cyklus vytvorenia klubu registrovaným užívateľom



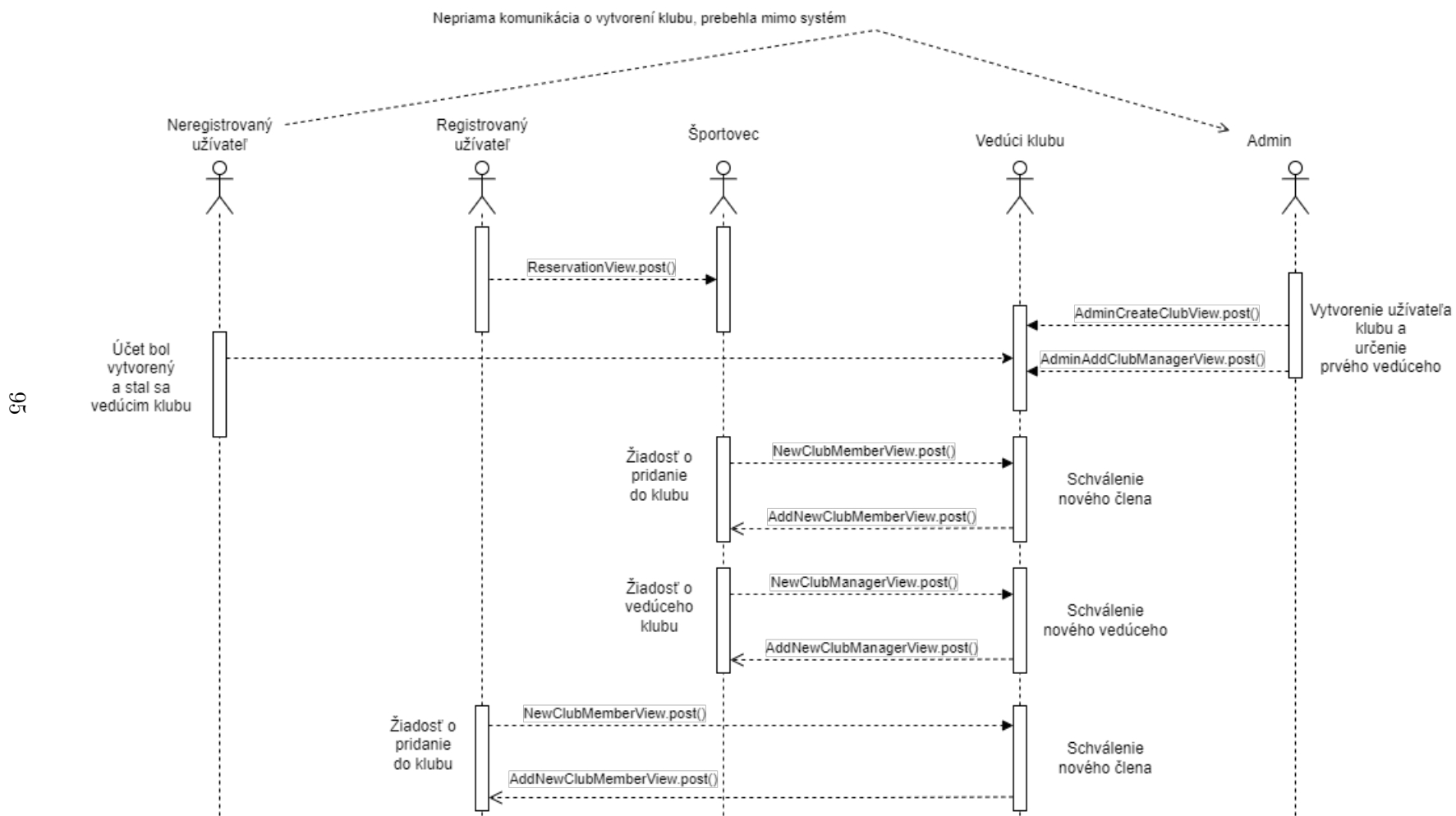
Obr. C.3: Životný cyklus vytvorenia klubu registrovaným užívateľom

Sekvenčný diagram vytvorenia klubu registrovaným užívateľom



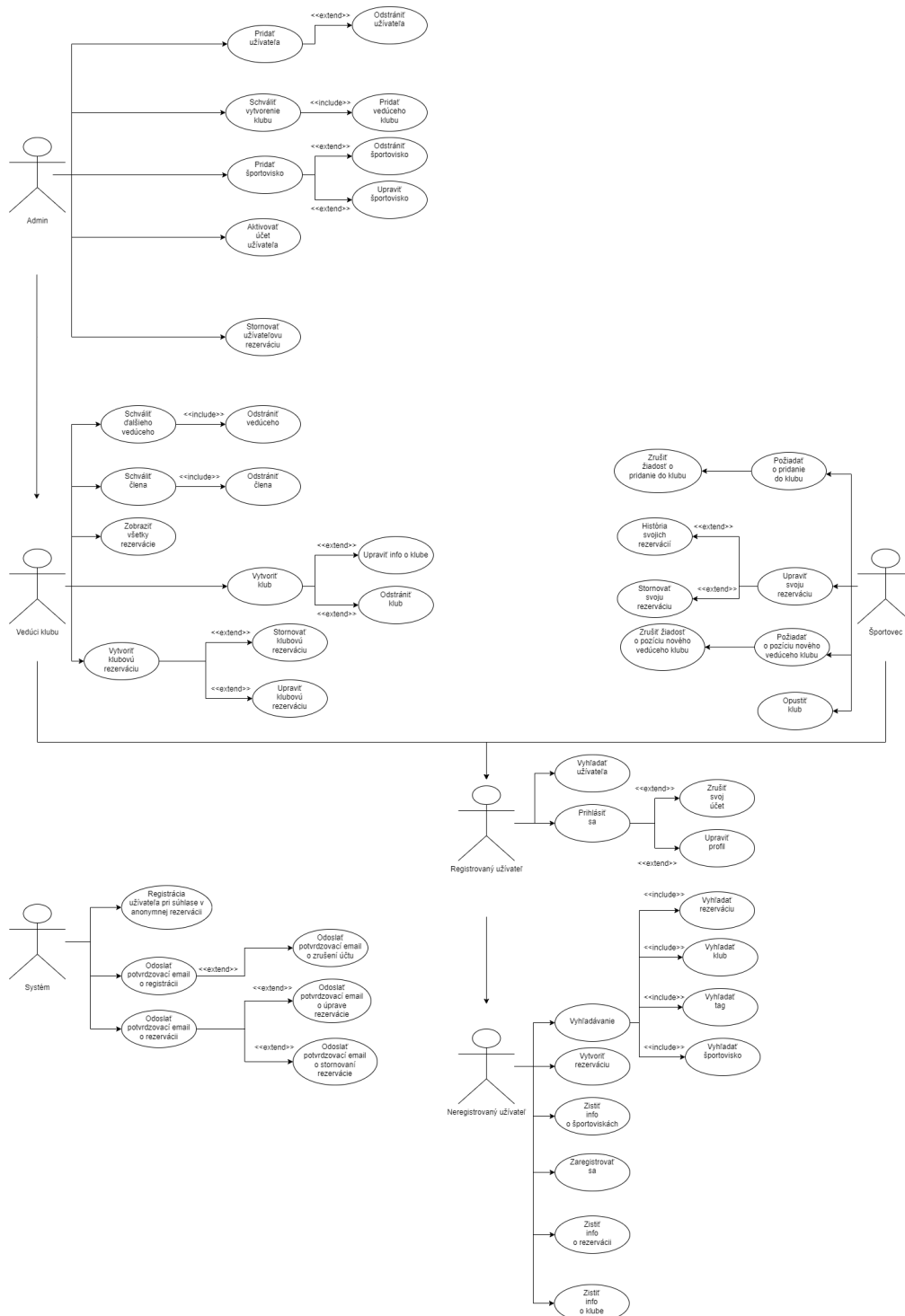
Obr. C.4: Sekvenčný diagram vytvorenia klubu registrovaným užívateľom

Sekvenčný diagram vytvorenia klubu neregistrovaným užívateľom



Obr. C.5: Sekvenčný diagram vytvorenia klubu neregistrovaným užívateľom

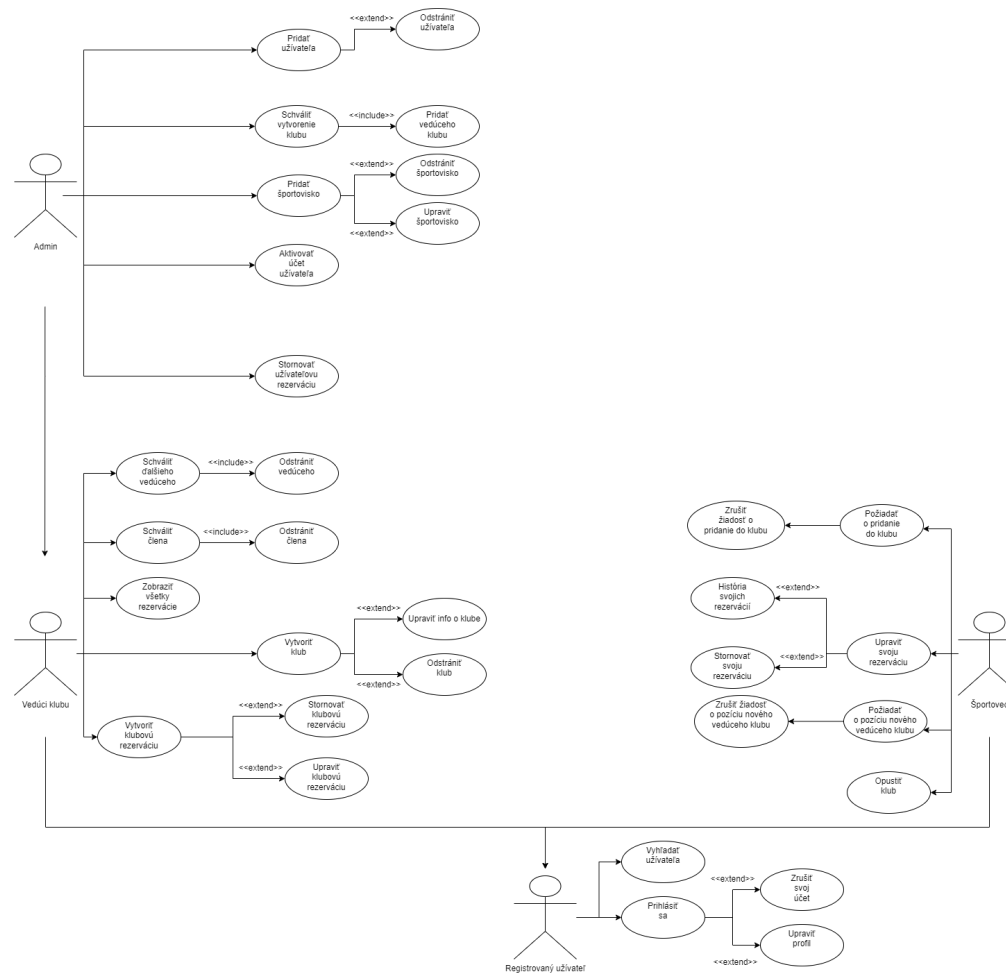
Kompletný Diagram prípadov použitia popisujúci chovanie celého systému



Obr. C.6: Diagram prípadov použitia celého systému

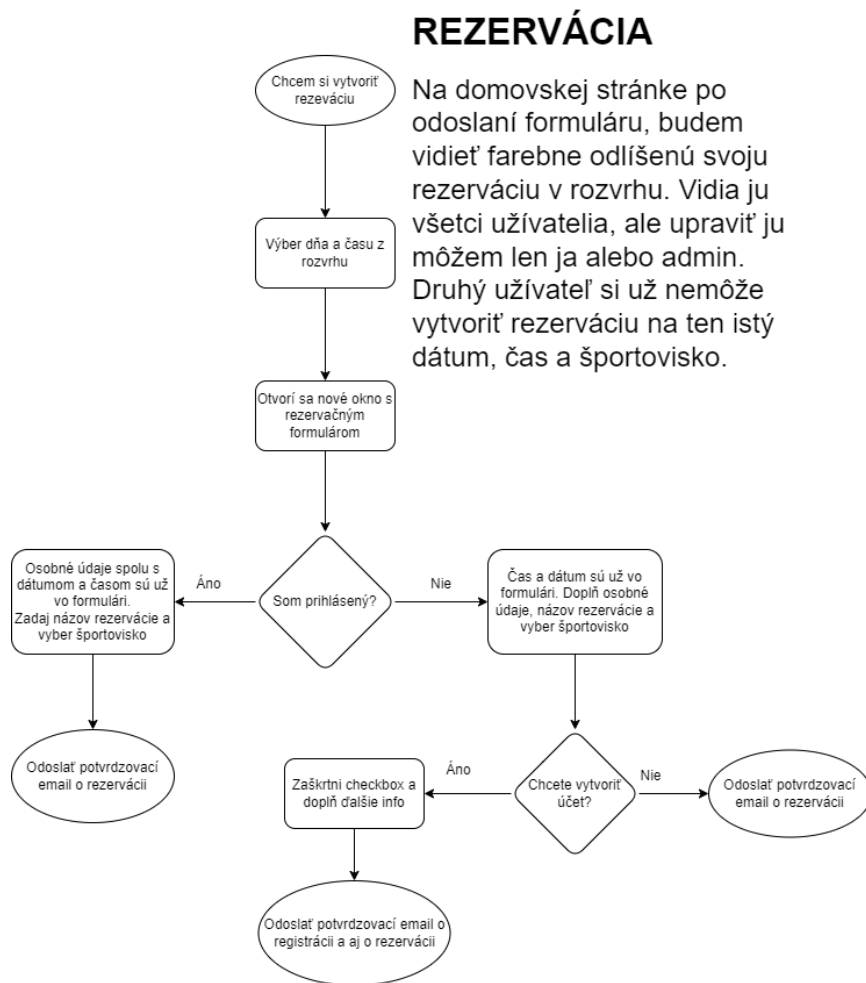
Diagram prípadov použitia popisujúci registrovaných užívateľov ako entitu

97



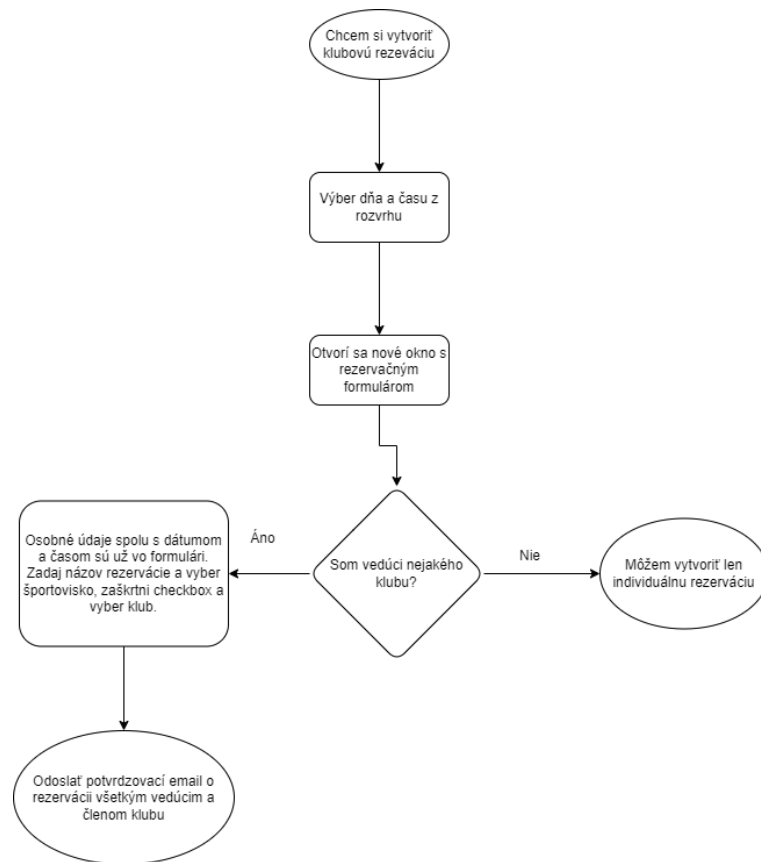
Obr. C.7: Diagram prípadov použitia registrovaných užívateľov

Vývojový diagram vytvorenia rezervácie



Obr. C.8: Vývojový diagram vytvorenia rezervácie

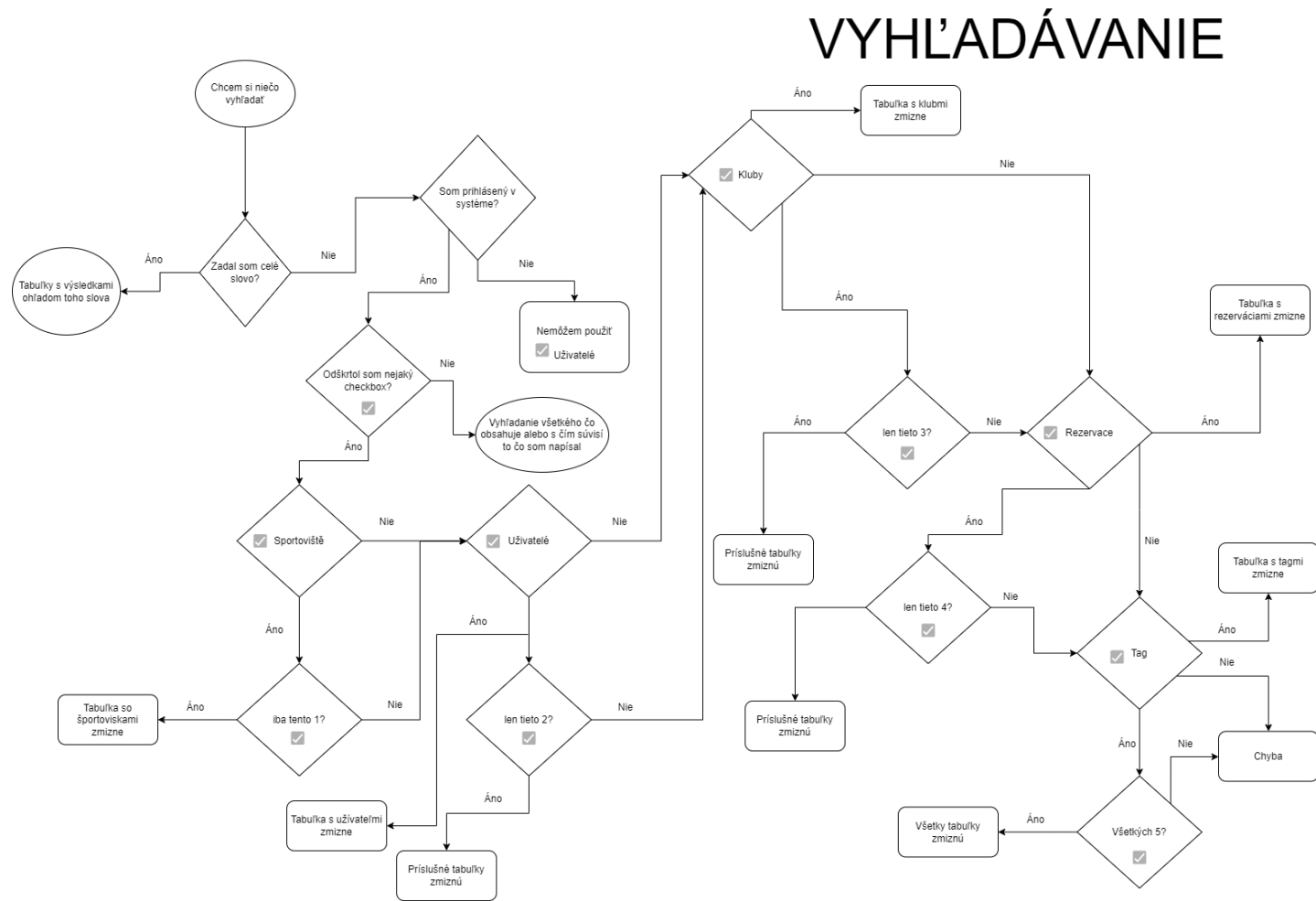
Vývojový diagram vytvorenia klubovej rezervácie



Obr. C.9: Vývojový diagram vytvorenia klubovej rezervácie

Vývojový diagram vyhľadávania

100



Obr. C.10: Vývojový diagram vyhľadávania

Príloha D

Stránky

Domovská stránka vedúceho klubu



[Kluby](#) [Rezervace](#) [Manuál](#) [Upravit profil](#) [Odhlásit se](#)

Email: belticek@gmail.com

Celé jméno: Juraj Remeň

Tag: #futbal

Vedoucí klubu: FC Medlánky, FK Iskra Hnúšťa

Člen klubu:

- Umelka
- Tenisový kurt
- Posilka
- Športová hala
- Futbalové ihrisko
- Basketbalové koše
- Piesočné ihrisko

	Po 06.05.2024	Út 07.05.2024	St 08.05.2024	Čt 09.05.2024	Pá 10.05.2024	So 11.05.2024	Ne 12.05.2024
8:00							
8:30	email .env	docker-compose	ssl	ssl club	Manual		
9:00	email .env	docker-compose	ssl	ssl club	Manual		

Obr. D.1: Domovská stránka vedúceho klubu

Domovská stránka športovca



[Kluby](#) [Moje rezervace](#) [Manuál](#) [Upravit profil](#) [Odhlásit se](#)

Email: xremen02@stud.fit.vutbr.cz

Celé jméno: Anton Berrolák

Tag: #volejbal

Vedoucí klubu:

Člen klubu:

Umelka

Tenisový kurt

Posilka

Športová hala

Futbalové ihrisko

Basketbalové koše

Piesočné ihrisko

103

	Po 06.05.2024	Út 07.05.2024	St 08.05.2024	Čt 09.05.2024	Pá 10.05.2024	So 11.05.2024	Ne 12.05.2024
8:00							
8:30	email .env	docker-compose	ssl	ssl club	Manual		
9:00	email .env	docker-compose	ssl	ssl club	Manual		

Obr. D.2: Domovská stránka športovca

Domovská stránka neregistrovaného uživatele



[Manuál](#) [Registrovat se](#) [Přihlásit se](#)

- Umelka
- Tenisový kurt
- Posilka
- Športová hala
- Futbalové ihrisko
- Basketbalové koše
- Piesočné ihrisko

	Po 06.05.2024	Út 07.05.2024	St 08.05.2024	Čt 09.05.2024	Pá 10.05.2024	So 11.05.2024	Ne 12.05.2024
8:00							
8:30	email .env	docker-compose	ssl	ssl club	Manual		
9:00	email .env	docker-compose	ssl	ssl club	Manual		
9:30	email .env	docker-compose	ssl	ssl club	Manual		
10:00							
10:30							
11:00							
11:30							
12:00							

104

Obr. D.3: Domovská stránka neregistrovaného uživatele

Príloha E

Abstrakt adresárovej štruktúry

```
reservation_system/  
├── app_name/  
│   ├── __init__.py  
│   ├── *.py  
│   └── migrations/  
│       ├── .3 templatetags/  
│       └── *.py  
├── reservation_system/  
│   ├── __init__.py  
│   └── *.py  
├── static/  
│   ├── style.css  
│   ├── js/  
│   │   └── *.js  
│   └── logos/  
│       └── *.png  
├── templates/  
│   └── app_name/  
│       └── *.html  
└── manage.py  
Pipfile  
Dockerfile  
docker-compose.yml  
db.sqlite3  
.env  
run_tests.sh
```