

Aplikace pro zákaznickou podporu

Radim Žaluda

Bakalářská práce
2023



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav informatiky a umělé inteligence

Akademický rok: 2022/2023

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Radim Žaluda**
Osobní číslo: **A20473**
Studijní program: **B0613A140020 Softwarové inženýrství**
Forma studia: **Kombinovaná**
Téma práce: **Aplikace pro zákaznickou podporu**
Téma práce anglicky: **Application for Customer Support**

Zásady pro vypracování

1. Proveďte rešerši existujících řešení.
2. Vypracujte stručný rozbor technologií, které budou použity k návrhu.
3. Proveďte rozbor a analýzu požadavků na zvolené řešení.
4. Zpracujte aplikaci na základě výsledků analýzy.
5. Věnujte pozornost zabezpečení aplikace.

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

Seznam doporučené literatury:

1. GORETTI, Anthony. *Introducing ASP.NET Core 6*. In: *Beginning gRPC with ASP.NET Core 6*. Apress, Berkeley, CA, 2022. p. 33-81.
2. JOHNSON, Glenn. *Programming in HTML5 with JavaScript and CSS3: training guide*. Redmond, Wash.: Microsoft, 2013. ISBN 978-0735674387.
3. UNHELKAR, Bhuvan. *Software engineering with uml*. Auerbach Publications, 2017.
4. LETT, Jacob. *Bootstrap 4 Quick Start: A Beginners Guide to Building Responsive Layouts with Bootstrap 4*. Bootstrap Creative, 2018.
5. JAKOBUS, Benjamin. *Mastering Bootstrap 4: Master the latest version of Bootstrap 4 to build highly customized responsive web apps*. Packt Publishing Ltd, 2018.
6. BEN-GAN, Itzik; DAVIDSON, Louis; VARGA, Stacia. *MCSA SQL Server 2016 Database Development Exam Ref 2-pack: Exam Refs 70-761 and 70-762*. Microsoft Press, 2017.

Vedoucí bakalářské práce:

doc. Ing. Petr Šilhavý, Ph.D.

Ústav počítačových a komunikačních systémů

Datum zadání bakalářské práce:

2. prosince 2022

Termín odevzdání bakalářské práce:

26. května 2023

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan



prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 7. prosince 2022

Prohlašuji, že

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

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 23.05.2023

Radim Žaluda, v. r.
podpis studenta

ABSTRAKT

Tato bakalářská práce vznikla za účelem realizace aplikace pro zákaznickou podporu. Důraz byl kladen zejména na uživatelskou přívětivost, jednoduchost a přehlednost, která za použití zdroje informací, helpdesk portálu, a real-time komunitního chatu zajistí rychlé a efektivní řešení uživatelských problémů či dotazů.

V teoretické části se tato práce zabývá rešerší již existujících řešení a dále rozbořem použitých technologií.

V praktické části lze poté nalézt rozbor a analýzu požadavků, na jejichž základě je následně vytvořena výsledná webová aplikace. Ta je potom podrobněji popsána formou uživatelské příručky. Praktická část této práce je zakončena přehledem možných bezpečnostních zranitelností aplikace, včetně toho, jak je proti nim aplikace zabezpečena.

Klíčová slova: Helpdesk, ASP.NET Core 6, MVC, WebAPI, C#, Angular, real-time chat

ABSTRACT

This bachelor thesis was created to implement a customer support application. The main focus was on user-friendliness, simplicity and clarity, using the information source, helpdesk portal, and real-time community chat to ensure quick and efficient resolution of user problems or queries.

The theoretical part of this thesis deals with the research of already existing solutions and the analysis of already existing technologies.

Within the practical part of this thesis you will see the analysis and requirement analysis, to which the resulting web application has been created. It is described in more detail in the form of a user manual. The practical part of this work concludes with an overview of possible security vulnerabilities of the application and then shows how the application is secured against them.

Keywords: Helpdesk, ASP.NET Core 6, MVC, WebAPI, C#, Angular, real-time chat

Zde bych rád poděkoval zejména vedoucímu této práce doc. Ing. Petrovi Šilhavému, Ph.D., za odborné konzultace, cenné rady a připomínky. Dále patří mé poděkování rodině a přítelkyni za podporu a motivaci, po celou dobu studia a vypracování této práce.

Prohlašuji, že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

„Život je čekání na to, kdy se naše sny změni ve skutečnost.“ – Victor Hugo

OBSAH

ÚVOD	9
I TEORETICKÁ ČÁST	11
1 REŠERŠE EXISTUJÍCÍCH ŘEŠENÍ	12
1.1 SPICEWORKS CLOUD HELPDESK.....	12
1.1.1 Popis aplikace.....	13
1.1.2 Analýza uživatelského rozhraní aplikace.....	13
1.1.3 Shrnutí.....	14
1.2 FRESHDESK.....	14
1.2.1 Popis aplikace.....	15
1.2.2 Analýza uživatelského rozhraní aplikace.....	15
1.2.3 Shrnutí.....	16
1.3 JITBIT.....	17
1.3.1 Popis aplikace.....	17
1.3.2 Analýza uživatelského rozhraní aplikace.....	18
1.3.3 Shrnutí.....	19
1.4 OSTICKET.....	19
1.4.1 Popis aplikace.....	19
1.4.2 Analýza uživatelského rozhraní aplikace.....	20
1.4.3 Shrnutí.....	20
1.5 SHRNUÍ A VÝSLEDEK REŠERŠE.....	21
1.6 VLASTNÍ PŘÍNOS.....	22
2 ROZBOR POUŽITÝCH TECHNOLOGIÍ	23
2.1 ASP.NET CORE MVC.....	23
2.1.1 Účel použité technologie v aplikaci.....	23
2.1.2 Popis technologie.....	23
2.2 C#.....	24
2.2.1 Účel použité technologie v aplikaci.....	24
2.2.2 Popis technologie.....	24
2.3 VISUAL STUDIO 2022.....	25
2.3.1 Účel použité technologie v aplikaci.....	25
2.3.2 Popis technologie.....	25
2.4 ENTITY FRAMEWORK CORE.....	25
2.4.1 Účel technologie v aplikaci.....	25
2.4.2 Popis technologie.....	25
2.5 SIGNALR.....	26
2.5.1 Účel technologie v aplikaci.....	26
2.5.2 Popis technologie.....	26
2.6 ASP.NET WEB API.....	27
2.6.1 Účel použité technologie v aplikaci.....	27
2.6.2 Popis technologie.....	27
2.7 ANGULAR.....	28
2.7.1 Účel technologie v aplikaci.....	28

2.7.2	Popis technologie	28
2.8	BOOTSTRAP	29
2.8.1	Účel technologie v aplikaci	29
2.8.2	Popis technologie	29
2.8.3	HTML	29
2.8.4	CSS.....	30
2.8.5	JavaScript	30
2.9	MICROSOFT SQL SERVER	31
2.9.1	Účel technologie v aplikaci	31
2.9.2	Popis technologie	31
2.10	UML	31
II PRAKTICKÁ ČÁST		33
3	ANALÝZA SYSTÉMU	34
3.1	POPIS.....	34
3.1.1	Uživatelé systému	34
3.1.2	Funkční požadavky	34
3.1.2.1	Správa systému	35
3.1.2.2	Správa tiketů	36
3.1.2.3	Správa úvodní stránky	37
3.1.2.4	Správa uživatelů.....	38
3.1.3	Nefunkční požadavky.....	39
3.1.4	Aktéři.....	40
3.1.5	Use Case model.....	42
3.1.6	Specifikace případů užití.....	42
3.2	DATABÁZE HELPDESK PORTÁLU	48
3.2.1	Users.....	49
3.2.2	Admin.....	50
3.2.3	Tickets	51
3.2.4	TicketsReply	51
3.2.5	Categories.....	51
3.2.6	ResetPassword.....	52
3.2.7	SMTPSettings	52
3.2.8	FileUploadLogs.....	53
4	APLIKACE.....	54
4.1	ÚVODNÍ STRÁNKA	54
4.2	REAL-TIME CHAT.....	54
4.3	HELPDESK PORTÁL	57
4.3.1	Uživatelská sekce	58
4.3.2	Administrátorská sekce	62
5	ZABEZPEČENÍ APLIKACE	66

5.1	UKLÁDÁNÍ HESEL DO DATABÁZE.....	66
5.2	OCHRANA PROTI SQL INJECTION	67
5.3	OCHRANA PROTI CROSS-SITE SCRIPTING (XSS)	67
5.4	OCHRANA PROTI CROSS-SITE REQUEST FORGERY (CSRF).....	68
5.5	AUTENTIZACE	69
5.6	AUTORIZACE	69
5.7	SSL	70
ZÁVĚR		71
SEZNAM POUŽITÉ LITERATURY.....		72
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....		74
SEZNAM OBRÁZKŮ		76
SEZNAM TABULEK.....		78
SEZNAM PŘÍLOH.....		79

ÚVOD

V dnešní digitální době, které jsme všichni chtěli nechtěli součástí, se většina lidí neobejde bez informačního či komunikačního prostředku, jak v běžném, tak pracovním životě. Uživatelé těchto prostředků se díky tomu mohou dennodenně potýkat s nejrůznějšími problémy, které potřebují řešit. Tyto problémy lze efektivně řešit prostřednictvím aplikací k tomu určených – aplikací pro zákaznickou podporu. Aplikací tohoto typu existuje značná řada, jsou ale většinou robustní a určeny spíše pro komplexní správu požadavků většího týmu řešitelů.

Tato práce (potažmo výsledná aplikace) si tedy klade za cíl, do maximální možné míry zjednodušit a uživatelsky zpříjemnit proces, který vede k úspěšnému vyřešení problému či dotazu. K tomu ve výsledné aplikaci slouží helpdesk portál, kde je kladen důraz na jednoduchost a uživatelskou přívětivost – nezatěžovat uživatele složitým zakládáním požadavků a přehrše formulářových polí. Dále real-time komunitní chatovací aplikace (dále jen real-time chat), která umožní v reálném čase vznést dotaz či problém k ostatním uživatelům chatu, a jako podpůrný nástroj je součástí výsledné aplikace znalostní báze a FAQ (dále jen zdroj informací), kde lze nalézt informace o již dříve řešených problémech.

Teoretická část je rozdělena na dvě kapitoly – rešerši dostupných řešení a stručný popis použitých technologií. Pro rešerši byla zvolena čtveřice již existujících řešení, na základě hodnotících kritérií, která budou konkrétně pospány dále. V souvislosti s použitými technologiemi jsou primárně detailněji rozepsány C#, ASP.NET Core 6, MVC, Web API, Angular, Bootstrap, MSSQL, Entity Framework, SignalR.

V praktické části je popsán celý proces návrhu, přes tvorbu a popis výsledné aplikace formou uživatelské příručky. Celý popisovaný proces začíná zpracováním analýzy, která zahrnuje funkční a nefunkční požadavky na aplikaci + popis aktérů. Dále pak Use Case model, včetně podrobného popisu jednotlivých tříd a jejich atributů, následuje specifikace případů užití a diagram tříd. Poslední část analýzy požadavků tvoří E-R diagram s popisem jednotlivých atributů.

V další části je pak názorně zobrazena a popsána výsledná aplikace. Nejprve krátce úvodní stránka, která obsahuje sekce novinky, znalostní bázi, FAQ, vstup do real-time chatu a helpdesk portálu. Následuje podrobnější uživatelská příručka pro real-time chat a helpdesk portál.

Závěr celé praktické části je věnován zabezpečení aplikace – tedy přehledem možných zranitelností, včetně reálné ukázky a popisem toho, jak je proti nim výsledná aplikace zabezpečena.

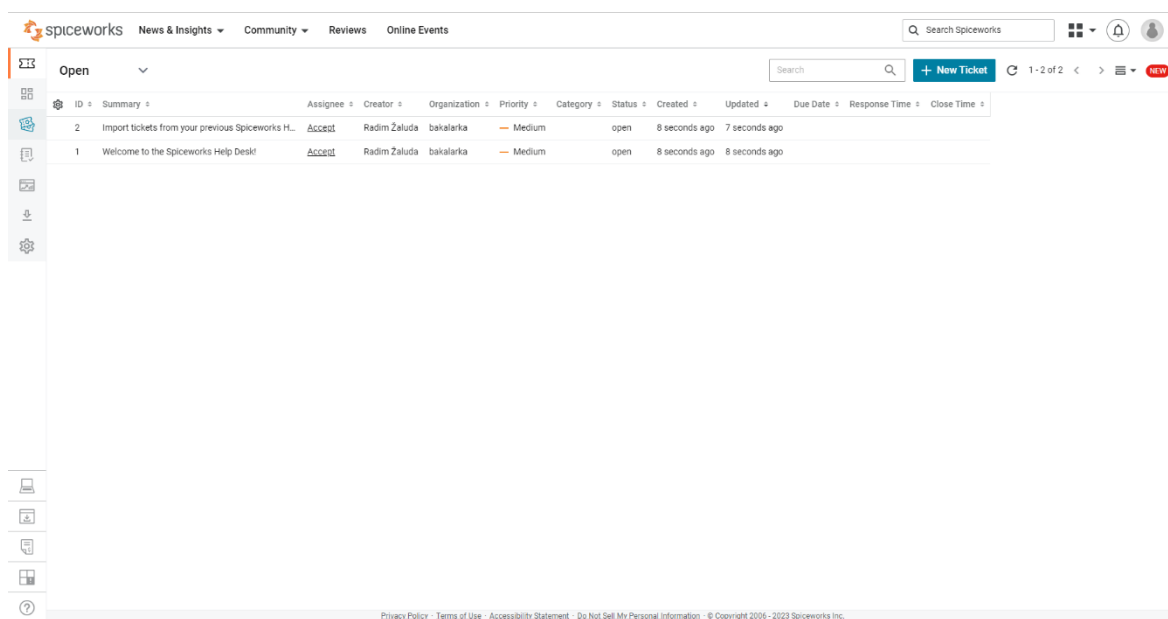
I. TEORETICKÁ ČÁST

1 REŠERŠE EXISTUJÍCÍCH ŘEŠENÍ

Internet nabízí poměrně velké množství komplexních řešení webových aplikací pro zákaznickou podporu. Aby bylo srovnání relevantní a účelné, byl v závěrečném vyhodnocení kladen důraz na přítomnost následujících aspektů:

- **Webová aplikace**
- **Moderní a přehledné uživatelské rozhraní**
- **Jednoduché zadávání a správa tiketů**
- **Znalostní báze**
- **FAQ**
- **Možnost provozu v cloudu nebo on-premise**
- **Zdarma**
- **Česká jazyková mutace**
- **Real-time chat**

1.1 SpiceWorks Cloud Helpdesk



Obrázek 1 Přehled sekce „Tickets“ v aplikaci SpiceWorks (zdroj spiceworks.com)

1.1.1 Popis aplikace

SpiceWorks Helpdesk nabízí centralizované sledování a správu tiketů. Uživatelé mohou vytvářet tikety prostřednictvím webového rozhraní a sledovat stav svých požadavků v reálném čase. Helpdesk také poskytuje nástroje pro kategorizaci, prioritizaci a eskalaci tiketů, což pomáhá zlepšit řízení ICT požadavků. [1]

Další výhodou Spiceworks Helpdesk je jeho aktivní komunita uživatelů, díky které lze sdílet informace, zkušenosti a řešení mezi uživateli. Uživatelé mohou komunikovat, diskutovat a spolupracovat na řešení ICT problémů a získávat cennou zpětnou vazbu od ostatních členů komunity.[1]

1.1.2 Analýza uživatelského rozhraní aplikace

- **Dashboard** – přehledová obecná sekce, kde lze nalézt aktuální počty a stavy tiketů, dále pak grafické znázornění evidovaných tiketů
- **Tickets** – sekce, ve které lze vytvořit nový tiket, dále zobrazit podrobnosti o konkrétním tiketu a v případě potřeby měnit jeho vlastnosti (priorita, kategorie, přiřazení, status)
- **Knowledge Base** – zde je možné procházet znalostní bázi. Informace lze vyhledávat v rámci všech uživatelů, pouze svého týmu nebo jen těch vlastních.
- **Reports** – tato sekce nabízí možnost generování reportů, na základě filtrů (mj. datum vytvoření, status, priorita, kategorie). Výstup je pak možný do dvou formátů .csv nebo .json
- **Settings** – nabízí několik sekcí, kde lze definovat např. pravidla přiřazování akcí tiketům či kategoriím, dále pak vlastní pohledy tiketů, administraci uživatelů a obecná nastavení jako formát času nebo automatická aktualizace tiketů

Plusy

- + Kompletně zdarma
- + Přehledná administrace
- + UI/UX
- + Jednoduché a uživatelsky přívětivé zakládání a editace tiketů
- + Možnost generování reportů
- + Sdílení informací s ostatními uživateli

Mínusy

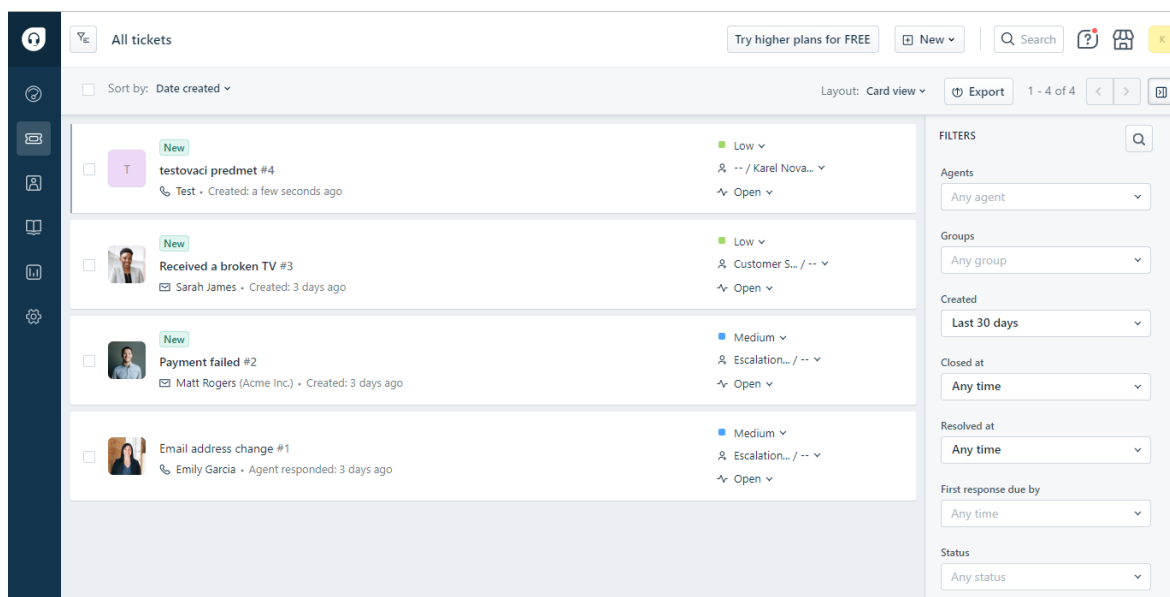
- Pouze v angličtině
- Chybí real-time chat
- Kód není open-source
- pouze cloudová verze

1.1.3 Shrnutí

Aplikace SpiceWorks Cloud Helpdesk působí moderně, má přehledné uživatelské rozhraní, snadné zakládání a správu tiketů. Nechybí znalostní báze a možnost sdílení problémů, a jejich řešení mezi ostatními uživateli.

Bohužel aplikaci jde provozovat pouze v cloudu, kód není open-source. Provozovatel aplikace je tedy odkázán pouze na funkcionalitu, kterou nabízí výrobce softwaru, tedy SpiceWorks. Vytknout je záhodno absenci real-time chatu.

1.2 Freshdesk



Obrázek 2 Přehled sekce „All tickets“ v aplikaci Freshdesk (zdroj freshdesk.com)

1.2.1 Popis aplikace

Freshdesk Helpdesk je moderní cloudový helpdesk navržený pro efektivní správu zákaznické podpory a řešení ICT problémů. Tento systém nabízí pokročilé funkce a uživatelsky přívětivé rozhraní, které usnadňují sledování a správu požadavků zákazníků a zlepšují jejich celkový zážitek. [2]

Jednou z klíčových funkcí Freshdesk Helpdesk je centralizované sledování a správa zákaznických tiketů. Helpdesk také nabízí pokročilé nástroje pro kategorizaci, prioritizaci a eskalaci tiketů, což umožňuje efektivní řízení a rozdělení práce mezi členy týmu. [2]

Freshdesk Helpdesk poskytuje širokou škálu funkcí pro automatizaci a správu pracovních postupů. Uživatelé mohou vytvářet pravidla pro různé typy tiketů, což usnadňuje jejich správu a řešení. Systém také nabízí možnosti sledování časů a činností, což umožňuje monitorování výkonnosti a zlepšení produktivity týmu. [2]

1.2.2 Analýza uživatelského rozhraní aplikace

- **Dashboard** – přehledová sekce o stavu tiketů (otevřených, uzavřených, čekajících, nepřirazených), graf vykreslující počty tiketů v čase a tabulka spokojenosti zákazníků
- **Tickets** – zde lze založit nový tiket, zobrazit již založené tikety, editovat jejich vlastnosti (např. stav, priorita, typ, přiřazení konkrétnímu řešiteli, aj.)
- **Contacts** – tato sekce nabízí možnost vést jakousi evidenci kontaktů jednotlivců (jméno, příjmení, e-mail, telefon, aj.) a společností.
- **Solutions** – zde se nachází znalostní báze a FAQ, příspěvky a informace lze vyhledávat, a následně kategorizovat do složek
- **Analytics** – analytická sekce pro generování předdefinovaných reportů, které lze zaslat e-mailem či stáhnout ve formátu .pdf
- **Admin** – v administrátorské sekci je nepřehledné množství nastavení, funkcionalit a rozšíření. V rámci vytyčeného směru popisu funkcionalit se jimi ale nebylo zabýváno.

Plusy

- + UI/UX
- + Přehledná administrace
- + Součást platformy obsahující nejrůznější moduly
- + Znalostní báze

+ Mobilní aplikace

Mínusy

- Česká lokalizace pouze v omezené míře
- Integrace real-time chatu není přímo do helpdeskové aplikace
- Omezený počet agentů (verze zdarma)
- Pouze v cloudu
- Kód není open-source

1.2.3 Shrnutí

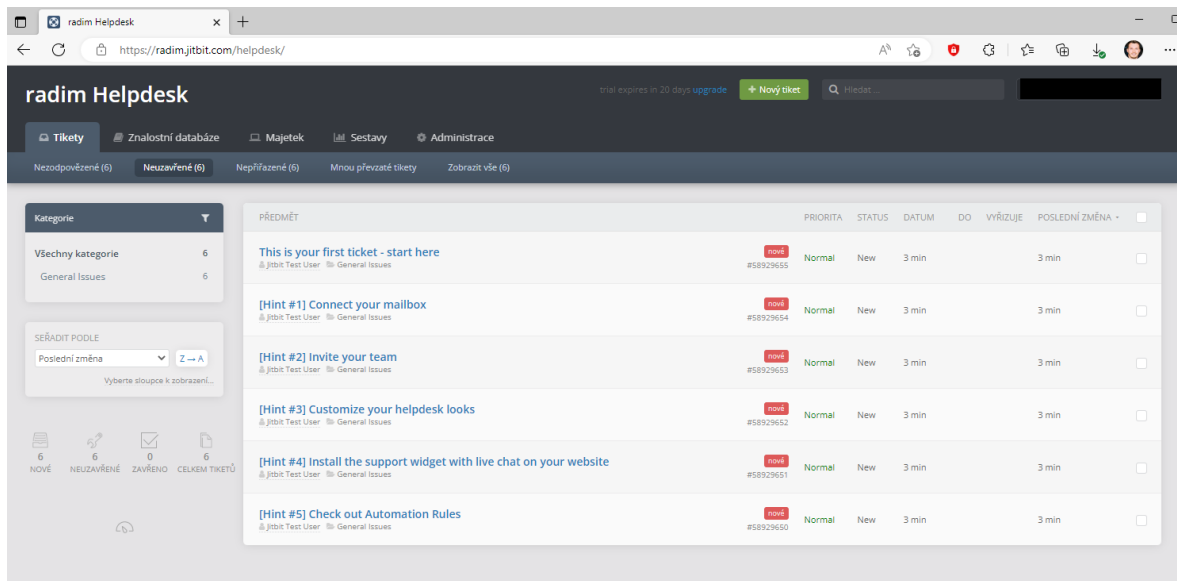
Freshdesk je součástí platformy Freshworks, který nabízí komplexní portfolio produktů pro správu zákaznické podpory. Bohužel ne všechny produkty jsou zdarma. Oproti ostatním zkoumaným řešením má určitě nejpříjemnější vzhled a uživatelské rozhraní. Formulář pro zakládání tiketů je komplexnější, než je nezbytně nutné. Editace a správa tiketů

Freshdesk nabízí i českou lokalizaci, kde je překlad hodně strojový, kostrbatý a nepřesný. Jelikož se jedná o komerční produkt, není kód vyvíjen open-source. Aplikaci lze provozovat pouze v cloudu.

Real-time chat je sice součástí Freshworks platformy, lze jej provozovat zdarma, bohužel ale není plně integrován do produktu Freshdesk.

Toto řešení je použitelné i pro jednotlivce, nicméně díky robustnosti je určeno spíše podnikům s více pobočkami a vlastním ICT oddělením, kde se o řešení problémů uživatelů a zákazníků stará více lidí.

1.3 JitBit



Obrázek 3 Přehled sekce „Tikety“ v aplikaci JitBit (zdroj jitbit.com)

1.3.1 Popis aplikace

Jitbit Helpdesk je postaven na Microsoft ASP.NET a MSSQL databázovém systému. Poskytuje tiketovací systém, který umožňuje zákazníkům zadávat dotazy nebo problémy. Zákaznická podpora může tikety spravovat, přiřazovat je konkrétním členům týmu, sledovat jejich stav a poskytovat odpovědi či řešení. [3]

Jitbit Helpdesk umožňuje vytvářet a spravovat znalostní bázi, často kladené dotazy (FAQ), návody, dokumentace a další informace pro zákazníky. To umožňuje zákazníkům vyhledávání odpovědí na své dotazy a týmu zákaznické podpory ulehčuje řešení opakujících se dotazů. [3]

Lze jej integrovat s dalšími nástroji a systémy, jako je e-mailový klient, CRM systémy, sociální média atd. Navíc poskytuje rozsáhlé funkce reportingu, které umožňují sledovat výkonnost zákaznické podpory, analyzovat data o tiketech, doby řešení, spokojenosti zákazníků a dalších metrik. [3]

Tento helpdesk je navržen s důrazem na uživatelskou přívětivost a jednoduchost použití. Poskytuje intuitivní a přehledné rozhraní, které usnadňuje týmu zákaznické podpory práci s tikety, sledování stavu a komunikaci se zákazníky. [3]

1.3.2 Analýza uživatelského rozhraní aplikace

- **Tikety** – tato sekce nabízí přehled vytvořených tiketů, jejich stav a možnost editace vlastností (změna kategorie, stavu, označení tiketu jako SPAM, smazat atd.)
- **Znalostní Báze** – sekce kde lze hledat, vytvářet a editovat znalostní bázi. Je zde i tzn. fórum nápadů, které umožňuje navrhnout různá vylepšení
- **Majetek** – evidence majetku, kterou lze přizpůsobit dle svých potřeb. Lze jej třídit do kategorií. Existující majetek lze i importovat pomocí .csv souboru
- **Sestavy** – tuto sekci lze použít pro potřeby souhrnných výstupů. Výstup lze filtrovat dle různých vlastností tiketu (např. status, priorita, společnost, oddělení, uzavřeno od-do atd.)
- **Administrace** – v administrační sekci lze upravit vzhled aplikace dle svých preferencí. Dále nastavení velké škály různých jazykových mutací celé aplikace. Možnost zapnutí a vypnutí různých nastavení (např. viditelnost tiketů ostatním, automatické uzavírání tiketů po definovaném čase, možnost otevřít uzavřený tiket atd.). Rozsáhle lze spravovat i nastavení ověření (např. dvou faktorové ověrování 2FA)

Plusy

- + UI/UX
- + Jednoduché a přehledné zakládání tiketů
- + Lze provozovat v cloudu (zpoplatněno) i on-premise
- + Mobilní aplikace
- + Velká míra přizpůsobení vzhledu a funkcí
- + Čeština (některých případech kostrbatá)

Mínusy

- Dashboard tiketů
- Chybí real-time chat
- Zdarma pouze v případě provozu on-premise
- Kód není open-source (lze jej zakoupit, nicméně za neveřejnou cenu)
- Cloudová verze je zpoplatněna (od 29\$/měsíc)

1.3.3 Shrnutí

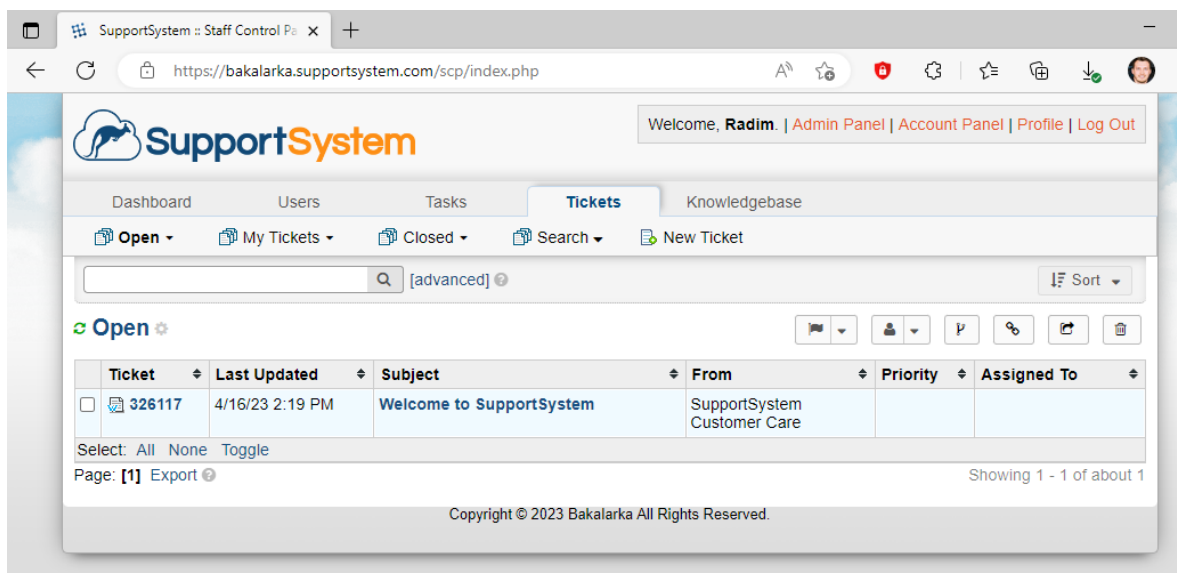
JitBit helpdesk upoutá svou jednoduchou a přehlednou webovou aplikací.

Vytváření tiketů je jednoduché a uživatelsky přívětivé, bez zbytečných formulářových polí.

Pokud je provozována na vlastním prostředí, je k dispozici časově neomezená trial verze zdarma, nicméně v případě potřeby lze rozšířit funkcionalitu prostřednictvím modifikace kódu, který je ale nutné zakoupit (cena je na stránkách neveřejná).

Vytknout lze místy nepřeložené názvy do češtiny a chybějící real-time chat.

1.4 osTicket



Obrázek 4 Přehled sekce „Tickets“ v aplikaci osTicket (zdroj osticket.com)

1.4.1 Popis aplikace

osTicket je open-source helpdesk aplikace, kterou lze využít pro správu tiketů a zákaznickou podporu. Je vyvíjena v jazyce PHP a MySQL databázovém systému. [4]

Je možno jej provozovat dvěma způsoby:

1. Vlastním hostingem (na IIS nebo Apache), kdy jsou v rámci krátkého průvodce navo- leny pluginy a jazykové balíčky. Následně lze stáhnout zdrojové kódy. [4]
2. V cloudu, kdy je ale provoz aplikace zpoplatněn řádově v jednotkách dolarů/měsíc. V rámci této verze je k dispozici podpora, zálohování a vysoká dostupnost. [4]

1.4.2 Analýza uživatelského rozhraní aplikace

Po přihlášení, je úvodní stránka členěna na pětici záložek – Dashboard, Users, Tasks, Ticket a Knowledge Base.

- **Dashboard** – přehledový graf a statisticky tiketů (např. počty otevřených, uzavřených) včetně doby zpracování
- **Users** – správa uživatelů. Lze zde přidávat, odebírat
- **Tasks** – možnost založit novou úlohu zvolenému řešiteli či oddělení
- **Tickets** – sekce, kde lze nalézt založené tikety, přidělené tikety, uzavřené tikety. Lze zde tikety spravovat, řadit či vyhledávat
- **Knowledgebase** – sekce databáze znalostí, je zde možno vytvářet často kladené otázky a řadit je do kategorií

Plusy

- + Kompletně zdarma včetně open-source kódu
- + Jednoduché zakládání tiketů

Mínusy

- Nevzhledný design aplikace
- Chybí real-time chat
- Chybí česká lokalizace

1.4.3 Shrnutí

osTicket nepůsobí po vizuální stránce moderně. Aplikace obsahuje znalostní bázi i FAQ. Co se týče správy tiketů, nabízí poměrně jednoduché zakládání a správu, včetně možnosti přiřazování tiketů jiným uživatelům. Jelikož je kód k dispozici open-source, lze si aplikaci přizpůsobit dle svých potřeb.

Aplikace bohužel není k dispozici v českém jazyce a chybí real-time chat.

1.5 Shrnutí a výsledek rešerše

Na základě hodnotících kritérií definovaných v úvodu této kapitoly byla vytvořena přehledová tabulka, která demonstruje, jak daná aplikace splňuje či nespĺňuje tato kritéria. Jak lze z tabulky (Tabulka 1) vyčíst, všechna analyzovaná existující řešení ve své podstatě splňují základní požadovanou a zamýšlenou funkcionalitu výsledné aplikace. Tedy aplikace, kde lze nalézt znalostní bázi, FAQ a helpdesk portál pro zadávání, správu a přehled problémů.

Tabulka 1 Přehledová tabulka výsledků aplikací dle hodnotících kritérií

Faktor	SpiceWorks	FreshDesk	osTicket	JitBit
Správa tiketů	ANO	ANO	ANO	ANO
Dashboard	ANO	ANO	ANO	ANO
Real-time chat	NE	NE	NE ¹	NE
Znalostní báze	ANO	ANO	ANO	ANO
FAQ	ANO	ANO	ANO	ANO
Open-source kód	NE	NE	ANO	NE
Webová aplikace	ANO	ANO	ANO	ANO
UI/UX ²	+	+	-	+
Cloud	ANO	ANO	ANO	ANO
On-premise	NE	NE	ANO	ANO
Česká lokalizace	NE	ANO ³	NE	ANO
Cena	Zdarma	Zdarma ⁴	Zdarma	Zdarma

¹ je k dispozici i real-time chat zdarma, bohužel ale není přímo integrován do aplikace

² myšleno moderní, příjemné, a hlavně jednoduché uživatelské rozhraní (+ splňuje, - nespĺňuje)

³ pouze některé sekce, často strojově a kostrbatě

⁴ aplikace nabízí i placené verze, které poskytují mnoho dalších funkcionalit. Na základě definovaných kritérií jsou pro toto srovnání nerelevantní.

1.6 Vlastní přínos

Z výše uvedeného vyplývá, že hlavním nedostatkem všech analyzovaných řešení je absence real-time chatu. Jako stěžejní přínos této aplikace bude právě přímá implementace real-time chatu do výsledné aplikace pro zákaznickou podporu.

Helpdesk portál vychází částečně z již existujícího řešení. Oproti původnímu řešení byl ale vylepšen, optimalizován a přizpůsoben požadavkům na základě analýzy systému, která bude popsána v praktické části. Konkrétní vylepšení oproti původnímu řešení budou popsána v praktické části této práce.

2 ROZBOR POUŽITÝCH TECHNOLOGIÍ

V této kapitole se nachází stručný rozbor použitých technologií. Vždy byl uveden účel, ke kterému daná technologie v aplikaci slouží a dále pak popis technologie jako takové.

2.1 ASP.NET CORE MVC

2.1.1 Účel použité technologie v aplikaci

Helpdesk portál vychází z řešení, které bylo vyvíjeno na technologii ASP.NET 5, nicméně pro účely této aplikace bylo předěláno do novějšího ASP.NET Core 6 za použití architektonického vzoru MVC, tedy Model-View-Controller.

2.1.2 Popis technologie

ASP.NET Core MVC umožňuje vytvářet webové aplikace. Poprvé se stalo součástí ekosystému .NET v roce 2008. [5]

Framework byl prvně založen na událostech (kliknutí, načtení stránky atd.), takže měl tendenci být složitý, těžký a neefektivní. Snaha Microsoftu vytvořit odlehčený a efektivnější framework ASP.NET vyústila v ASP.NET MVC, který se řídí vzorem Model-View-Controller (MVC) a zavádí mnoho výhod do ASP.NET (a ASP.NET Core), jako jsou následující: [5]

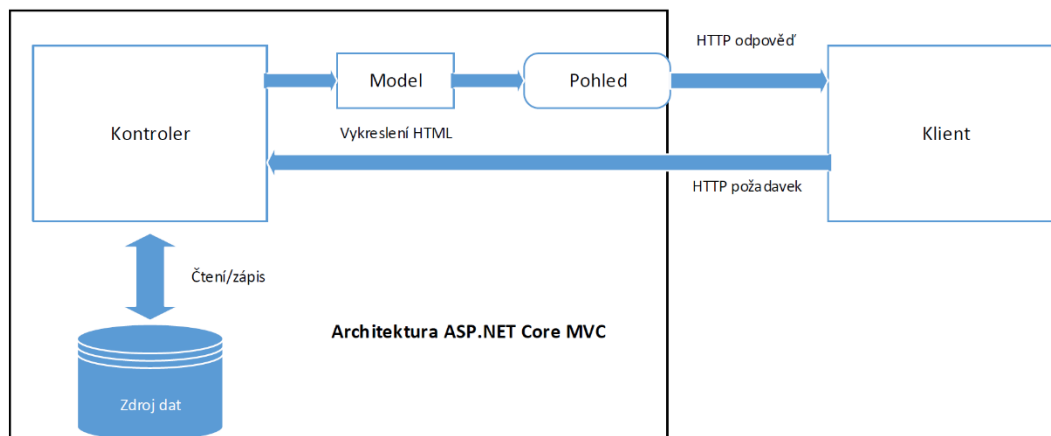
- Čistá architektura s principem Separation of Concerns (SoC).
- Efektivní a snadná kombinace s JavaScript knihovnami jako JQuery nebo Prototype.js
- Search Engine Optimization (SEO) Funkce ASP.NET Core Routing poskytuje více afinity k vyhledávačům
- Úplná kontrola vykreslování HTML, na rozdíl od ASP.NET WebForms
- Test-driven development (TDD)

V ASP.NET Core je architektura MVC následující: [5]

1. **Model** je komponenta v návrhovém vzoru MVC, spravující data, která mají být zobrazena v komponentě View
2. **View** je komponenta používaná k zobrazení dat modelu
3. **Controller** je komponenta v aplikaci ASP.NET MVC, která se používá ke zpracování příchozího požadavku HTTP generovaného akcí uživatele. Controller se zabývá

modelem a pohledem prostřednictvím Action (což je metoda controlleru) a poté odešle odpověď uživateli.

Pro lepší pochopení lze nahlédnout na obrázek níže (obrázek 5).



Obrázek 5 Architektura ASP.NET Core MVC. Zdroj [5]

2.2 C#

2.2.1 Účel použité technologie v aplikaci

V závislosti na volbě frameworku (ASP.NET Core) byl použit programovací jazyk C#.

2.2.2 Popis technologie

C# je objektově orientovaný, moderní a typově bezpečný programovací jazyk. Umožňuje vývojářům vytvářet nejrůznější typy bezpečných a robustních aplikací běžících na .NET. C# vychází z jazyka C. Funkce, které pomáhají vytvářet robustní a odolné aplikace jsou např: [6]

- Uvolňování paměti – automaticky uvolňuje paměť zabranou již nedostupnými či nepoužívanými prostředky
- Typy s hodnotou NULL – díky této funkci je možné provádět bezpečné operace s hodnotami, kterou mohou být NULL (speciální hodnota která neobsahuje žádnou hodnotu, je neznámá, chybějící nebo neinicializovaná)
- Zpracování výjimek – poskytuje strukturovaný a rozšiřitelný přístup k detekci chyb
- LINQ (Language Integrated Query) – technologie pro práci s daty a jejich zpracování, která umožňuje vytvářet dotazy nad datovými zdroji (objekty, databázové tabulky, aj.)

C# klade důraz na verzování z důvodu zajištění budoucího vývoje programů a knihoven kompatibilním způsobem. [6]

2.3 Visual Studio 2022

2.3.1 Účel použité technologie v aplikaci

Pro vývoj samotné aplikace bylo použito integrované vývojové prostředí (IDE) Visual Studio v aktuálně nejnovější verzi 2022.

2.3.2 Popis technologie

Visual Studio je integrované vývojové prostředí (IDE) vyvinuté společností Microsoft pro vývoj softwaru. Poskytuje širokou škálu nástrojů a funkcí pro vývoj a ladění aplikací ve více jazycích, včetně C#, C++, Visual Basic, F# a dalších. Visual Studio nabízí uživatelsky přívětivé rozhraní, které umožňuje vývojářům vytvářet, testovat, ladit a nasazovat aplikace pro různé platformy, včetně Windows, Linux, macOS a mobilních zařízení. [7]

Nová verze Visual Studia 2022 (oproti předchozím verzím 64bitová) přináší vylepšení výkonu, rychlejší hledání v souborech, novinkou je Hot Reload (umožňuje vývojářům upravovat zdrojový kód aplikace za běhu, tzn. bez nutnosti vždy znovu-spouštět aplikaci), vylepšena byla i funkce IntelliCode (inteligentní doplňování kódu) a mnoho dalšího. [8]

2.4 Entity Framework Core

2.4.1 Účel technologie v aplikaci

Tento framework ve výsledné aplikaci slouží jako nástroj pro práci s databází.

2.4.2 Popis technologie

Entity Framework Core je open-source framework pro vývoj databázových aplikací v jazyce C#. Framework umožňuje vytvářet aplikace pomocí objektově-relačního mapování (ORM), což znamená, že databázové tabulky jsou mapovány na třídy v C# kódu. Tento přístup umožňuje vývojářům pracovat s databázovými záznamy jako s objekty, což zjednodušuje manipulaci s daty. [9]

Entity Framework Core je multiplatformní a podporuje několik typů databázových systémů, včetně Microsoft SQL Server, MySQL, PostgreSQL a SQLite. Framework také umožňuje vytvářet databázové migrace, což zjednodušuje správu změn v databázi. [9]

2.5 SignalR

2.5.1 Účel technologie v aplikaci

Pro real-time chat byla použita open-source knihovna SignalR. Tato knihovna poskytuje real-time komunikaci mezi klientem a serverem pomocí WebSocketů, které jsou podporovány v mnoha prohlížečích.

2.5.2 Popis technologie

SignalR je open-source knihovna, která zjednodušuje přidávání webových funkcí v reálném čase do aplikací. Webové funkce, v reálném čase, umožňují kód na straně serveru okamžitě odesílat obsah klientům. [10]

SignalR poskytuje rozhraní API pro vytváření vzdálených volání procedur (RPC) mezi serverem a klientem. RPC vyvolávají funkce na klientech na základě kódu z .NET Core na straně serveru.[10]

Kdy je dobré použít knihovnu SignalR: [10]

- Pro aplikace, které vyžadují časté aktualizace ze serveru. Příkladem mohou být hry, sociální sítě, mapy a aplikace GPS.
- Dashboardy a monitorovací aplikace. Příkladem jsou firemní řídicí panely, okamžité aktualizace prodejů nebo upozornění na cesty.
- Aplikace, které vyžadují oznámení, např. sociální sítě, e-mail, chat, hry, upozornění na cesty, aj.

Funkce SignalR pro ASP.NET Core jsou například: [10]

- Automatické zpracování požadavků na připojení
- Odesílání zpráv všem připojeným klientům současně (například chatovací místnost)
- Odesílá zprávy konkrétním klientům
- Hub protokol

2.6 ASP.NET Web API

2.6.1 Účel použité technologie v aplikaci

Pro back-end část real-time chatu byl použit ASP.NET Web API. Prostřednictvím něj se získávají a předávají data pro front-end část.

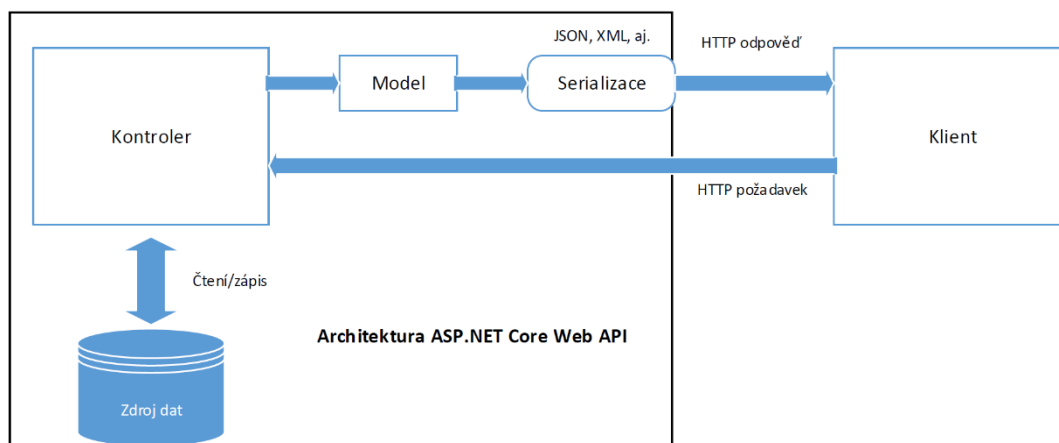
2.6.2 Popis technologie

ASP.NET Core Web API umožňuje vytvářet webové API. API neboli aplikační programový interface je používán ve spojitosti s HTTP. V současné době webová API používají protokol REST (Representational State Transfer), který je spojen s JSON (JavaScript Object Notation) a XML (Extensible Markup Language) formáty. [5]

Jelikož uživatelé internetu používají k přístupu na web nejrůznější terminály, a v této spojitosti chceme poskytovat data prohlížečům a aplikacím rychlým a bezpečným způsobem, potřebujeme, aby bylo Web API kompatibilní se všemi přístupovými body. ASP.NET Core Web API je navržen jako framework určený speciálně pro vytváření webových služeb pro použití velkém množství uživatelů. [5]

ASP.NET Core Web API se řídí MVC návrhovým vzorem. V tradičních webových aplikacích je View v MVC webová stránka, kdežto u Web API se vrací JSON, XML nebo jiný formát.

Pro lepší pochopení lze nahlédnout na obrázek níže (obrázek 6)



Obrázek 6 Architektura ASP.NET Core Web API. Zdroj [5]

2.7 Angular

2.7.1 Účel technologie v aplikaci

Pro front-end část real-time chatovací aplikace byl použit framework Angular. Skrze něj je pak tvořeno uživatelské rozhraní pro interakci s ASP.NET Web API.

2.7.2 Popis technologie

Angular je open-source JavaScriptový framework napsaný v jazyce TypeScript. Je spravován společností Google a jeho hlavním účelem je vývoj jednostránkových aplikací. Jako framework, Angular poskytuje vývojářům standardní strukturu, se kterou mohou pracovat. [11]

"Angular" je souhrnný termín pro různé verze tohoto frameworku. Angular byl vyvinut v roce 2009 a poté se dočkal mnoha nových verzí. [11]

Nejprve existoval původní Angular, který se nazýval Angular 1 a nakonec byl známý jako AngularJS. Poté přišly verze Angular 2, 3, 4, 5, až nakonec přišla aktuální verze Angular 11, která byla vydána 11. 11. 2020. Každá následující verze Angularu vylepšuje svého předchůdce, opravuje chyby, řeší problémy a přizpůsobuje se rostoucí složitosti současných platforem. [11]

Přednostmi Angularu jsou: [11]

- Vlastní komponenty – Angular umožňuje uživatelům vytvářet vlastní komponenty, které mohou funkcionalitu spolu s logikou vykreslování zabalit do opakovaně použitelných částí.
- Data binding – Angular umožňuje uživatelům snadno přesouvat data z kódu JavaScriptu do View a reagovat na události uživatele, aniž by museli psát kód ručně.
- Dependency Injection – Angular umožňuje uživatelům psát modulární služby a injektovat je tam, kde jsou potřeba. Tím se zlepšuje testovatelnost a opakovaná použitelnost stejných služeb.
- Testování – testy jsou důležité nástroje a Angular byl od základu vytvořen s ohledem na testovatelnost.
- Komplexita – Angular je plnohodnotný framework a poskytuje hotová řešení pro komunikaci se serverem, směrování v rámci aplikace a další.

- Kompatibilita s prohlížeči – Angular je multiplatformní a kompatibilní s více prohlížeči. Aplikaci Angular lze obvykle spustit ve všech prohlížečích (např.: Chrome, Firefox) a operačních systémech, jako jsou Windows, macOS a Linux.

2.8 Bootstrap

2.8.1 Účel technologie v aplikaci

Aby byla výsledná aplikace – tedy její obsah, design a rozvržení responzivní (tzn. automatické přizpůsobování se zařízením napříč různými velikostmi obrazovek), byl použit framework Bootstrap.

Pro účely této aplikace byla použita šablona OnePage Single, což umožnilo měnit a přizpůsobit šablonu přesně dle požadavků. [12]

2.8.2 Popis technologie

Bootstrap je framework pro vývoj webu, který pomáhá vývojářům vytvářet webová rozhraní. Byl původně vytvořený firmou Twitter v roce 2011, konkrétně pány Markem Otto a Jacobem Thorntonem. Nyní je Bootstrap open source a stal se jedním z nejpoužívanějších frameworků pro vývoj webu. Samotný framework se skládá ze směsice HTML, CSS a JavaScriptu a poskytuje vývojářům všechny základní komponenty potřebné k vývoji plně funkčního webového uživatelského rozhraní. [13]

2.8.3 HTML

HTML (HyperText Markup Language) je jednoduchý značkovací jazyk, který se poměrně snadno učí a jakékoliv zařízení se základním webovým prohlížečem je schopno jej zobrazit. [14]

Počátky HTML sahají do 90. let kdy vzniknul krátký dokument, který detailně popsal několik prvků používání k vytvoření webové stránky. Většina z těchto prvků byla pro textový obsah, tzn. nadpisy, odstavce, seznamy nebo odkazy na jiné stránky. [14]

S postupným vývojem jazyka (další prvky a vylepšení) se zvyšovala i verze HTML. Poslední aktuální verze je HTML 5. [14]

HTML5 je přirozeným vývojem dřívějších verzí HTML a snaží se odrážet potřeby současných a budoucích webových stránek. Přebírá drtivou většinu funkcí od svých předchůdců a je zpětně

kompatibilní. HTML 5 přidává řadu nových elementů (např. article, native, figure) a funkcí (např. přehrávání audia a videa přímo v prohlížeči. Mimo to i další, komplexnější, které pomáhají vytvářet výkonné webové aplikace. [14]

2.8.4 CSS

Cascading Style Sheets je jazyk používaný pro popis vzhledu a stylu webových stránek. S CSS lze specifikovat barvy, fonty, rozměry, vzdálenosti a další vizuální vlastnosti elementů na webové stránce. [15]

CSS byl vyvinut jako nezávislý jazyk a je používán spolu s HTML pro tvorbu moderních webových stránek. Umožňuje oddělit obsah a styl, což usnadňuje údržbu a změny vzhledu stránky. [15]

2.8.5 JavaScript

JavaScript je dynamický skriptovací programovací jazyk. Je velmi často používán jako součást webových stránek, jejichž implementace umožní na klientské straně interagovat s uživatelem a vytvářet dynamické stránky. Jedná se o interpretovaný programovací jazyk s objektově orientovanými schopnostmi. [16]

JavaScript byl dříve znám jako LiveScript, nicméně společnost Netscape změnila jméno na JavaScript. Poprvé se JavaScript objevil v Netscape 2.0 v roce 1995. Univerzální jádro jazyka bylo zabudováno do Netscape, Internet Exploreru a dalších webových prohlížečů. [16]

Výhody použití JavaScriptu jsou například: [16]

- Méně interakce se serverem – lze validovat vstupy od uživatele zadané prostřednictvím formuláře ještě před odesláním na server
- Okamžitá zpětná vazba pro uživatele – nemusí se čekat na opětovné načtení stránky v případě nezadaných požadovaných údajů
- Zvýšená interaktivita – lze vytvářet rozhraní, která reagují např. při přejetí myši či stisku klávesy na klávesnici
- Bohatší rozhraní – pomoc Javascriptu lze implementovat např. posuvníky, formuláře, tlačítka, ovládací prvky, aj.

2.9 Microsoft SQL Server

2.9.1 Účel technologie v aplikaci

O ukládání, správu a zpracování dat výsledné aplikace se stará relační databázový systém od společnosti Microsoft (Microsoft SQL Server).

2.9.2 Popis technologie

Dušan Petković ve své knize tvrdí [17], že použití Microsoft SQL Serveru je nejvhodnější z následujících důvodů:[17]

1. SQL Server je určitě nejlepším systémem pro operační systémy Windows, protože má úzkou integraci a nízkou cenu. Vzhledem k obrovskému počtu nainstalovaných systémů Windows, který stále roste, je SQL Server široce používaný databázový systém.
2. Od SQL Server verze 2017, společnost Microsoft podporuje také Linuxové distribuce (např. Ubuntu, Red Hat Linux) a macOS.
3. Database Engine, jako relační databázová systémová komponenta, je pro používání nej-jednodušší databázový systém. Mimo dobře známého uživatelského stylu vzhledu jako Windows u Database Engine, nabízí Microsoft několik různých nástrojů, jak vytvořit objekty v databázi, vyladit databázové aplikace a spravovat administrátorské systémové úlohy.

SQL Server není pouze relační databázový systém. Je to platforma, která nespravuje pouze strukturovaná, polostrukturovaná a nestrukturovaná data, ale také nabízí komplexní, integrovaný operační a analytický software, který umožňuje organizacím spolehlivě spravovat naprosto nezbytné informace. [17]

2.10 UML

Nezbytnou součástí návrhu a vývoje každé aplikace by mělo být vizuální modelování (diagramy a modely), prostřednictvím kterého lze lépe pochopit a porozumět, jak bude celý systém fungovat. Nejinak tomu bylo v případě této aplikace. Pro tyto účely byl použit standardizovaný jazyk UML (Unified Modeling Language). [18]

UML poskytuje standardizovaný mechanismus k modelování softwarových systémů. Tento standard, který umožňuje vytvářet diagramy a související specifikace, ušel dlouhou cestu ve

zlepšování komunikace a zvyšování účasti všech zainteresovaných stran. Zvyšování této účasti zlepšuje kvalitu softwaru, snižuje množství chyb a podporuje snadnější přijetí výsledného řešení ze strany uživatelů. [18]

Dále také poskytuje notaci a pravidla pro popis aspektů, jako struktura, chování, vztahy a v neposlední řadě interakce mezi jednotlivými komponentami. [18]

II. PRAKTICKÁ ČÁST

3 ANALÝZA SYSTÉMU

3.1 Popis

Jak již bylo zmíněno v úvodu, účelem vzniku této práce je webová aplikace pro zákaznickou podporu. S návrhem každé aplikace je neodmyslitelně spjata prvotní analýza a požadavky na výsledný systém.

Tato kapitola se tedy v první části věnovala definici aktérů, detailnímu popisu funkčních a nefunkčních požadavků. Tyto požadavky byly následně graficky demonstrovány Use Case modelem neboli modelem případů užití.

Celá tato kapitola byla zpracována především prostřednictvím aplikace Enterprise Architect [19] a její výstupy vloženy do této práce formou obrázků či textů.

3.1.1 Uživatelé systému

Aplikace je určena pro uživatele, kteří budou mít problém s prostředkem ICT a budou jej chtít řešit. Nepřihlášení uživatelé mají možnost k vyřešení požadavku použít znalostní bázi, FAQ či svůj dotaz/problém vznést v real-time chatu.

V sekci helpdesk portálu, který je přístupný až po úspěšném zaregistrování a přihlášení, má uživatel možnost zadat požadavek formou tiketu.

Nedílnou součástí systému je administrátor, který primárně řeší uživatelské požadavky zadané prostřednictvím tiketů. Dále udržuje znalostní bázi, FAQ, poskytuje komunitní podporu v rámci real-time chatu.

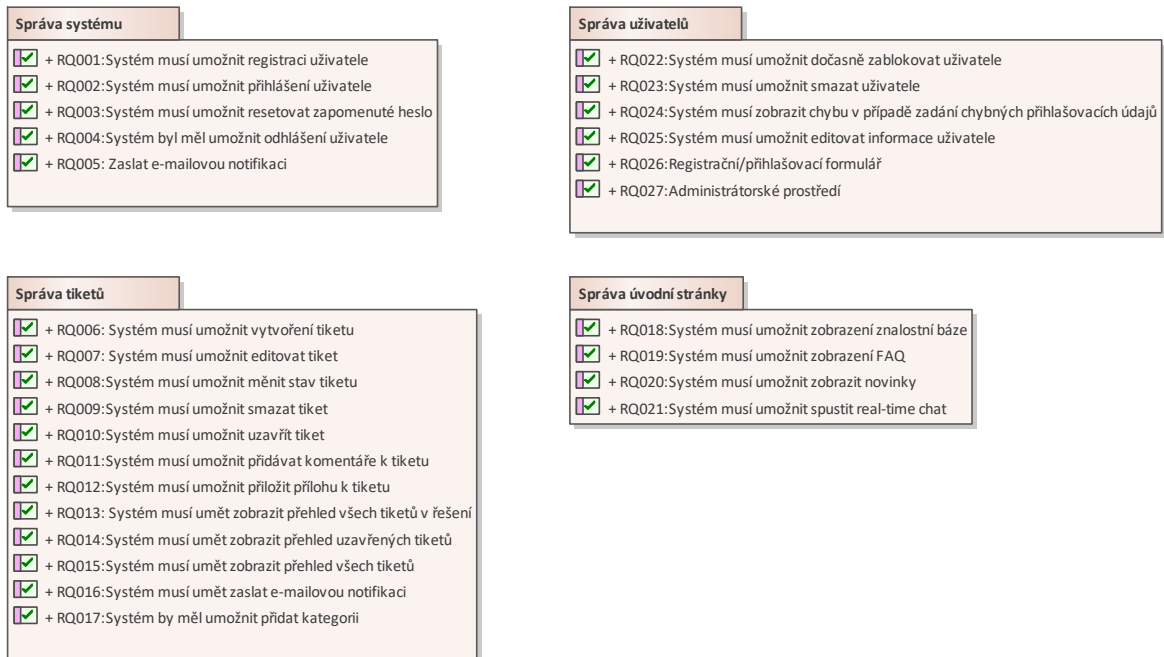
3.1.2 Funkční požadavky

Dle Arlowa a Neustadta [20] lze požadavek definovat jako „*specifikaci toho, co by mělo být implementováno*“. Rozlišujeme mezi funkčními a nefunkčními požadavky.

Funkční požadavky určují, jaké chování bude systém nabízet. Níže je uveden přehled a popis jednotlivých funkčních požadavků, které byly pro přehlednost rozděleny do následujících balíčků:

- Správa systému
- Správa tiketu

- Správa uživatelů
- Správa úvodní stránky
-



Obrázek 7 Funkční požadavky organizovány do balíčků

3.1.2.1 Správa systému

RQ001: Systém musí umožnit registraci uživatele

Uživatel bude schopen provést registraci zadáním uživatelského jména, e-mailu a hesla.

RQ002: Systém musí umožnit přihlášení uživatele

Po úspěšné registraci se bude schopen uživatel přihlásit.

RQ003: Systém musí umožnit resetovat zapomenuté heslo

V případě zapomenutého hesla bude možnost, prostřednictvím odkazu zasláního do e-mailu, resetovat uživatelské heslo, další možností je změna hesla v administrátorském prostředí.

RQ004: Systém by měl umožnit odhlášení uživatele

Přihlášený uživatel se bude schopen odhlásit ze systému.

3.1.2.2 *Správa tiketů*

RQ006: Systém musí umožnit vytvoření tiketu

Systém musí umožnit založení nového tiketu, včetně názvu, kategorie, popisu problému a vložení přílohy.

RQ007: Systém musí umožnit editovat tiket

Systém musí umožnit editování tiketu. Tedy pokud uživatel či administrátor potřebuje měnit název tiketu, kategorii či popis.

RQ008: Systém musí umožnit měnit stav tiketu

Systém musí administrátorovi umožnit měnit stav tiketu. Stavby mohou být: běžící, vyřešený, uzavřený.

RQ009: Systém musí umožnit smazat tiket

Systém musí umožnit uživateli i administrátorovi smazat tiket. Přihlášený uživatel může mazat pouze své tikety, kdežto administrátor jakýkoliv tiket.

RQ010: Systém musí umožnit uzavřít tiket

Systém musí umožnit uživateli i administrátorovi uzavřít tiket. Přihlášený uživatel může uzavírat pouze své tikety, administrátor jakýkoliv tiket.

RQ011: Systém musí umožnit přidávat komentáře k tiketu

Systém musí umožnit přidávat komentáře k tiketu, tedy pro upřesnění požadavku, poznámek či připomínek.

RQ012: Systém musí umožnit přiložit přílohu k tiketu

Systém musí umožnit přidat přílohu k tiketu. Přílohou může být textový nebo obrázkový formát či hypertextový odkaz odkazující na externí úložiště.

RQ013: Systém musí umět zobrazit přehled tiketů v řešení

Systém musí umět zobrazit přehled všech tiketů, které jsou aktuálně v řešení. Tento přehled je dostupný pouze pro administrátora systému.

RQ014: Systém musí umět zobrazit přehled uzavřených tiketů

Systém musí umět zobrazit přehled všech tiketů, které jsou uzavřeny. Tento přehled je dostupný pouze pro administrátora systému.

RQ015: Systém musí umět zobrazit přehled všech tiketů

Systém musí umět zobrazit přehled všech tiketů. Tento přehled je dostupný pouze administrátorovi systému. Přihlášený uživatel vidí pouze přehled svých tiketů.

RQ016: Systém musí umět zaslat e-mailovou notifikaci

Systém musí umět zaslat e-mail v případě změny stavu tiketu nebo přidání komentáře k tiketu.

RQ017: Systém musí umožnit přidat kategorii

Systém musí umožnit přidání nové kategorie tiketu. Kategorii může přidávat pouze administrátor systému.

3.1.2.3 Správa úvodní stránky**RQ018: Systém musí umožnit zobrazení znalostní báze**

Systém musí umožnit i nepřihlášeným uživatelům zobrazit znalostní bázi, kde se nachází informace, které mohou vést k vyřešení problému bez použití helpdesk systému.

RQ019: Systém musí umožnit zobrazení FAQ

Systém musí umožnit i nepřihlášeným uživatelům zobrazit FAQ (tedy často kladené otázky), jejichž odpovědi mohou vést k vyřešení problému bez použití helpdesk systému.

RQ020: Systém musí umožnit zobrazit novinky

Systém musí umožnit i nepřihlášeným uživatelům zobrazit novinky, které informují o aktualitách systému

RQ021: Systém musí umožnit spustit real-time chat

Systém musí umožnit i nepřihlášeným uživatelům spustit real-time komunitní chat, kde lze v reálném čase chatovat s ostatními návštěvníky.

3.1.2.4 *Správa uživatelů*

RQ005: Systém by měl umožnit přidat uživatele

Systém musí administrátorovi umožnit ručně vytvořit uživatele na základě jména, příjmení, e-mailu a hesla.

RQ022: Systém musí umožnit dočasně zablokovat uživatele

Systém musí umožnit administrátorovi dočasně zablokovat uživatele, tento uživatel nebude po tuto dobu schopen přihlášení do helpdesk systému.

RQ023: Systém musí umožnit smazat uživatele

Systém musí umožnit administrátorovi trvale smazat uživatele. Uživatel se již nebude moci přihlásit do helpdesk systému.

RQ024: Systém musí zobrazit chybu v případě zadání chybných přihlašovacích údajů

Systém musí zobrazit chybovou hlášku informující o tom, že uživatel chybně zadal přihlašovací údaje.

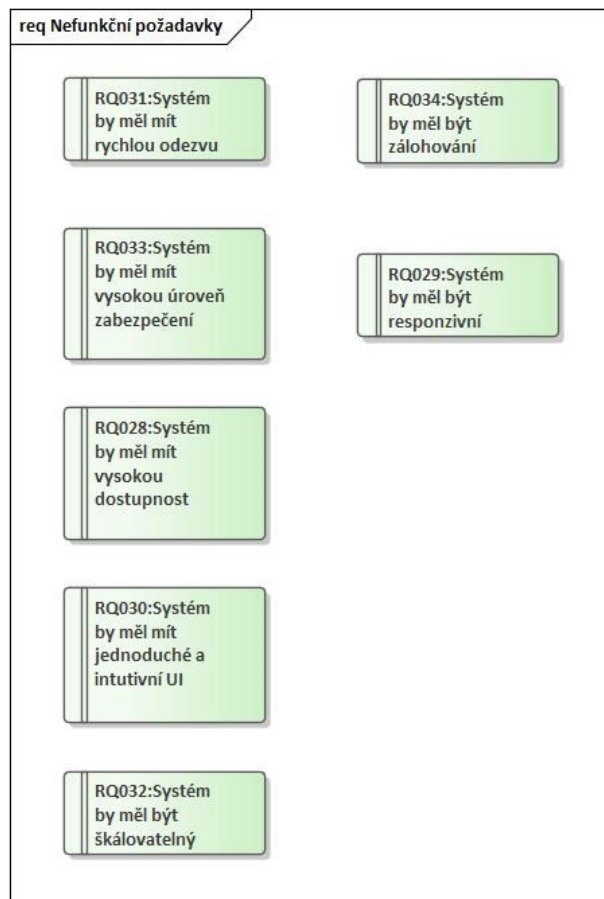
RQ025: Systém musí umožnit editovat informace uživatele

Systém musí administrátorovi umožnit editovat informace o uživateli (jméno, příjmení, e-mail, heslo).

RQ026: Registrační/přihlašovací formulář**RQ027: Administrátorské prostředí**

3.1.3 Nefunkční požadavky

Nefunkční požadavky specifikují vlastnosti nebo omezující podmínky daného systému. Níže je uveden přehled a následný popis jednotlivých nefunkčních požadavků. [20]



Obrázek 8 Nefunkční požadavky

RQ028: Systém by měl mít vysokou dostupnost

Systém by měl být dostupný 24/7. S tím souvisí zajištění bezpečnosti systému proti vnějším faktorům tak, aby byly všechny komponenty zálohované a redundantní.

RQ029: Systém by měl být responzivní

Systém by měl mít přizpůsobený vzhled pro většinu nejrůznějších velikostí obrazovek

RQ030: Systém by měl mít jednoduché a intuitivní UI

Systém by měl mít jednoduché, moderní a intuitivní uživatelské rozhraní

RQ031: Systém by měl mít rychlou odezvu

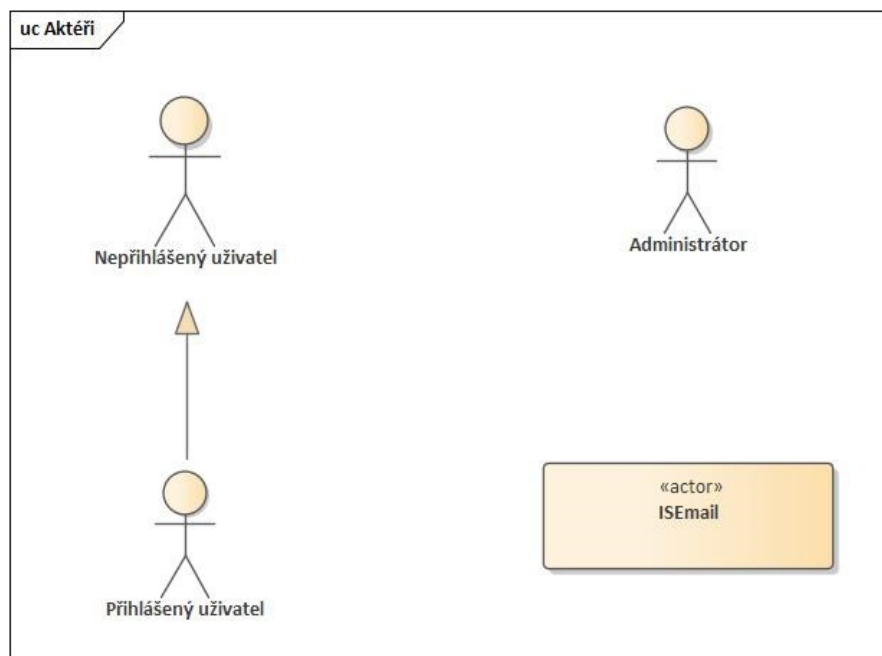
Systém by měl být schopen rychle a efektivně reagovat na požadavky uživatelů a poskytnout odpověď v co možná nejkratším čase.

RQ032: Systém by měl být škálovatelný

Systém by měl být schopen se přizpůsobit změnám v případě zvýšeného množství požadavků od uživatelů bez ztráty výkonu a se zajištěním plynulosti a spolehlivosti.

3.1.4 Aktéři

Aktéři jsou v systému entity, které jej využívají. Pro tuto aplikaci byli zvoleni tito 4 aktéři:



Obrázek 9 Přehled aktérů v systému

Nepřihlášený uživatel

Nepřihlášený uživatel má možnost zobrazit novinky, znalostní bázi, FAQ a spustit real-time komunitní chat.

Přihlášený uživatel

Přihlášený uživatel má možnost vstoupit do sekce helpdesku, kde může zakládat, editovat, mazat tikety nebo přidávat komentáře k vlastním již existujícím tiketům. Dále si pak změnit heslo, jméno nebo e-mailovou adresu.

Administrátor

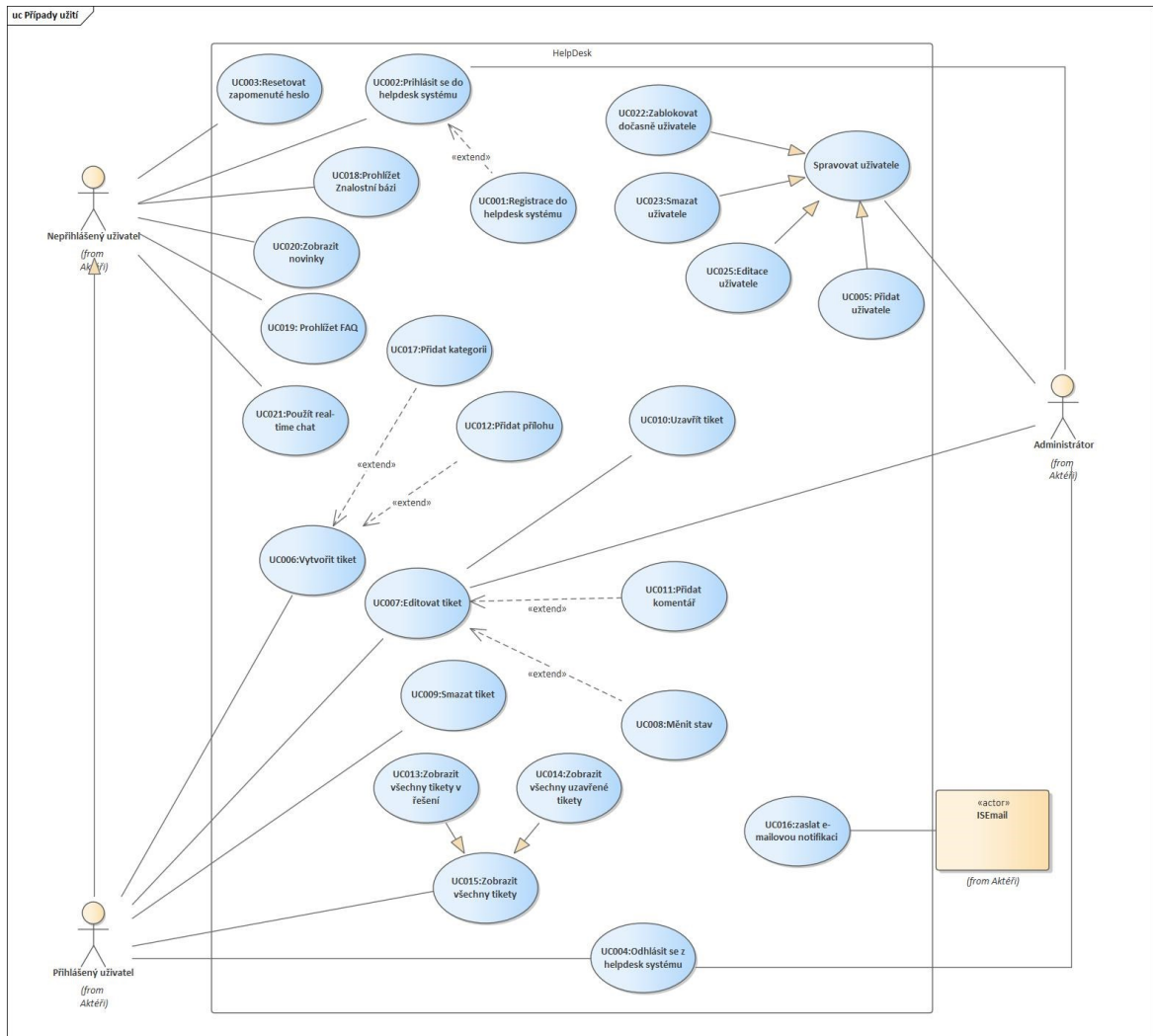
Administrátor systému je schopen spravovat všechny tikety v systému, odpovídat na jednotlivé tikety formou komentáře k ticketu, měnit jejich stav či mazat. Dále může přidávat nové uživatele na základě jména, e-mailu, hesla, kategorií tiketů. V neposlední řadě systém umožňuje administrátorovi nastavit parametry pro konfiguraci poštovního serveru, který slouží k odeslání notifikací.

E-mail

E-mailová notifikace je odeslána v případě úspěšného založení ticketu, jeho změně či smazání. Dále pak prostřednictvím e-mailu dochází k zaslání resetovacího linku při zapomenutém hesle uživatele.

3.1.5 Use Case model

Use Case model, nebo také model případů užití, slouží k interakci mezi aktérem a systémem. Níže lze nalézt obrázek kompletního Use Case modelu pro helpdesk portál.



Obrázek 10 Use Case model helpdesk portálu

3.1.6 Specifikace případů užití

Specifikace případů užití má sloužit k popisu funkcionalit, které se v systému nacházejí. Popis je z pohledu uživatelů, a jejich interakcí se systémem jako takovým. Pro ukázkou a představu byly níže uvedeny dva scénáře případů užití: Přihlášení uživatele do helpdesk portálu a Vytvoření tiketu v helpdesk portálu.

Tabulka 2 Primární scénář – Přihlášení uživatele do helpdesk portálu

Název: Přihlášení uživatele do helpdesk portálu		
ID: UC002		
<u>Charakteristika:</u> Zachycení případu užití pro přihlášení uživatele do systému		
<u>Primární aktér:</u> Nepřihlášený uživatel		
<u>Vedlejší aktéři:</u> Nejsou		
<u>Vstupní podmínky:</u> Nepřihlášený uživatel je připojený k internetu Nepřihlášený uživatel má v databázi vytvořen uživatelský účet Nepřihlášený uživatel není aktuálně přihlášen Uživatel není dočasně zablokován administrátorem		
<u>Výstupní podmínky:</u> Systém vytvoří novou relaci pro přihlášeného uživatele Systém zobrazí domovskou obrazovku přihlášeného uživatele helpdesk portálu		
<u>Hlavní scénář:</u>		
Krok	Aktér/Systém	Popis
1	A	Tento případ užití začíná, když nepřihlášený uživatel zvolí volbu přihlásit
2	S	Systém zobrazí formulář pro zadání přístupových údajů – uživatelské jméno (e-mailovou adresu) a heslo

3	A	Nepřihlášený uživatel vyplní potřebné přihlašovací údaje – uživatelské jméno (e-mailovou adresu) a heslo Alternate: 3a. Uživatel zatím nemá přihlašovací údaje
4	S	Systém ověří zadané údaje Alternate: 4a. Přihlášení se nezdařilo
5	S	Systém zobrazí potvrzení o úspěšném přihlášení uživatele
6	S	Systém zobrazí domovskou stránku přihlášeného uživatele.
7	A	Případ užití končí.
<u>Alternativní scénáře:</u>		
UC002a – Uživatel zatím nemá přihlašovací údaje		
UC002b – Přihlášení se nezdařilo		

Tabulka 3 Alternativní scénář – Uživatel zatím nemá přihlašovací údaje

Název – Alternativní scénář: Uživatel zatím nemá přihlašovací údaje		
ID: UC002a		
<u>Charakteristika:</u> Zachycení alternativního toku případu užití ve chvíli, kdy uživatel zatím nemá přihlašovací údaje.		
<u>Alternativní scénář:</u>		
Krok	Aktér/Systém	Popis
1	S	<extend> UC001: Registrace do helpdesk portálu

Tabulka 4 Alternativní scénář – Přihlášení se nezdařilo

Název – Alternativní scénář: Přihlášení se nezdařilo		
ID: UC002b		
<u>Charakteristika:</u> Zachycení alternativního toku případu užití ve chvíli, kdy se přihlášení nezdařilo.		
<u>Alternativní scénář:</u>		
Krok	Aktér/Systém	Popis
1	S	Ověření zadaných přihlašovacích údajů se nezdařilo.
2	S	System zobrazí informativní hlášku o nezdařeném přihlášení
3	S	System znovu načte přihlašovací obrazovku
4	A	Uživatel opakovaně zadává přihlašovací údaje

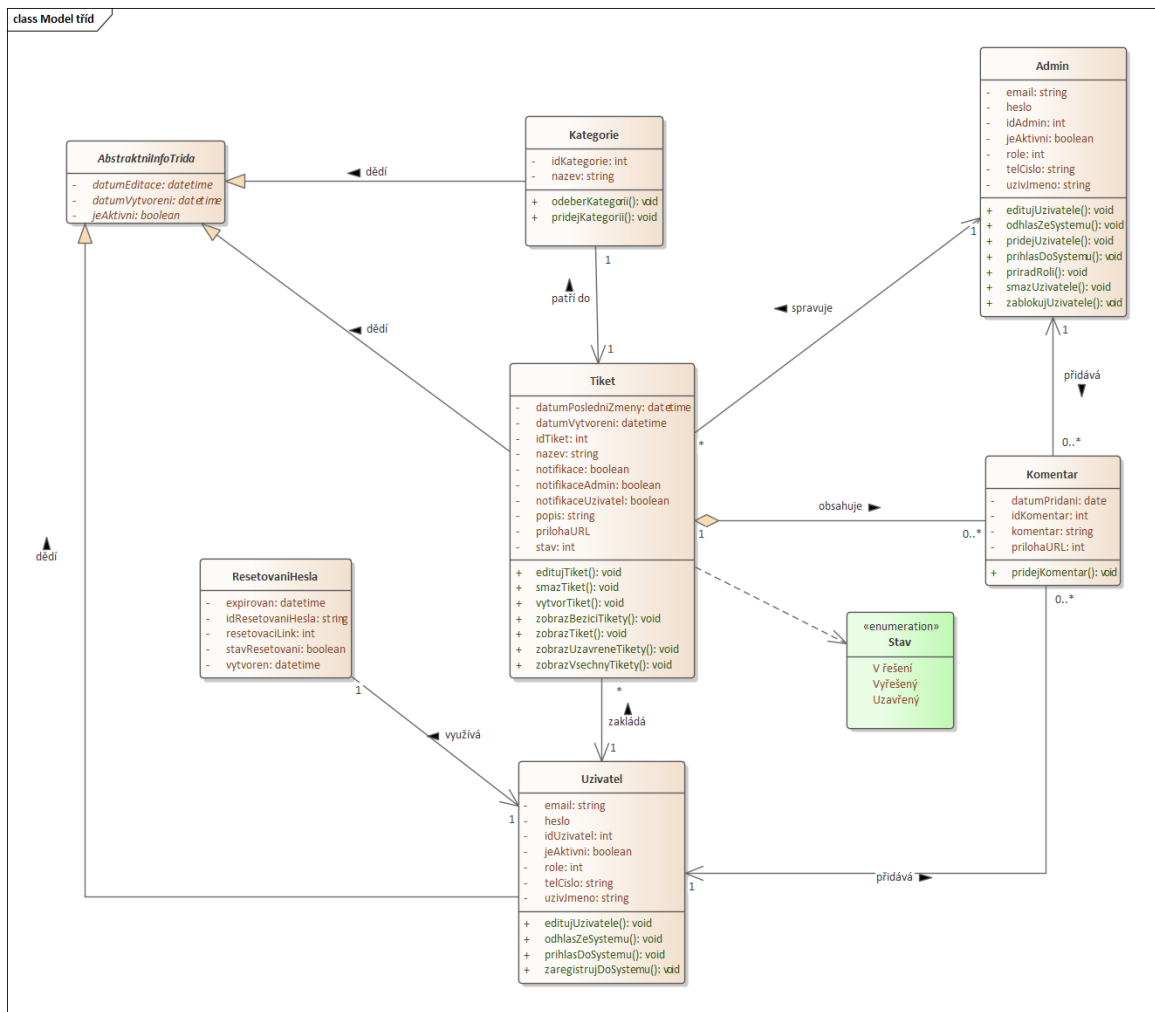
Tabulka 5 Primární scénář – Vytvoření tiketu v helpdesk portálu

Název: Vytvoření tiketu v helpdesk portálu
ID: UC006
<u>Charakteristika:</u> Zachycení případu užití pro vytvoření nového tiketu v helpdesk portálu
<u>Primární aktér:</u> Přihlášený uživatel
<u>Vedlejší aktéři:</u> Nejsou
<u>Vstupní podmínky:</u> Uživatel je přihlášen do systému (autorizován i autentizován)

<u>Výstupní podmínky:</u>		
Systém založí nový tiket do systému		
Uživatel je informován o úspěšném založení tiketu		
<u>Hlavní scénář:</u>		
Krok	Aktér/Systém	Popis
1	A	Tento případ užití začíná, když přihlášený uživatel zvolí volbu vytvořit nový tiket.
2	A	Systém zobrazí formulář pro zadání informací k tiketu ohledně problému – Název tiketu, kategorie, popis problému a příloha.
3	S	Systém ověří, zdali jsou vyplněné požadovaná pole a zdali jsou vstupy validní. Alternate: 3a. Uživatel nevyplnil požadované vstupy nebo zadal nevalidní vstupy.
4	S	Systém zobrazí informativní hlášku o úspěšném založení tiketu do systému.
5	S	Systém zobrazí potvrzení o založení tiketu.
6	S	Systém zobrazí přehled založených tiketů uživatelem.
7	A	Případ užití končí.
<u>Alternativní scénáře:</u>		
UC006a – Uživatel nevyplnil požadované vstupy nebo zadal nevalidní vstupy.		

Tabulka 6 Alternativní scénář: Přihlášení se nezdařilo

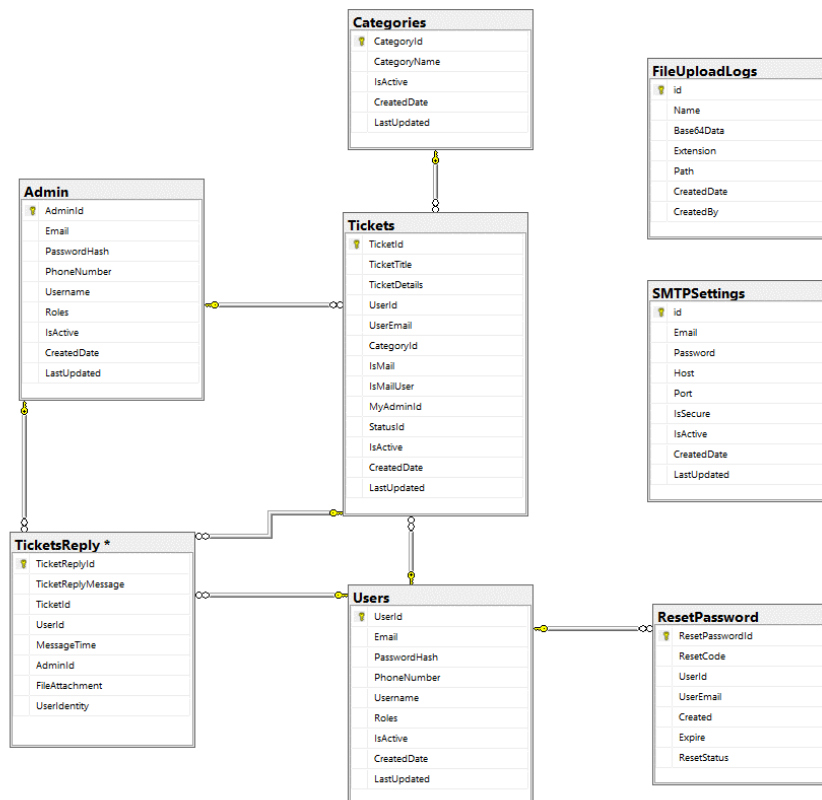
Název – Alternativní scénář: Přihlášení se nezdařilo		
ID: UC006a		
<u>Charakteristika:</u> Zachycení alternativního toku případu užití ve chvíli, kdy uživatel nevyplnil požadované vstupy nebo zadal nevalidní vstupy.		
<u>Alternativní scénář:</u>		
Krok	Aktér/Systemém	Popis
1	S	Ověření zadaných vstupů se nezdařilo.
2	S	System zobrazí informativní hlášku v polích, které nebyly validně vyplněny.
3	S	System vyzve uživatele k opětovnému vyplnění formulářových polí.
4	A	Uživatel opakovaně vyplňuje požadovaná pole.



Obrázek 11 Diagram tříd

3.2 Databáze helpdesk portálu

Databázové schéma bylo vytvořeno metodou „Code First“. Tuto metodu doporučuje Wade [21] pro nově vzniklé projekty, pokud zatím neexistuje databáze (původní aplikace sice databázi měla, nicméně pro výslednou aplikaci byla nutná rozsáhlá modifikace, a proto byla vytvořena nová, vycházející z analýzy systému). V „Code First“ přístupu se nejprve definují modely a logika aplikace přímo v kódu, a až následně poté probíhá generování databáze. Pokud by v budoucnu výsledná aplikace, potažmo helpdesk portál, byla provozována na jiném než Microsoft SQL serveru, je využití přístupu Code First ideální.



Obrázek 12 Databázové schéma helpdesk portálu

3.2.1 Users

Tato třída reprezentuje přihlášeného uživatele, který může založit vícero tiketů, přidávat komentářů k tiketu do doby, než je tiket uzavřen a uživateli je zaslána notifikace při jakékoliv změně založeného tiketu.

Třída Users má následující atributy:

- **UserId** – unikátní primární identifikátor uživatele
- **Email** – e-mail uživatele, který slouží jako uživatelské jméno pro přihlášení do systému
- **PasswordHash** – heslo uživatele, které slouží pro přihlášení a do databáze je ukládáno v hashované podobě
- **PhoneNumber** – telefonní kontakt na uživatele, sloužící pouze pro informaci v případě nutnosti kontaktování telefonicky
- **Username** – uživatelské jméno uživatele, sloužící pro přesnější identifikaci uživatele v systému

- **Roles** tento atribut určuje, že se jedná o uživatele s právy běžného uživatele. Po registraci uživatele dostává v databázi automaticky hodnotu „user“
- **IsActive** – tento atribut určuje, zdali účet uživatele je nebo není aktivní. Pokud je atribut false, nemůže se uživatel do systému přihlásit
- **CreatedDate** – atribut, který nese informaci o datu vytvoření uživatele v systému
- **LastUpdated** – atribut, který nese informaci o datu poslední úpravy vlastností uživatele v systému

3.2.2 Admin

Tato třída reprezentuje administrátora systému, který spravuje tikety, odpovídá na komentáře, je schopen vytvářet uživatele, editovat jejich vlastnosti, či uživatele v systému blokovat nebo mazat. Administrátor je v systému pouze jeden a vytvoření nového je podmíněno přímým vytvořením v databázi.

Třída Admin má následující atributy:

- **AdminId** – unikátní primární identifikátor administrátora
- **Username** – uživatelské jméno administrátora, které slouží pouze pro identifikaci uživatele v systému
- **Email** – email administrátora, který slouží pro registraci a následné přihlášení do systému
- **PasswordHash** – heslo administrátora, které slouží pro přihlášení do admin sekce
- **PhoneNumber** – telefonní kontakt na administrátora, sloužící pouze pro informaci v případě nutnosti kontaktování telefonicky
- **Roles** – tento atribut určuje, že se jedná o administrátora. Atribut je naplněn hodnotou „admin“. Administrátor může být vytvořen pouze v databázi.
- **IsActive** – tento atribut určuje, zdali účet administrátora je nebo není aktivní. Pokud není aktivní, nemůže se do systému přihlásit
- **CreatedDate** – atribut, který nese informaci o datu vytvoření administrátora v systému
- **LastUpdated** – atribut, který nese informaci o datu poslední úpravy vlastností administrátora v systému

3.2.3 Tickets

- **TicketId** – unikátní identifikátor administrátora
- **TicketTitle** – atribut pro název tiketu
- **TicketDetails** – určen pro detailnější popis problému
- **UserEmail** – e-mail uživatele, který tiket zadal
- **IsMail** – pokud administrátor zapne tuto notifikaci, bude mu na e-mail zaslána notifikace při každé změně v tiketu
- **IsMailUser** – pokud uživatel tento atribut zvolí, bude mu na e-mail zaslána notifikace při každé změně v tiketu
- **StatusId** – stav tiketu (v řešení, vyřešený a uzavřený)
- **CreatedDate** – atribut, který nese informaci o datu vytvoření tiketu v systému
- **LastUpdated** – atribut, který nese informaci o datu poslední úpravy tiketu v systému

3.2.4 TicketsReply

Tato třída umožňuje administrátorovi i uživatelům vkládat podrobnější či upřesňující informace formou komentáře k tiketu. Komentáře jsou součástí tiketu a je možné je přikládat průběžně, ale pouze do doby, než je tiket uzavřen.

Třída TicketsReply má následující atributy:

- **TicketReplyId** – unikátní primární identifikátor komentáře
- **TicketReplyMessage** – text komentáře
- **MessageTime** – datum a čas přidání komentáře ke konkrétnímu tiketu
- **FileAttachment** – příloha související s konkrétním komentářem v tiketu. Podporované formáty jsou .jpeg, .png, .jpg, .doc, .docx, .xlsx, .pptx, .zip, .txt, .pdf
- **UserIdentity** – identifikace uživatele, který komentář/e přidal (uživatel či administrátor)

3.2.5 Categories

Tato třída slouží pro určení kategorie zadávaného problému (např. hardware, software, mobilní služby apod.). Její přidání je umožněno pouze administrátorovi systému. Kategorie je součástí tiketu při jeho zakládání.

Třída Categories má následující atributy

- **CategoryId** – unikátní identifikátor kategorie

- **CategoryName** – atribut pro volbu kategorie tiketu
- **IsActive** – tento atribut určuje, zdali je kategorie aktivní, pokud není, nelze jí zvolit při zakládání tiketu
- **CreatedDate** – atribut, který nese informaci o datu vytvoření kategorie
- **LastUpdated** – atribut, který nese informaci o poslední změně kategorie (názvu)

3.2.6 ResetPassword

Tato třída slouží pro resetování hesla. Uživateli je do e-mailu, který zadal při registraci zaslán resetovací link, pomocí kterého

Třída ResetPassword má následující atributy:

- **ResetPasswordId** – unikátní primární identifikátor
- **ResetCode** – resetovací odkaz, který je zaslán e-mailem
- **UserEmail** – e-mail uživatele, který chce změnit heslo
- **CreatedDate** – datum vytvoření žádosti o resetování hesla
- **ExpireDate** – doba, po kterou je odkaz pro resetování hesla validní
- **ResetStatus** – informace o tom, zdali je heslo již vyresetováno

3.2.7 SMTPSettings

Tato třída slouží k zasílání notifikací v případě jakékoliv změny v tiketu (pokud ji uživatel zvolí). Notifikace je zaslána uživateli do e-mailu a obsahuje informaci o tom, že došlo ke změně v tiketu. Dále je určena k zasílání resetovací linku, v případě zapomenutí či odcizení hesla. Nastavení této třídy je plně v kompetenci administrátora.

Třída SMTPSettings má následující atributy:

- **Id** – unikátní primární identifikátor
- **Email** – e-mailová adresa, prostřednictvím které se budou e-maily zasílat
- **Password** – heslo k e-mailové adrese
- **Host** – hostitelský server
- **Port** – komunikační port, administrátor má možnost nastavit, zdali bude komunikace probíhat prostřednictvím SSL/TLS na portu 465 nebo standardním nezabezpečeným portem

3.2.8 FileUploadLogs

Účel této třídy spočívá v uchovávání informací o nahrávání souborů do systému. Přesněji řečeno se jedná o informace o nahrávání příloh k jednotlivým tiketům.

Třída FileUploadLogs má následující atributy

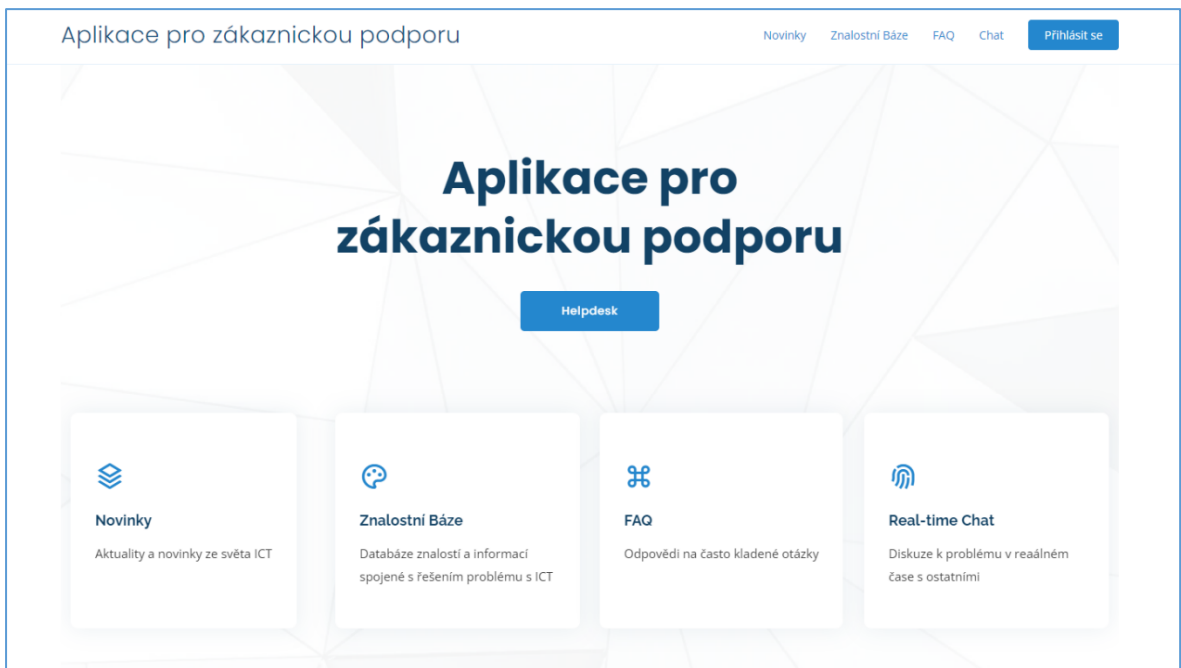
- **Id** – unikátní primární identifikátor
- **Name** – název přílohy
- **Base64Data** – aby bylo nahrávání přílohy nezávislé na konkrétním znakovém kódování a aby byla zajištěna spolehlivost a bezztrátovost přenosu informací, je pro převod použito kódování Base64
- **Extension** – typ nahraného souboru, podporované formáty jsou .jpeg, .png, .jpg, .doc, .docx, .xlsx, .pptx, .zip, .txt, .pdf
- **Path** – cesta, kde se nahraná příloha fyzicky nachází na serveru
- **CreatedDate** – datum, kdy byla příloha nahrána
- **CreatedBy** – identifikace uživatele, který přílohu nahrál

4 APLIKACE

V této části bude podrobně popsána výsledná aplikace, která byla rozdělena na 2 části:

- **volně přístupnou bez nutnosti registrace** (novinky, znalostní báze, FAQ a real-time chat)
- **nutná registrace** (helpdesk portál)

4.1 Úvodní stránka



Obrázek 13 Úvodní stránka celé výsledné aplikace

Jako první se uživateli zobrazí úvodní stránka, kde lze nalézt:

- Novinky
- Znalostní Bázi
- FAQ
- Real-time chat

4.2 Real-time chat

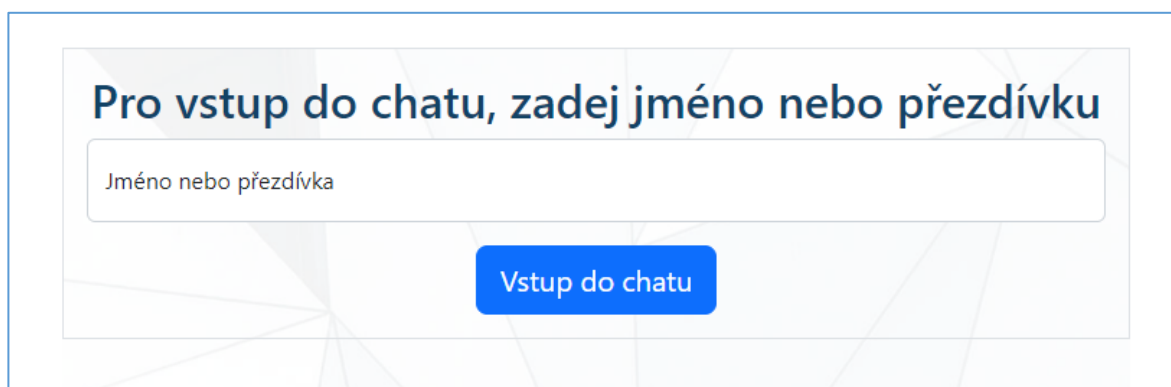
Hned v úvodu je dobré zmínit, že pod pojmem real-time chat, je v této práci myšlen real-time komunitní chat, kde mohou uživatelé po přihlášení napsat svůj dotaz či problém. Ostatní

přihlášení uživatelé chatu pak mohou na tyto otázky a dotazy odpovídat. Smyslem je poskytování komunitní podpory, která se neomezuje pouze na interakci mezi uživatelem a administrátorem. Je totiž mnohem pravděpodobnější, že se v případě komunitního řešení problému najdou uživatelé, kteří se již se stejným nebo podobným problémem v minulosti setkali. Navíc je díky komunitnímu chatu více rozvíjen sociální aspekt.

Jak již bylo zmíněno dříve, jako back-end, pro vývoj real-time chatovací aplikace, byl použit framework ASP.NET Web API, pro front-end byl využit framework Angular.

Vstup do chatu začíná kliknutím na možnost „Chat“ nebo „Real-time Chat“ na úvodní stránce webové aplikace. Uživatel je následně přesměrován na přihlašovací obrazovku pro vstup do real-time chatu. Pro tyto účely není nutná registrace uživatele, stačí pouze zvolit jméno či přezdívku (v kódu je ošetřeno, aby minimální počet znaků pro jméno či přezdívku, byl 3 a maximální 12) a poté pomocí tlačítka „Vstup do chatu“ vstoupit do chatovací místnosti.

Do chatu se nelze připojit stejným jménem či přezdívkou, která se již v chatu nachází – je tedy nutné, aby vždy byla zvoleno unikátní jméno či přezdívka.

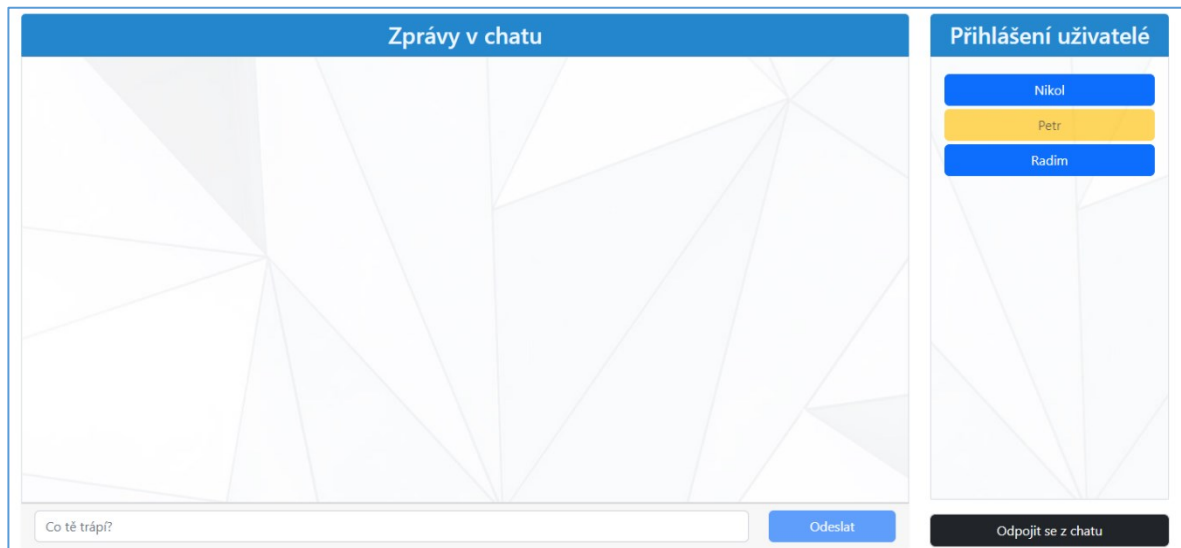


Obrázek 14 Přihlašovací formulář pro vstup do real-time chatu

Po úspěšném přihlášení je uživateli zobrazeno okno samotné chatovací aplikace (obrázek 15). V levé části jsou zobrazeny jednotlivé zprávy, které lze odeslat pomocí tlačítka „Odeslat“. Zpráva poté obsahuje informaci o konkrétním odesílateli a aktuálním čase odeslání zprávy

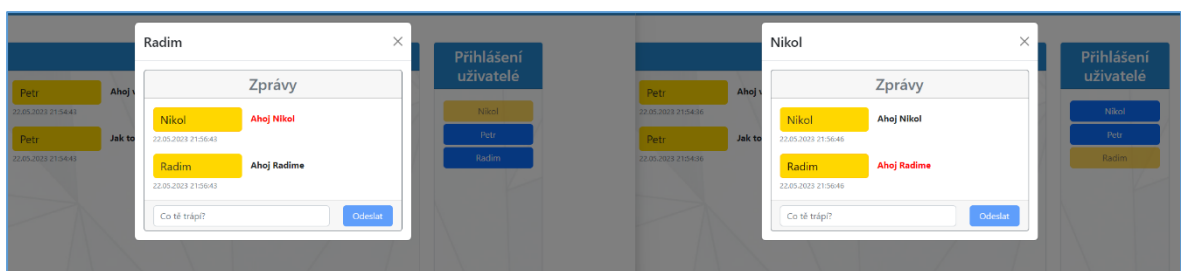
V pravé části je poté přehled aktuálně přihlášených uživatelů. Pro vizuální rozlišení uživatelů, je konkrétní uživatel (který se přihlašuje) vždy označen zlatou barvou, ostatní účastníci chatu poté barvou modrou.

Chat lze opustit prostřednictvím černého tlačítka „*Odpojit se z chatu*“. Po této akci je uživatel odebrán ze seznamu aktuálně přihlášených uživatelů a jeho jméno či přezdívka uvolněna.



Obrázek 15 Obrazovka real-time chatovací aplikace

Další možností, kterou chat nabízí, je zaslání soukromé zprávy jinému účastníkovi chatu. Toto lze provést pomocí dvojkliku myši na příslušného uživatele v seznamu „*Přihlášení uživatelé*“. Následně se zobrazí nové okno, kde po zadání textu zprávy, přijde daná zpráva pouze zvolenému uživateli (obrázek 16).



Obrázek 16 Zaslání soukromé zprávy konkrétnímu uživateli chatu

4.3 Helpdesk portál

Kostra helpdesk portálu, jak již bylo zmíněno dříve, vychází z existujícího řešení. Nicméně toto řešení bylo vylepšeno, optimalizováno a přizpůsobeno požadavkům, na základě provedené analýzy.

Provedená vylepšení:

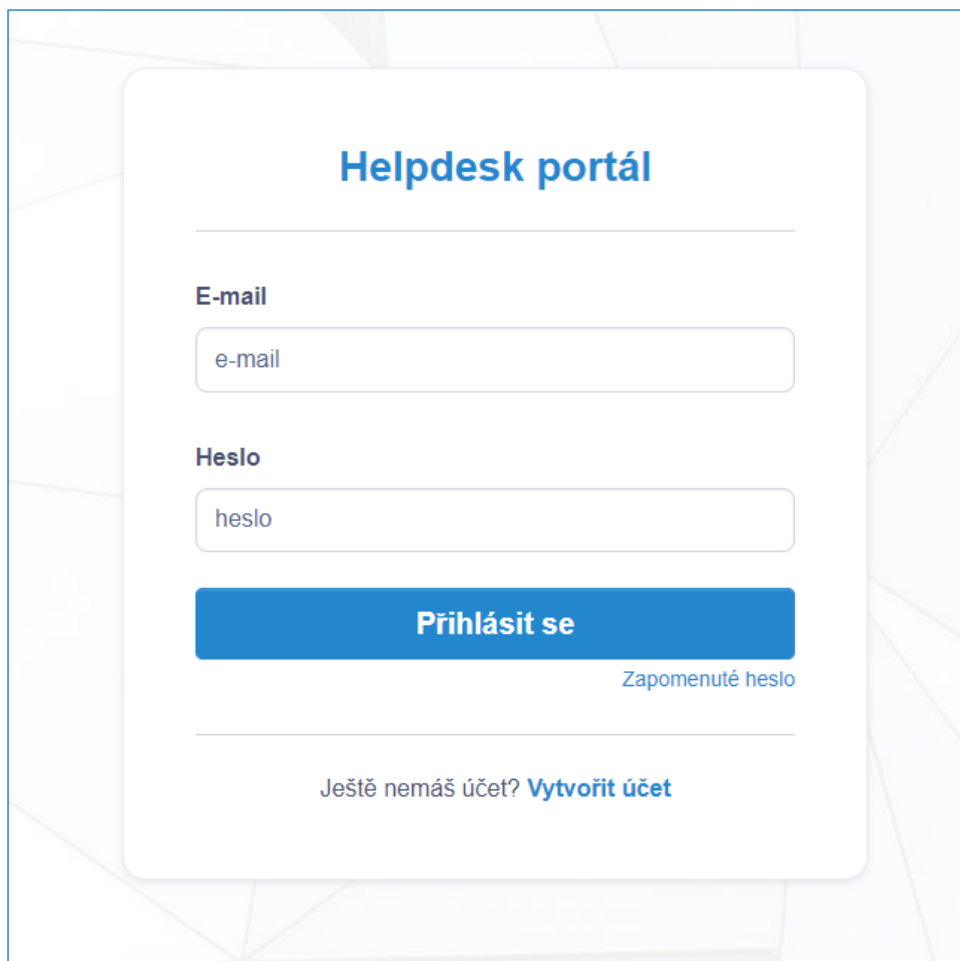
- původní řešení bylo vyvinuto v .NET 5, byl tedy proveden upgrade na aktuálnější .NET 6
- nova struktura databáze, včetně atributů, která vznikla na základě analýzy systému (původní řešení např. vůbec nepracovalo, v rámci tabulek, s relacemi)
- přizpůsobeno uživatelské rozhraní a celkový vzhled aplikace zapadající do jednotného vizuální vzhledu s real-time chatem a zdrojem informací
- citlivá data (hesla) jsou nově ukládána do databáze pomocí hashovacího algoritmu BCrypt, původní verze používala SHA-256
- aplikovány požadavky na komplexnost hesel. Heslo nyní musí mít minimálně 10 znaků a obsahovat malé písmeno, velké písmeno, číslo a speciální znak. Původní řešení tuto komplexnost vůbec nemělo. Jediným omezením byla minimální délka 6 znaků
- implementován návrhový vzor Dependency Injection pro oddělení závislosti mezi komponentami prostřednictvím interface, což mimo jiné zlepšuje následnou flexibilitu kódu při nahrazování či změně rozhraní bez nutnosti modifikace kódu
- implementováno asynchronní programování (pomocí klíčových slov „async“ a „await“). Díky tomu dochází k zrychlení výkonu aplikace, jelikož asynchronní programování umožňuje efektivněji využívat vlákna.

Samotný helpdesk portál je rozdělen na dvě sekce:

- Uživatelskou sekci
- Administrátorskou sekci

4.3.1 Uživatelská sekce

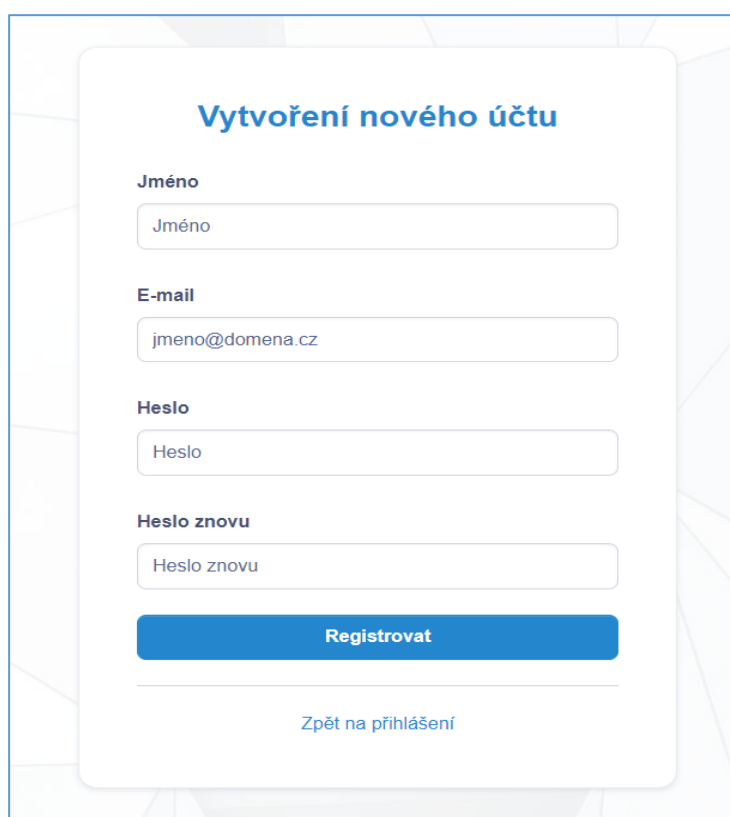
Nejprve byla popsána uživatelská sekce. Snahou uživatelské sekce je co nejvíce zjednodušit zakládání požadavků. Pro vstup do uživatelské sekce je nutná nejprve registrace nového uživatele. Tu lze provést přes kliknutí na tlačítko „Vytvořit účet“ (obrázek 17).



The image shows a login form for the 'Helpdesk portál'. The form is centered on a light blue background with a subtle geometric pattern. At the top, the title 'Helpdesk portál' is displayed in a bold, blue font. Below the title is a horizontal line. The form contains two input fields: 'E-mail' with the placeholder text 'e-mail' and 'Heslo' with the placeholder text 'heslo'. Below these fields is a prominent blue button labeled 'Přihlásit se'. To the right of the button, there is a link labeled 'Zapomenuté heslo'. At the bottom of the form, there is a horizontal line and a text prompt: 'Ještě nemáš účet? [Vytvořit účet](#)'.

Obrázek 17 Přihlašovací obrazovka pro vstup do helpdesk portálu

Mandatorní údaje pro úspěšnou registraci (obrázek 18) jsou: jméno, e-mail, a heslo. Pro eliminování potenciálních překlepů v zadaném hesle, je zde ještě druhé pole pro ověření tohoto hesla. V případě pokusu o registrování již existující e-mailové adresy je zobrazena hláška s touto informací a žádostí o zadání jiné, zatím neregistrované e-mailové adresy. Dále bylo implementováno ověření vstupů (pomocí JavaScriptu) na straně klienta – e-mail musí být ve správném formátu, heslo musí splňovat minimální požadavky na délku (10 znaků) a komplexnost (malé písmeno, velké písmeno, číslici a speciální znak), formulářová pole nesmí obsahovat nevyplněná pole. Tato validace umožňuje opravu chyb ještě před odesláním dat na server a zkracuje tak odezvu, jelikož neplatná data se na serveru vůbec nezpracovávají.

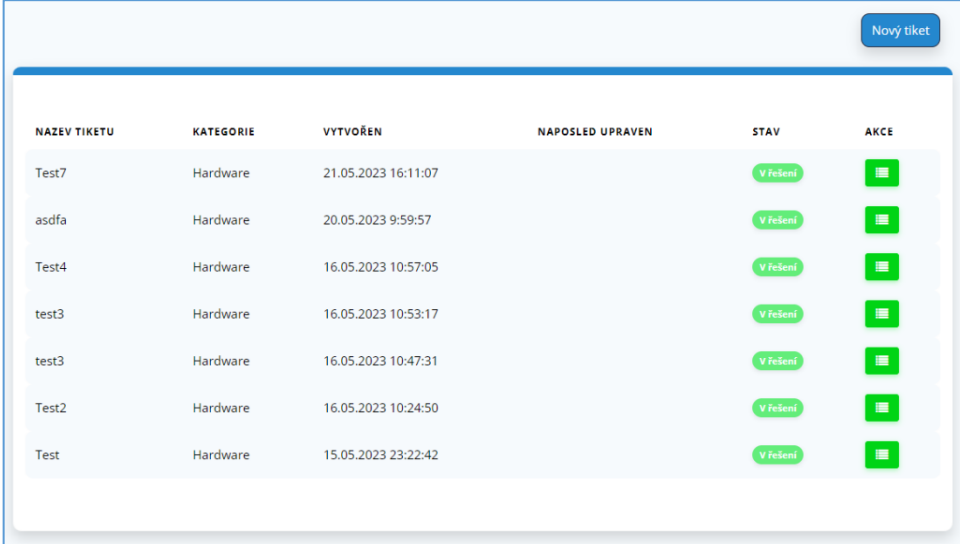


The image shows a registration form with the following fields and elements:

- Title:** Vytvoření nového účtu
- Jméno:** Input field with placeholder text "Jméno".
- E-mail:** Input field with placeholder text "jmeno@domena.cz".
- Heslo:** Input field with placeholder text "Heslo".
- Heslo znovu:** Input field with placeholder text "Heslo znovu".
- Buttons:** A blue "Registrovat" button and a "Zpět na přihlášení" link.

Obrázek 18 Obrazovka pro založení nového uživatele

Po úspěšném zaregistrování a následném přihlášení má uživatel k dispozici přehled všech jím založených tiketů (obrázek 19). V rámci každého tiketu lze přidávat komentář pomocí tlačítka ve sloupci „Akce“. Nový tiket lze vytvořit po kliknutí na tlačítko „Nový tiket“.

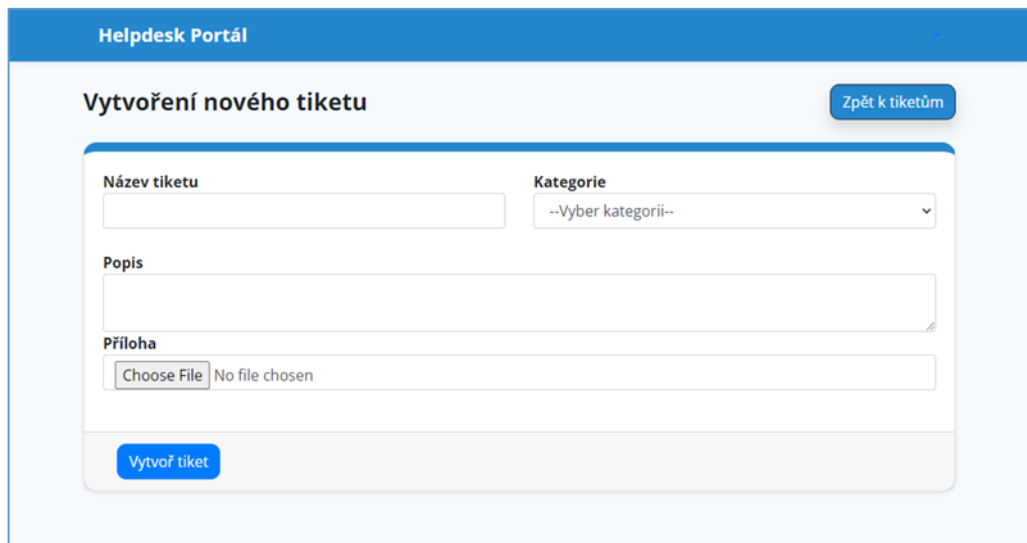


The screenshot shows a user interface for managing tickets. At the top right, there is a blue button labeled "Nový tiket". Below it is a table with the following columns: "NAZEV TIKETU", "KATEGORIE", "VYTVOŘEN", "NAPOSLED UPRAVEN", "STAV", and "AKCE". The table contains eight rows of test data, each with a "V řešení" status and an "AKCE" button.

NAZEV TIKETU	KATEGORIE	VYTVOŘEN	NAPOSLED UPRAVEN	STAV	AKCE
Test7	Hardware	21.05.2023 16:11:07		V řešení	[AKCE]
asdfa	Hardware	20.05.2023 9:59:57		V řešení	[AKCE]
Test4	Hardware	16.05.2023 10:57:05		V řešení	[AKCE]
test3	Hardware	16.05.2023 10:53:17		V řešení	[AKCE]
test3	Hardware	16.05.2023 10:47:31		V řešení	[AKCE]
Test2	Hardware	16.05.2023 10:24:50		V řešení	[AKCE]
Test	Hardware	15.05.2023 23:22:42		V řešení	[AKCE]

Obrázek 19 Přehled tiketů přihlášeného uživatele

Po kliknutí na tlačítko „Nový tiket“ se uživateli zobrazí okno pro založení nového tiketu (obrázek 20). Z podstaty zamýšlené jednoduchosti v zakládání požadavků jsou zde pouze nezbytně nutná pole: „Název tiketu“ (povinné pole), „Kategorie“ (nepovinné pole), „Popis“ (nepovinné pole) a „Příloha“ (nepovinné pole). Příloha je omezená pouze na formáty definované v kapitole 3 a maximální velikostí do 5 MB. Vytvoření je nutné potvrdit stiskem tlačítka „Vytvoř tiket“.



The image shows a web form for creating a new ticket. The form is titled "Vytvoření nového ticketu" and is part of the "Helpdesk Portál". It includes a "Zpět k ticketům" button in the top right corner. The form fields are:

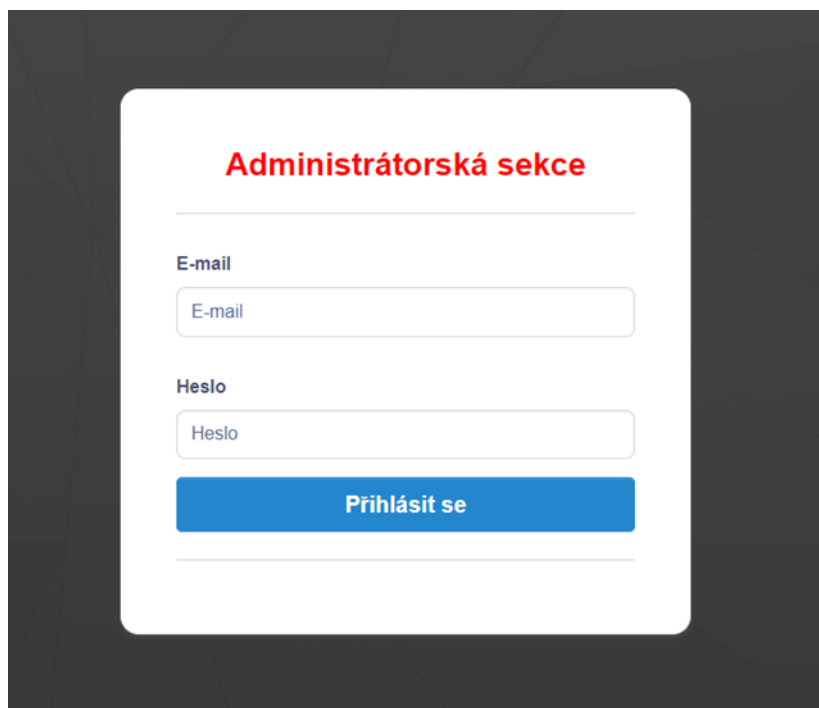
- Název ticketu**: A text input field.
- Kategorie**: A dropdown menu with the placeholder text "--Vyber kategorii--".
- Popis**: A large text area for the ticket description.
- Příloha**: A file upload section with a "Choose File" button and the text "No file chosen".

At the bottom of the form is a blue button labeled "Vytvoř ticket".

Obrázek 20 Vytvoření nového ticketu

4.3.2 Administrátorská sekce

Vstup do administrátorské sekce je umožněn pouze uživateli, který má administrátorské oprávnění (definováno hodnotou atributu v databázi). Přihlašovací obrazovka je přístupná pouze přes adresu, kterou je nutné zadat přímo do adresního řádku prohlížeče (../account/login).



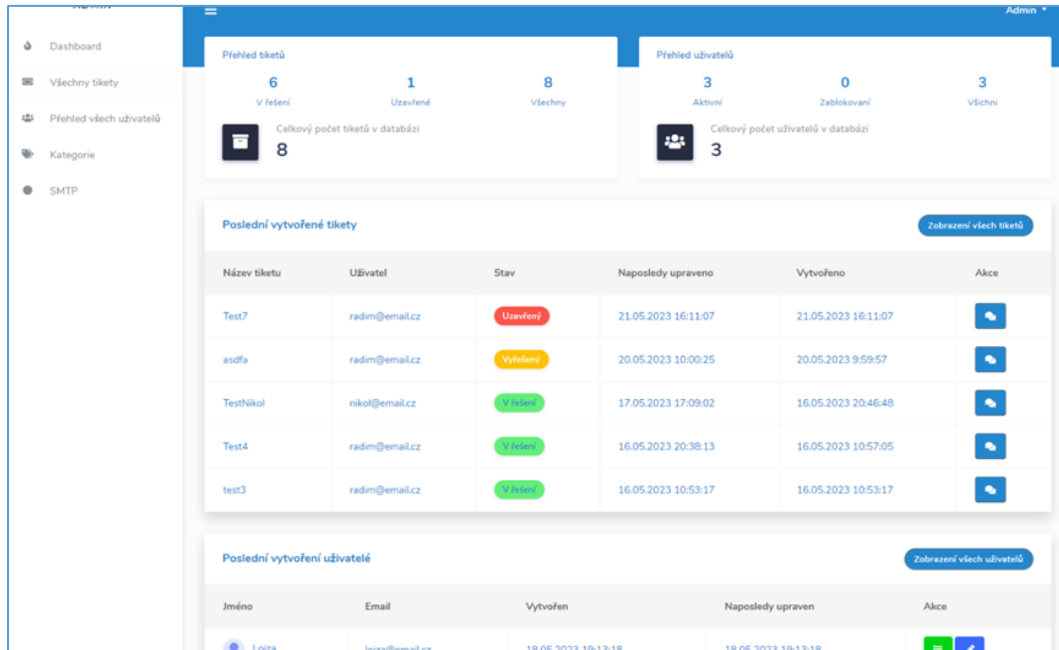
Obrázek 21 Přihlašovací obrazovka do administrátorské sekce

Po úspěšném přihlášení se administrátorovi zobrazí úvodní přehledová stránka – tzv. dashboard (obrázek 22).

Lze zde nalézt „Přehled tiketů“, kde jsou zobrazeny aktuální počty tiketů v databázi agregované do kategorií (v řešení, uzavřené a všechny tikety). Hned vedle je pak analogická sekce pro uživatele. Přehled je rozdělen do kategorií aktivní, zablokovaní a všichni uživatelé.

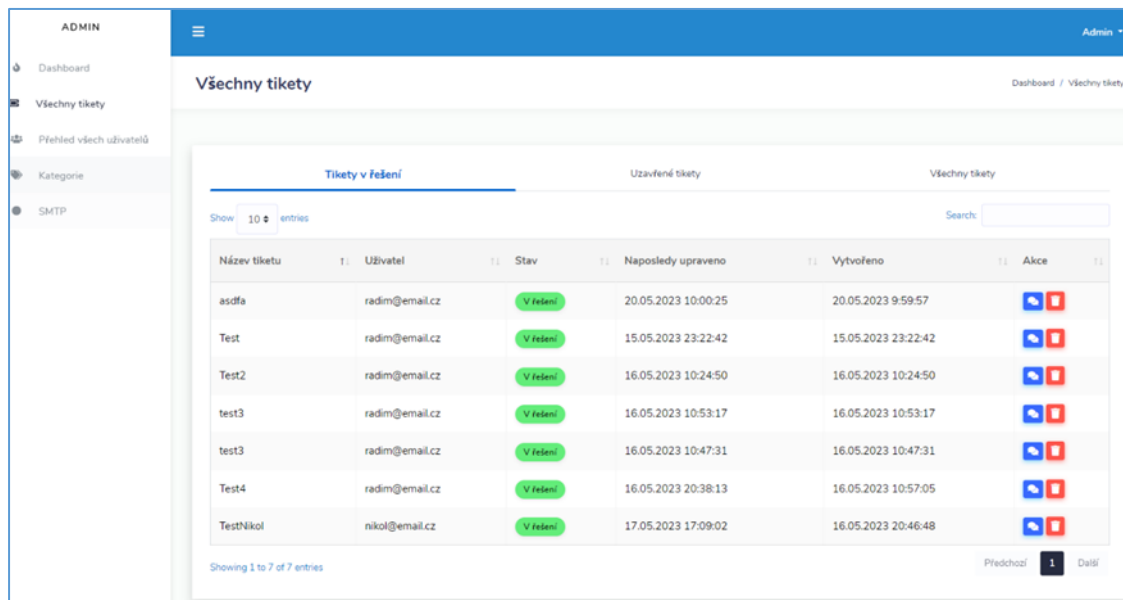
Následuje sekce „Poslední vytvořené tikety“ kde jsou tikety seřazeny dle data založení od nejnovější po nejstarší. U každého tiketu je poté tlačítko pro případné komentáře k tiketu.

Poslední částí dashboardu administrátora jsou „Poslední vytvoření uživatelé“. Zde je přehled uživatelů a jejich údajů seřazených dle data od nejnovějších.



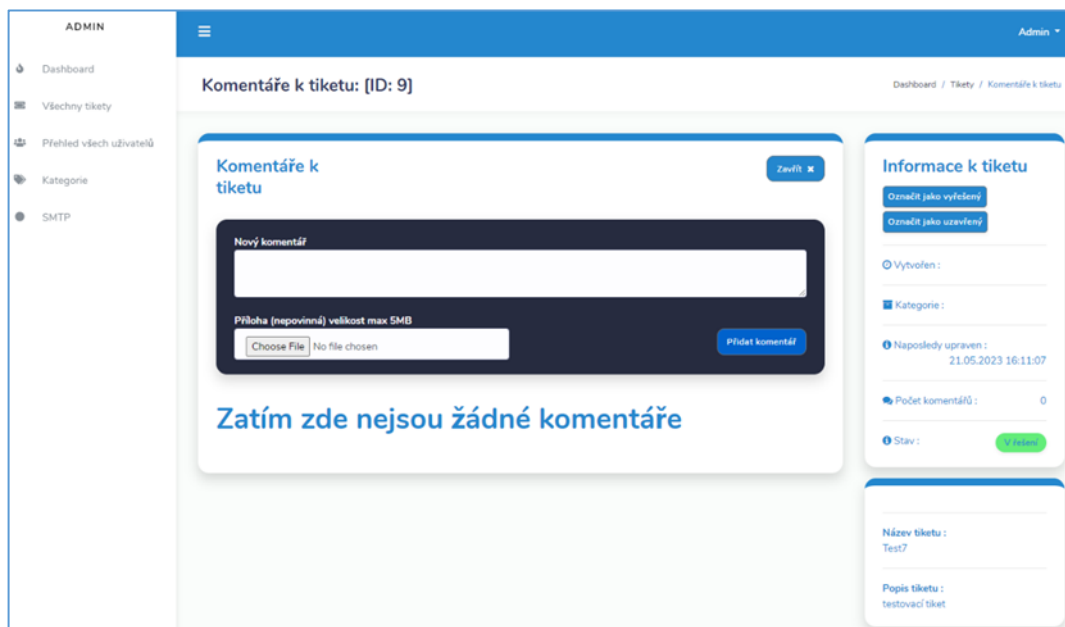
Obrázek 22 Administrátorský dashboard

Další položkou v menu administrátora jsou „Všechny tikety“. Zde může administrátor spravovat tikety založené v systému. Lze přepínat mezi třemi přehledy. Dále je pak možné ke každému tiketu přidávat komentář nebo tiket smazat.



Obrázek 23 Správa tiketů v administrátorské sekci

Komentáře k ticketům lze vkládat pomocí tlačítka „Přidat komentář“. Následně se zobrazí formulářové pole (obrázek 24), kde je možné přidat text komentáře a přílohu.



Obrázek 24 Přidání komentáře ke konkrétnímu ticketu

Následující možností, kterou lze v rámci administrátorské sekce použít, je zobrazení „Přehled všech uživatelů“ v systému (obrázek 25). Pomocí tlačítka „Přidat uživatele“ je možno přidat nového uživatele do helpdesk portálu. Prostřednictvím přepínače lze uživatele dočasně blokovat. V neposlední řadě nabízí tato sekce zobrazení všech ticketů konkrétního uživatele, editaci vlastností uživatele a případně jeho trvalé smazání z databáze.

Přehled všech uživatelů

Dashboard / Přehled všech uživatelů

Přidat uživatele

Show 10 entries Search:

Pořadové číslo	Jméno	E-mail	Vytvořen	Naposledy změněn	Stav	Akce
1	Test	test@test.cz	22-05-2023 19:19:47	01-01-0001 00:00:00	<input checked="" type="checkbox"/>	Editovat Smazat
2	Lojza	lojza@email.cz	18-05-2023 19:13:18	18-05-2023 19:13:18	<input checked="" type="checkbox"/>	Editovat Smazat
3	Nikol	nikol@email.cz	16-05-2023 20:46:11	01-01-0001 00:00:00	<input checked="" type="checkbox"/>	Editovat Smazat
4	Radim	radim@email.cz	15-05-2023 23:15:08	01-01-0001 00:00:00	<input checked="" type="checkbox"/>	Editovat Smazat

Showing 1 to 4 of 4 entries

Předchozí 1 Další

Obrázek 25 Přehled všech uživatelů v systému

Další možností, kterou administrátor má, je vytváření a správa jednotlivých kategorií požadavků. Tyto kategorie slouží k přesnější identifikaci a zařazení požadavku. Novou kategorii požadavků lze přidat pomocí tlačítka „Vytvořit kategorii“. Dále je možno nastavit viditelnost kategorie při zakládání nového požadavku pomocí přepínače. Název kategorie jde dále editovat a v případě potřeby lze kategorii smazat.

ADMIN

Dashboard

Všechny tikety

Přehled všech uživatelů

Kategorie

SMTP

Přehled všech kategorií ICT

Dashboard / Všechny kategorie

Vytvořit kategorii

Show 10 entries Search:

Číslo kategorie	Název kategorie	Stav	Naposledy upraveno	Akce
1	Mobilní služby	<input type="checkbox"/>	16-05-2023 20:41:16	Editovat Smazat
2	Software	<input checked="" type="checkbox"/>	16-05-2023 20:44:33	Editovat Smazat
3	Hardware	<input checked="" type="checkbox"/>	23-04-2023 15:41:10	Editovat Smazat

Showing 1 to 3 of 3 entries

Předchozí 1 Další

Obrázek 26 Přehled kategorií zakládaných požadavků

5 ZABEZPEČENÍ APLIKACE

V rámci helpdesk portálu, se zpracovávají citlivé informace uživatelů, jako jsou e-maily, telefonní čísla a především hesla. Je tedy nutné zajistit, aby uchování takto citlivých informací bylo řádně zabezpečeno. Dále by výsledná webová aplikace měla být zabezpečena proti častým typům útokům, jakou jsou SQL Injection, Cross-Site Scripting (XSS), Ochrana proti Cross-Site Request Forgery (CSRF). V neposlední řadě je nezbytné řešit autentizaci a autorizaci. Všechno výše vedené bylo podrobněji popsáno právě v této kapitole.

5.1 Ukládání hesel do databáze

Hesla jsou do databáze ukládána pomocí hashovacího algoritmu BCrypt, který je založen na šifře Blowfish. Tento algoritmus zvyšuje odolnost vůči slovníkovým útokům nebo útokům hrubou silou (brute-force) jelikož přidává náhodnou "sůl" pro každé heslo, tzn. pokud by vícero uživatelů mělo stejné heslo, hashe těchto hesel budou různé. [22] Jak lze vidět na obrázku níže (25) pro šifrování byl zvolen "work factor" 13, který značí, kolik iterací bude provedeno při hashování hesla. Byl zvolen na základě doporučení Aarona Toponce [23]

2^n	8	9	10	11	12	13	14	15	16	17	18	19	20
Čas(s)	0.018	0.035	0.068	0.137	0.266	0.54	1.082	2.154	4.244	8.488	16.997	34.144	69.13

Obrázek 27 Work faktory a časové náročnosti hashování. Zdroj dat [23]

```
string salt = BCrypt.Net.BCrypt.HashPassword(model.PasswordHash,
workFactor: 13);
model.PasswordHash = salt;
```

Obrázek 28 Generování hesla pomocí algoritmu BCrypt

5.2 Ochrana proti SQL Injection

Na zabezpečení aplikace proti SQL Injection útokům je využít již dříve zmíněný EF Core, který pro přístup k datům nevyužívá klasické SQL dotazy, ale tzv. LINQ dotazy, které jsou integrovány. Dotazy jsou psané v kódu a posléze transformované do SQL dotazu. Při použití LINQ dotazů lze omezit útoky typu SQL Injection.

```
ViewBag.list = await _context.Tickets.OrderByDescending(x =>
x.CreatedDate).ToListAsync();
    ViewBag.running = await _context.Tickets.Where(x => x.StatusId ==
0).OrderByDescending(x => x.CreatedDate).ToListAsync();
    ViewBag.closed = await _context.Tickets.Where(x => x.StatusId ==
1).OrderByDescending(x => x.CreatedDate).ToListAsync();
    return View();
```

Obrázek 29 Ukázka vykonání dotazu nad databází pomocí LINQ

5.3 Ochrana proti Cross-Site Scripting (XSS)

Proti zranitelnosti, kdy se útočník snaží vložit do webové stránky nebezpečný kód, který se pak provádí ve webovém prohlížeči uživatele, je tato aplikace zabezpečena validací uživatelských vstupů. Použité helpery ValidationSummary (obrázek 31) a ValidationMessageFor (obrázek 32) využívají validaci modelů. V odpovídajícím modelu pak byla vždy nadefinována pravidla pro validaci vstupů (obrázek 33).

```
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
```

Obrázek 30 Helper ValidationSummary

```
@Html.ValidationMessageFor(model => model.PasswordHash, "", new { @class = "text-
danger" })
```

Obrázek 31 Helper ValidationMessageFor

```
[Required]
[StringLength(100, ErrorMessage = "Heslo {0} musí mít minimálně {2} znaků.",
MinimumLength = 10)]
[DataType(DataType.Password)]
[Display(Name = "Password")]
[RegularExpression(@"^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[^\da-zA-Z]).{10,}$",
ErrorMessage = "Heslo musí obsahovat minimálně 10 znaků a zahrnovat alespoň jedno velké písmeno," +
" jedno malé písmeno, jednu číslici a jeden speciální znak.")]
public string? PasswordHash { get; set; }
```

Obrázek 32 Pravidla pro vytvoření hesla při registraci uživatele

5.4 Ochrana proti Cross-Site Request Forgery (CSRF)

Proti tomuto typu útoku, při kterém může útočník převzít relaci přihlášeného uživatele a následně odesílat požadavky na jiný server bez vědomí uživatele, byl v kódu implementován bezpečnostní prvek tzv. „Anti-Forger Token“. Jedná se o funkci obsaženou přímo v ASP.NET CORE.

Níže (Obrázek 31) je implementace v případě vytvoření nového tiketu. Prvně bylo nutno vložit do HTML formuláře pro vytvoření tiketu Anti-Forger token.

```
<form asp-action="Create" asp-controller="Tickets" enctype="multipart/form-data">
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true, "", new { @class = "text-danger" })
```

Obrázek 33 AntiForgeryToken v HTML formuláři vytvoření tiketu

Následně v controlleru tiketu bylo ještě potřeba přidat atribut „ValidateAntiForgeryToken“

```
[HttpPost]
[ValidateAntiForgeryToken]
public async Task<ActionResult> Create(Tickets tickets)
```

Obrázek 34 Atribut „ValidateAntiForgeryToken“ v tiket controlleru

Díky výše zmíněným implementacím do kódu je uživatel vždy ověřen a každý jeho požadavek obsahuje platný autentizační token. Chrání tedy před podvržení požadavků z jiného webu.

5.5 Autentizace

Autentizace je řešena za pomoci knihovny Microsoft Identity. Uživatel pro přístup do helpdesk portálu musí být zaregistrován, tzn. má v databázi vytvořené uživatelské jméno a heslo, a poté řádně přihlášen. Totéž platí pro administrátora, pouze s tím rozdílem, že administrátorské přihlašovací údaje jsou vytvořeny v databázi ručně. Heslo pro administrátora bylo generováno za pomoci online nástroje Bcrypt Password Generator [24] a poté vygenerovaný hash ručně vložen přímo do databáze.

5.6 Autorizace

Na proces autentizace navazuje proces autorizace. Opět je zde použita knihovna Microsoft Identity. Přístup do sekcí helpdesk portálu dostupného pouze pro přihlášené uživatele (rozděleno na Admin a User Areas) zabezpečuje atribut „Authorize“, který je vložen vždy v daných Controllerech.

Níže (Obrázek 33) je zobrazeno použití atributu „Authorize“ pro uživatele

```
namespace HelpDesk.Areas.User.Controllers
{
    [Area("User")]
    [Authorize]
    public class MyTicketsController : Controller
    {
```

Obrázek 35 Atribut „Authorize“ pro User sekci

Administrátorské prostředí má navíc dodatečný atribut „PermissionCheck“, který ověřuje, zdali má přihlášený uživatel administrátorské oprávnění. Administrátor má navíc oddělené přihlašování, které je dostupné pouze po zadání přesného odkazu do adresního řádku prohlížeče.

Níže je ukázáno použití atributů „Authorize“ a „PermissionCheck“ pro zajištění přístupu pouze pro uživatele, který má administrátorské oprávnění.

```
namespace HelpDesk.Areas.Admin.Controllers
{
    [Area("Admin")]
    [Authorize]
    [PermissionCheck("Admin")]
    public class AllTicketsController : Controller
    {
```

Obrázek 36 Atributy „Authorize“ a „PermissionCheck“ pro Admin sekci

5.7 SSL

Výsledná aplikace může být využívána jak v rámci intranetu, tak v rámci internetu. Zejména z důvodu přístupnosti aplikace v rámci internetu je nutné zajistit, aby data mezi serverem a prohlížečem byla šifrována. Pokud by se tedy aplikace v budoucnu využívala, je nutností zajistit SSL certifikát, v dnešní době by to mělo být již standardem.

ZÁVĚR

Cílem této bakalářské práce bylo vytvořit funkční webovou aplikaci pro zákaznickou podporu, která bude do maximální možné míry jednoduchá, uživatelsky přívětivá a efektivní v zakládání a správě požadavků, bude obsahovat zdroj informací a v neposlední řadě real-time chat.

V teoretické části byla zprvu provedena rešerše existujících řešení, dále byly popsány technologie frameworku .NET spolu s dalšími souvisejícími technologiemi. U každé z popsaných technologií bylo vysvětleno, k jakému účelu ve výsledné aplikaci slouží. Real-time chat byl u serverové části vytvořen pomocí ASP.NET Web API, pro front-end byl použit framework Angular. Zdroj informací a helpdesk portál byl vyvíjen pomocí ASP.NET Core MVC.

Praktická část poté popsala nezbytnou a důležitou součást, tedy analýzu výsledného systému. Na jejímž základě byla poté realizována výsledná aplikace. Ta byla dále obsáhle – textově i graficky popsána. Praktickou část zakončilo zabezpečení aplikace, formou přehledu možných zranitelností a útoků a jak je proti nim aplikace zabezpečena.

I když byla výsledná webová aplikace pro zákaznickou podporu realizována pomocí vícero technologií, tvoří v konečném důsledku jeden funkční celek – aplikaci pro zákaznickou podporu, která umožňuje rychlé, efektivní a uživatelsky přívětivé řešení uživatelských problémů či dotazů.

SEZNAM POUŽITÉ LITERATURY

- [1] Spiceworks Cloud Help Desk. Spiceworks [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://www.spiceworks.com/free-cloud-help-desk-software/>
- [2] Help Desk. Freshworks [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://www.freshworks.com/freshdesk/help-desk/>
- [3] Helpdesk Ticketing System. *Jitbit* [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://www.jitbit.com/helpdesk/>
- [4] Features. OsTicket [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://osticket.com/features/>
- [5] GIRETTI, Anthony. Beginning gRPC with ASP.NET Core 6: Build Applications using ASP.NET Core Razor Pages, Angular, and Best Practises in .NET 6. 1. La Salle, QC, Canada: Apress, 2022. ISBN 978-1-4842-8007-2.
- [6] A tour of the C# language. *Microsoft* [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>
- [7] Visual Studio 2022 Preview. *Microsoft* [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://visualstudio.microsoft.com/vs/preview/>
- [8] What's new in Visual Studio 2022 [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://learn.microsoft.com/en-us/visualstudio/ide/whats-new-visual-studio-2022?view=vs-2022>
- [9] Entity Framework Core. *Microsoft* [online]. 2021 [cit. 2023-05-21]. Dostupné z: <https://learn.microsoft.com/en-us/ef/core/>
- [10] Overview of ASP.NET Core SignalR. *Microsoft* [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/signalr/introduction?view=aspnetcore-6.0>
- [11] What is Angular?: Architecture, Features, and Advantages. *Simplilearn* [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://www.simplilearn.com/tutorials/angular-tutorial/what-is-angular>
- [12] <https://bootstrapmade.com/onepage-multipurpose-bootstrap-template/>
- [13] JAKOBUS, Benjamin. Mastering Bootstrap 4: Master the latest version of Bootstrap 4 to build highly customized responsive web apps. Packt Publishing Ltd, 2018. page. 6

- [14] CASTRO, Elizabeth a Bruce HYSLOP. HTML and CSS: visual quickstart guide. Eight edition. [Berkeley, CA]: Peachpit Press, [2014]. Visual quickstart guide. ISBN 978-0-321-92883-2. page 16-17
- [15] WHAT IS CSS? W3C [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://www.w3.org/Style/CSS/Overview.en.html>
- [16] JavaScript – Overview. Tutorialspoint [online]. 2023 [cit. 2023-05-21]. Dostupné z: https://www.tutorialspoint.com/javascript/javascript_overview.htm
- [17] PETKOVIĆ, Dušan. SQL Server 2019 A Beginner's Guide. 7. New York: McGraw-Hill Education, 2020. ISBN 978-1-26-045887-9.
- [18] UNHELKAR, Bhuvan. Software Engineering with UML. 1. Boca Raton London New York: CRC Press, 2017. ISBN 978-1-138-29743-2.
- [19] Enterprise Architect. SPARX Systems [online]. 2023 [cit. 2023-05-21]. Dostupné z: <https://sparxsystems.com/>
- [20] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. 2., aktualiz. a dopl. vyd. Brno: Computer Press, 2007. ISBN 978-80-251-1503-9.
- [21] Code First vs Database First vs Model First – EntityFramework Approaches Explained. .NET Core Tutorials [online]. 2022 [cit. 2023-05-21]. Dostupné z: <https://dotnetcoretutorials.com/code-first-vs-database-first-vs-model-first-entityframework-approaches-explained/>
- [22] ARIAS, Dan. Hashing in Action: Understanding bcrypt. Secure access for everyone. But not just anyone. [online]. 2021 [cit. 2023-05-21]. Dostupné z: <https://auth0.com/blog/ hashing-in-action-understanding-bcrypt/>
- [23] TOPONCE, Aaron. Let's Talk Password Hashing. Aaron Toponce : Linux. GNU. Freedom. [online]. Dostupné z: <https://pthree.org/2016/06/28/lets-talk-password-hashing/>
- [24] Bcrypt Encrypt - Bcrypt Hash Generator - Online - Browserling Web Developer Tools. Browserling – Online cross-browser testing [online]. 2023. [cit. 21.05.2023]. Dostupné z: <https://www.browserling.com/tools/bcrypt>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

FAQ	Frequently Asked Questions
ASP	Active Server Pages
MVC	Model-View-Controller
API	Application Programming Interface
SQL	Structured Query Language
MSSQL	Microsoft Structured Query Language
E-R	Entity Relationship
ICT	Information and Communication Technologies
UI	User Interface
UX	User Experience
CRM	Customer Relationship Management
2FA	Two-Factor Authentication
PHP	Hypertext Preprocessor
MySQL	My Structured Query Language
IIS	Internet Information Services
SoC	Separation of Concerns
SEO	Search Engine Optimization
HTML	Hypertext Markup Language
TDD	Test-Driven Development
IDE	Integrated Development Environment
ORM	Object-Relational Mapping
GPS	Global Positioning System
RPC	Remote Procedure Call

XML	Extensible Markup Language
JSON	JavaScript Object Notation
CSS	Cascading Style Sheets
UML	Unified Modeling Language
SHA	Secure Hash Algorithm
XSS	Cross-Site Scripting
CSRF	Cross-Site Request Forgery
LINQ	Language Integrated Query
REST	Representational State Transfer
EF	Entity Framework

SEZNAM OBRÁZKŮ

Obrázek 1 Přehled sekce „Tickets“ v aplikaci SpiceWorks (zdroj spiceworks.com).....	12
Obrázek 2 Přehled sekce „All tickets“ v aplikaci Freshdesk (zdroj freshdesk.com).....	14
Obrázek 3 Přehled sekce „Tikety“ v aplikaci JitBit (zdroj jitbit.com)	17
Obrázek 4 Přehled sekce „Tickets“ v aplikaci osTicket (zdroj osticket.com).....	19
Obrázek 5 Architektura ASP.NET Core MVC. Zdroj [5]	24
Obrázek 6 Architektura ASP.NET Core Web API. Zdroj [5]	27
Obrázek 7 Funkční požadavky organizovány do balíčků.....	35
Obrázek 8 Nefunkční požadavky.....	39
Obrázek 9 Přehled aktérů v systému.....	40
Obrázek 10 Use Case model helpdesk portálu	42
Obrázek 11 Diagram tříd	48
Obrázek 12 Databázové schéma helpdesk portálu	49
Obrázek 13 Úvodní stránka celé výsledné aplikace	54
Obrázek 14 Přihlašovací formulář pro vstup do real-time chatu	55
Obrázek 15 Obrazovka real-time chatovací aplikace	56
Obrázek 16 Zaslání soukromé zprávy konkrétnímu uživateli chatu.....	56
Obrázek 17 Přihlašovací obrazovka pro vstup do helpdesk portálu.....	58
Obrázek 18 Obrazovka pro založení nového uživatele.....	59
Obrázek 19 Přehled tiketů přihlášeného uživatele.....	60
Obrázek 20 Vytvoření nového tiketu.....	61
Obrázek 21 Přihlašovací obrazovka do administrátorské sekce	62
Obrázek 22 Administrátorský dashboard	63
Obrázek 23 Správa tiketů v administrátorské sekci.....	63
Obrázek 24 Přidání komentáře ke konkrétnímu tiketu	64
Obrázek 25 Přehled všech uživatelů v systému.....	65
Obrázek 26 Přehled kategorií zakládaných požadavků	65
Obrázek 27 Work faktory a časové náročnosti hashování. Zdroj dat [23]	66
Obrázek 28 Generování hesla pomocí algoritmu BCrypt.....	66
Obrázek 29 Ukázka vykonání dotazu nad databází pomocí LINQ	67
Obrázek 30 Helper ValidationSummary.....	67
Obrázek 31 Helper ValidationMessageFor.....	67

Obrázek 32 Pravidla pro vytvoření hesla při registraci uživatele.....	68
Obrázek 33 AntiForgeryToken v HTML formuláři vytvoření tiketu.....	68
Obrázek 34 Atribut „ValidateAntiForgeryToken“ v tiket controlleru.....	68
Obrázek 35 Atribut “Authorize” pro User sekci.....	69
Obrázek 36 Atributy „Authorize“ a „PermissionCheck“ pro Admin sekci.....	69

SEZNAM TABULEK

Tabulka 1 Přehledová tabulka výsledků aplikací dle hodnotících kritérií.....	21
Tabulka 2 Primární scénář – Přihlášení uživatele do helpdesk portálu	43
Tabulka 3 Alternativní scénář – Uživatel zatím nemá přihlašovací údaje	44
Tabulka 4 Alternativní scénář – Přihlášení se nezdařilo.....	45
Tabulka 5 Primární scénář – Vytvoření ticketu v helpdesk portálu	45
Tabulka 6 Alternativní scénář: Přihlášení se nezdařilo	47

SEZNAM PŘÍLOH

P1 Flash disk obsahující bakalářskou práci a zdrojové kódy výsledné aplikace

