

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Návrh a implementace helpdesku pro vybrané zdravotnické zařízení

Vladyslav Babyč

Vedoucí: Ing. Pavel Náplava, Ph. D.
Studijní program: Softwarové inženýrství a technologie
Specializace: Enterprise systémy
Květen 2024

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: **Babyč** Jméno: **Vladyslav** Osobní číslo: **507325**
Fakulta/ústav: **Fakulta elektrotechnická**
Zadávající katedra/ústav: **Katedra počítačů**
Studijní program: **Softwarové inženýrství a technologie**
Specializace: **Enterprise systémy**

II. ÚDAJE K BAKALÁŘSKÉ PRÁCI

Název bakalářské práce:

Návrh a implementace helpdesku pro vybrané zdravotnické zařízení

Název bakalářské práce anglicky:

Design and implementation of a helpdesk system for selected medical facility

Pokyny pro vypracování:

Pro vybrané zdravotnické zařízení analyzujte IT potřeby z pohledu helpdesku, a navrhnete aplikaci, která je naplní. Postupujte následovně:

- 1) Definujte pojem zdravotnické zařízení a specifika IT podpory ve zdravotnictví.
- 2) Ve vybraném zařízení proveďte analýzu procesů, souvisejících s helpdeskem.
- 3) Proveďte průzkum existujících nástrojů a vyhodnoťte jejich použitelnost.
- 4) Navrhnete nový systém, vycházející ze specifík prostředí a výhod existujících řešení.
- 5) Navržený systém implementujte ve verzi MVP s následujícími funkcionalitami, a uživatelsky jej otestujte:
 - přihlášení a přiřazení oprávnění role uživatele,
 - zadání požadavku a sledování systémových atributů (čas a datum zadání, autor, ...),
 - práce s upřesňujícími atributy (priorita, kategorie, ...), včetně editace,
 - přiřadit k požadavku řešitele (příp. skupinu řešitelů),
 - přeřazení požadavku jinému řešiteli,
 - různé pohledy na seznam požadavků dle vybraných filtrů.
- 6) Navrhnete, jakým způsobem bude systém nasazen a rozvíjen.

Seznam doporučené literatury:

- [1] Suzanne Robertson, James Robertson: Mastering the Requirements Process: Getting Requirements Right 3rd Edition, ISBN-10: 0321815742
- [2] Roger S. Pressmann Bruce Maxim: Software Engineering: A Practitioner's Approach, ISBN-10: 9780078022128
- [3] Noel Bruton: How to Manage the IT Helpdesk: A guide for user support and call centre managers, ISBN-10: 0750638117
- [4] Robert C. Martin: Clean Code: A Handbook of Agile Software Craftsmanship, ISBN-10: 9780132350884

Jméno a pracoviště vedoucí(ho) bakalářské práce:

Ing. Pavel Náplava, Ph.D. Centrum znalostního managementu FEL

Jméno a pracoviště druhé(ho) vedoucí(ho) nebo konzultanta(ky) bakalářské práce:

Datum zadání bakalářské práce: **07.02.2024**

Termín odevzdání bakalářské práce: **24.05.2024**

Platnost zadání bakalářské práce: **21.09.2025**

Ing. Pavel Náplava, Ph.D.
podpis vedoucí(ho) práce

podpis vedoucí(ho) ústavu/katedry

prof. Mgr. Petr Páta, Ph.D.
podpis děkana(ky)

III. PŘEVZETÍ ZADÁNÍ

Student bere na vědomí, že je povinen vypracovat bakalářskou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je třeba uvést v bakalářské práci.

Datum převzetí zadání

Podpis studenta

Poděkování

V první řadě bych chtěl poděkovat vedoucímu této bakalářské práce Ing. Pavlu Náplavovi, Ph. D. za jeho odbornou pomoc, profesionální rady a osobní přístup při vedení této bakalářské práce. Dále bych chtěl poděkovat své rodině za podporu během studia a psání této bakalářské práce. Nakonec bych také chtěl poděkovat všem kolegyním a kolegům, kteří se podíleli na definici požadavků a testování aplikace.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 21. května 2024

Abstrakt

Bakalářská práce se věnuje oblasti uživatelské podpory ve vybraném zdravotnickém zařízení.

V první části, práce obsahuje podrobnou analýzu stávajícího stavu s identifikací existujících problémů a navrhuje jejich řešení. Dále je provedena analýza existujících helpdesk platform, která je zakončena jejich kvalitativním srovnáním. Po vyhodnocení tohoto srovnání a také na základě rozhovoru s managementem organizace bylo rozhodnuto, že je preferováno, aby řešení bylo vytvořeno cestou vlastního vývoje na základě uživatelských požadavků.

V druhé části se práce věnuje definici funkčních a nefunkčních požadavků, definici uživatelských rolí a oprávnění, definici případů užití a také výběru technologií pro implementaci aplikace. Vycházejíc z těchto definic byl vytvořen návrh systému, součástí kterého byl prototyp uživatelského rozhraní, jež byl otestován pěti vybranými zaměstnanci a připomínky z tohoto testování byly zohledněny ve finální implementaci. Po provedení těchto kroků byla aplikace vyvinuta a nasazena v prostředí organizace, načež byla uživatelsky otestována. V závěru druhé části je nasazení aplikace vyhodnoceno a je zde popsán její další rozvoj.

Klíčová slova: Helpdesk, zdravotnické zařízení, Spring Boot, Java, Svelte, Docker

Vedoucí: Ing. Pavel Náplava, Ph. D.

Abstract

The bachelor thesis focuses on the area of user support in a selected medical facility.

In the first part, the thesis contains a detailed analysis of the existing situation with identification of existing problems and also proposes their solutions. Then, an analysis of existing helpdesk platforms is made, which is concluded with their qualitative comparison. After evaluating this comparison and also based on an interview with the management of the organization, it was decided that it is preferred that the solution will be created through in house development based on user requirements.

In the second part of the thesis, the definition of functional and non-functional requirements, definition of user roles and permissions, definition of use cases and also selection of technologies for application implementation are made. Based on these definitions, a system design was created, which included a prototype user interface that was tested by five selected employees and the comments from this testing were taken into account in the final implementation. Following these steps, the application was developed and deployed in the organisation's environment, after which it was user tested. At the end of the second part, the deployment of the application is evaluated and its further development is described.

Keywords: Helpdesk, medical facility, Spring Boot, Java, Svelte, Docker

Title translation: Design and implementation of a helpdesk system for selected medical facility

Obsah

1 Úvod	1	6 Návrh systému	43			
1.1 Motivace	1	6.1 Prototyp uživatelského rozhraní	43			
1.2 Cíl	2	6.1.1 Uživatelské testování prototypu	44			
1.3 Struktura práce	2	6.1.2 Výsledky testování	45			
2 Specifikace prostředí zdravotnického zařízení	3	6.1.3 Vyhodnocení testování prototypu	47			
Část I						
Analýza stávajícího stavu řešení požadavků a návrh řešení						
3 Analýza současného stavu řešení požadavků	7	6.2 Entity a class diagram	47			
3.1 Aktuální proces řešení uživatelských požadavků	7	6.3 Autentizace a autorizace	48			
3.1.1 Identifikované problémy v procesu řešení uživatelských požadavků	8	6.4 Závěr návrhu systému	48			
3.1.2 Způsoby řešení jednotlivých problémů	8	7 Implementace	49			
3.1.3 Shrnutí aktuálního stavu	11	7.1 Frontend aplikace	49			
3.2 Existující platformy pro implementaci helpdesku	12	7.2 Backend aplikace	52			
3.2.1 Hodnotící kritéria nabízených řešení	13	7.3 Nasazení aplikace	54			
3.2.2 Jednotlivá řešení	14	7.3.1 Spuštění databáze	56			
3.2.3 Závěr hodnocení	24	7.3.2 Spuštění backendu	57			
Část II				7.3.3 Spuštění frontendu	57	
Tvorba vlastního helpdesk řešení				7.4 Problémy při implementaci	57	
Úvod druhé části	29	7.5 Závěr implementace systému	58	8 Uživatelské testování aplikace	59	
4 Požadované vlastnosti systému	31	8.1 Výsledky testování	59	8.2 Závěr uživatelského testování	62	
4.1 FR - Funkční požadavky	31	9 Vyhodnocení nasazení aplikace	63	9.1 Další vývoj aplikace	63	
4.2 NFR - Nefunkční požadavky	32	9.1.1 Škálovatelnost	63	10 Závěr	65	
4.3 Uživatelské role a oprávnění	32	Literatura	67	Přílohy		
4.4 UC - Případy užití	33					
4.4.1 Domovská stránka helpdesku	34					
4.4.2 Práce s požadavkem	35					
4.4.3 Správa uživatelů	37					
5 Výběr technologií	39					
5.1 Backend aplikace	39					
5.1.1 Ukládání dat	39					
5.1.2 Zpracování uživatelských akcí a logika aplikace	40					
5.2 Frontend aplikace	40					
				A Obsah příložených souborů		73
				A.1 Struktura příloh		73

Obrázky

3.1 Seznam dostupných jazyků softwaru Freshdesk	15	7.8 Diagram nasazení	56
3.2 Cenová nabídka z oficiálních stránek Freshdesk	15		
3.3 Uživatelské rozhraní softwaru Freshdesk	16		
3.4 Seznam dostupných jazyků softwaru Zendesk	17		
3.5 Cenová nabídka z oficiálních stránek Zendesk	18		
3.6 Uživatelské rozhraní softwaru Zendesk	19		
3.7 Seznam dostupných jazyků softwaru Zoho Desk	20		
3.8 Cenová nabídka z oficiálních stránek Zoho Desk	20		
3.9 Uživatelské rozhraní softwaru Zoho Desk	21		
3.10 Seznam dostupných jazyků softwaru Intercom	22		
3.11 Cenová nabídka z oficiálních stránek Intercom	22		
3.12 Uživatelské rozhraní softwaru Intercom	23		
4.1 Diagram případů užití	33		
5.1 Prvek Akordeon	41		
6.1 Diagram propojení jednotlivých obrazovek	43		
6.2 Domovská obrazovka navrženého uživatelského rozhraní	44		
6.3 Diagram znázorňující závislost počtu nalezených problémů na počtu testovacích uživatelů[21]	45		
6.4 Navržený class diagram systému	47		
7.1 Obrazovka přihlášení	50		
7.2 Domovská obrazovka s požadavky	50		
7.3 Obrazovka s detailem požadavku	51		
7.4 Obrazovka pro založení požadavku	51		
7.5 Obrazovka s přehledy pro vedení organizace	52		
7.6 Obrazovka s přehledem uživatelů	52		
7.7 Struktura tříd aplikace	53		

Tabulky

3.1 Bodové hodnocení jednotlivých kandidátů	24
---	----

Kapitola 1

Úvod

V současné chvíli je situace s rozvojem informačních technologií (dále jen IT) ve zdravotnických zařízeních velmi různorodá. Napříč celou Českou republikou můžeme sledovat různé trendy ve vybraných nemocnicích. Například nemocnice v Jihlavě patří ke špičce v oboru, když na ni nahlédneme skrze prizma využitých technologií v oblasti informatiky a kyberbezpečnosti. Na druhou stranu tu jsou také nemocnice jako je nemocnice v Benešově, která se v roce 2019 stala terčem kybernetického útoku a musela na 20 dní přerušit svůj provoz. Je důležité zdůraznit, že tímto nikterak nevyvozují závěry ohledně úrovně IT v jednotlivých nemocnicích, ale pouze konstatují fakta.

1.1 Motivace

IT je v každém zdravotnickém zařízení velmi široký pojem, pod který se dá schovat cokoliv co je spojeno s programy, počítači, sítí nebo dalšími technologiemi.

Ze znalosti těchto aspektů a jakožto zaměstnanec jednoho ze zdravotnických zařízení jsem se rozhodl podívat se na toto zdravotnické zařízení z pohledu vyspělosti v oblasti IT. To s sebou přineslo také nutnost prozkoumat jiné aspekty, které s implementací zlepšení pomocí softwaru souvisí, konkrétně je řeč o byznys procesech a řízení oddělení IT v této organizaci. Z mého působení ve zdravotnickém zařízení jsem navením, že existuje spousta oblastí ke zlepšení, kde bych mohl aplikovat své znalosti z oboru softwarového inženýrství. Přišel jsem na to, že v organizaci chybí jakýkoliv nástroj pro řešení požadavků od zaměstnanců vůči nejen oddělení IT, ale i vůči jiným oddělením. Vycházejíc z těchto skutečností jsem usoudil, že by pro organizaci mohlo být přínosné, pokud by byl implementován helpdesk, jež by problematiku uživatelské podpory vyřešil. Helpdesk by byl přínosný nejen pro zaměstnance organizace, ale i pro management, jelikož by lidé na vedoucích pozicích měli větší přehled o tom, co se v organizaci děje a s jakými problémy se zaměstnanci potýkají. Dle mého názoru má oblast uživatelské podpory signifikantní dopad na bezproblémový chod zdravotnického zařízení, a právě proto jsem se zaměřil na tuto oblast.

1.2 Cíl

V rámci této práce si kladu za cíl zmapovat aktuální stav řešení uživatelských požadavků v organizaci, navrhnout řešení existujících problémů ve spojitosti s oblastí uživatelské podpory. Posoudit a ohodnotit již existující platformy, které řešení helpdesku nabízí a v případě absence vhodného kandidáta navrhnout a implementovat své vlastní softwarové řešení.

1.3 Struktura práce

Na začátku se práce věnuje prostředí zdravotnického zařízení, konkrétně jeho popisu a specifikům.

Dále kapitola č. 3 obsahuje analýzu současného stavu řešení požadavků, identifikované problémy a způsoby jejich řešení. Součástí této analýzy je také rozhovor s náměstkem pro technicko-ekonomickou část organizace. Další část této kapitoly se zaměřuje na analýzu existujících helpdesk platforem a je ukončena jejich kvalitativním srovnáním.

Kapitola č. 4 popisuje požadované vlastnosti systému a jsou v ní definovány funkční a nefunkční požadavky, a také požadované případy užití společně s definicí uživatelských rolí a oprávnění. Na základě těchto definovaných vlastností jsou v kapitole č. 5 vybrány technologie pro jednotlivé části aplikace. Kapitola č. 6 pak obsahuje prototyp uživatelského rozhraní a jeho uživatelské testování. Po provedeném testování jsou popsány entity, které systém obsahuje a je definován class diagram aplikace. Další část této kapitoly je věnována autentizaci a autorizaci. Kapitola č. 7 je věnována implementaci a nasazení vlastního softwarového řešení na základě provedené analýzy a definovaných požadavků. V kapitole č. 8 je popsáno uživatelské testování aplikace. Následně kapitola č. 9 vyhodnocuje nasazení aplikace a popisuje její další vývoj.

Kapitola 2

Specifikace prostředí zdravotnického zařízení

Pro lepší uchopení problematiky je nutné nejprve definovat několik pojmů, aby bylo pro čtenáře pochopitelné, proč na vybrané věci budeme nahlížet z konkrétního úhlu pohledu.

Zdravotnické zařízení - Dle Národního informačního portálu pojmem zdravotnické zařízení se rozumí prostory určené k poskytování zdravotních služeb (např. ambulance soukromého lékaře, nemocnice, lékárny atd.). [1] V mém konkrétním případě se jedná o zdravotnické zařízení, které disponuje šestnácti odděleními a lékárnou.

Specifika zdravotnictví z pohledu IT podpory - Ustáleným zvykem ve zdravotnictví je skutečnost, že se veškeré věci mají řešit neodkladně, čili prostými slovy hned. Tato zvyklost se však bohužel přenáší i na problematiku řešení provozních IT nebo i jiných požadavků, které zdravotníci často považují za neodkladné, i přes to, že tomu tak ve skutečnosti není.

Chování uživatelů - Pro definici tohoto pojmu je nutné si nejdříve zadefinovat slovo uživatel. V tomto konkrétním případě považujeme za uživatele takového člověka, který k vykonávání své každodenní práce používá výpočetní techniku, v našem případě počítač a nějaký pomocný software, v našem případě programy, které tomuto člověku pomáhají vykonávat své pracovní povinnosti. Z výše uvedené definice specifík zdravotnictví z pohledu IT podpory vyplývá i chování uživatelů. Často se stává, že uživatelé naléhají na vyřešení svého požadavku a přidružují mu tu nejvyšší možnou prioritu, i když ve skutečnosti by se tento konkrétní požadavek mohl vyřešit později. Odůvodněním této situace může být skutečnost, že v aktuální chvíli neexistuje, žádný systém pro zadávání těchto požadavků a také žádná evidence těchto zadaných a vyřešených požadavků.

Helpdesk - Dle webu společnosti Alvao je *"IT Helpdesk jedno místo, které odbavuje všechny požadavky na IT. Je to systém, který umožňuje každému ze zaměstnanců firmy (v případě interního helpdesku) nebo každému ze zákazníků (externí helpdesk) nahlásit incidenty, žádat o změny či vznést dotaz. O helpdesk se stará tým, který požadavky obsluhuje a komunikuje s koncovými uživateli."* [2]



Část I

Analýza stávajícího stavu řešení požadavků a návrh řešení

Kapitola 3

Analýza současného stavu řešení požadavků

Abychom měli větší představu o vybraném zdravotnickém zařízení je nutné ho popsat podrobněji. Zvolené zdravotnické zařízení disponuje 16 odděleními, včetně radiologického oddělení, oftalmologie a oddělení neurologie. V současné chvíli pracuje v tomto zařízení každý pracovní den kolem 60 lékařů a lékařek společně s dalším zdravotnickým personálem. V rámci každodenního provozu zdravotnického zařízení jsou využívány tři hlavní programy, jejichž chod musí být téměř bezvýpadkový. V případě, že k výpadku softwaru nebo hardwaru dojde, potřebuje zdravotnický personál rychlou odezvu a efektivní způsob řešení problémů.

Nejprve se podíváme, jak aktuálně vypadá proces řešení uživatelských požadavků, jež jsou směřovány na IT oddělení. Důkladně zanalyzujeme proces jejich samotného zadávání a řešení a pokusíme se identifikovat existující problémy.

3.1 Aktuální proces řešení uživatelských požadavků

V současné chvíli neexistuje žádný systém, který by zastřešoval uvedenou problematiku. Zdůrazním slovo systém z předešlé věty, jelikož nemyslím nějaký program, či software, nýbrž právě systém, který by dával této problematice řád. Veškeré požadavky směřující na IT oddělení jsou nyní zadávány prostřednictvím tří hlavních způsobů:

1. Zavoláním po telefonu jednomu ze zaměstnanců IT oddělení.
2. Náhodilým setkáním jednoho ze zaměstnanců IT v prostorách zdravotnického zařízení.
3. Napsáním e-mailu jednomu ze zaměstnanců IT oddělení.

Navíc jak již bylo zmíněno ve druhé kapitole, veškeré požadavky mívají tu nejvyšší možnou urgentnost a dle zadavatelů je nutné je řešit ihned. Pokud vybraný požadavek byl zadán prostřednictvím emailu je možné ho alespoň nějakým způsobem zaznamenat v elektronické podobě. V jiných případech je vyřešení požadavku odsouzeno pouze na paměť konkrétního ze zaměstnanců IT oddělení.

3.1.1 Identifikované problémy v procesu řešení uživatelských požadavků

1. **Absence jednoho místa pro evidenci požadavků** - Pokud by se jednalo pouze o jednotky požadavků denně, mohli bychom zavedení systému opomenout a řešit veškeré požadavky způsobem, kterým se řeší v aktuální chvíli. Aktuální stav je však takový, že požadavků jen na IT oddělení jsou denně desítky. Dle mého názoru spoléhat se na paměť jednotlivců v IT oddělení není dlouhodobě udržitelný stav. Ostatně se to projevuje již teď, kdy během dne chodí do kanceláře IT oddělení několik žadatelů s tím, že především o něco žádali a dodnes to není hotové. Nemusíme v tomto případě vinit zaměstnance oddělení IT, je totiž naprosto přirozené, že člověk věci zapomíná, navíc je nutné zmínit také to, že nás dennodenně obklopuje čím dál tím větší množství informací.
2. **Náročnost řízení** - Zaměříme se také na management organizace. V aktuálním stavu totiž vedení organizace nemá absolutně žádný přehled o tom, co za požadavky se každý den řeší a zda jsou vůbec relevantní. Management organizace, tak nemůže přijímat opatření, kterým by například odprostil zaměstnance IT od nerelevantních požadavků a umožnil by jim tak se věnovat své expertní činnosti. Když se na to podíváme z jiného pohledu, neexistuje žádný přehled, jak dlouho byly jednotlivé požadavky řešeny, jaké kroky byly podniknuty nebo zda nastaly nějaké okolnosti, které zabránily vyřešení požadavku.
3. **Nemožnost delegování** - Tento problém úzce souvisí s problémem uvedeném v prvním bodě. Vedoucí oddělení IT nemá jednotný přehled o všech ať už vyřešených či nevyřešených požadavcích a nemůže práci efektivně rozdělovat a delegovat. Někdy se navíc stává, že dvě osoby řeší stejný požadavek nezávisle na sobě a tráví tak dvojnásobek času nad řešením jednoho problému, což je z pohledu řízení velmi neefektivní. (Možná by mohl vzniknout dotaz - Jak se stane, že oba zaměstnanci řeší stejný požadavek a jeden o druhém neví? Abychom mohli odpovědět na tento dotaz, je nutné se vrátit výše v textu a podívat se na způsoby zadávání požadavků. Zmíněna situace nastane ve chvíli, kdy jednomu zaměstnanci IT oddělení žadatel řekne o svém požadavku na chodbě a druhému informaci o svém požadavku sdělí po telefonu, právě proto, že už nemohl čekat a požadavek byl urgentní.)

3.1.2 Způsoby řešení jednotlivých problémů

Problémy, které jsou uvedeny v sekci 3.1.1 jsem diskutoval s vedoucím oddělení IT a náměstkem pro technicko-ekonomickou část. Z této diskuze vyplynula níže uvedená řešení:

První problém, kterým je - **Absence jednoho místa pro evidenci požadavků**, se dá vyřešit:

- **Zavedením sdíleného souboru pro evidenci požadavků** - Tento

dlouho byl jednotlivý požadavek řešen. Otevřenou otázkou zůstává, jakou časovou hodnotu by měl zaměstnanec do evidence uvést, jelikož určité budou požadavky, které byly zadány v pondělí a vyřešeny až například ve středu. Znamená to, že zaměstnanec strávil dva dny řešením tohoto požadavku, nebo pouze musel na někoho čekat, například na odpověď od dodavatele, bez které by tento požadavek nemohl vyřešit.

- **Přípravou zprávy, která bude management informovat o tom, jaké požadavky byly řešeny** - Stejně tak jako první způsob řešení i tento způsob vyžaduje evidenci požadavků ať už jednotnou či oddělenou. V případě jednotné by byla příprava zprávy značně jednodušší, jelikož by nebylo nutné jednotlivé požadavky dávat dohromady a zjišťovat, zda se během slučování dat nestala nějaká chyba. Zpráva pro management by musela obsahovat všechny požadavky, a navíc také poukazovat na požadavky, které z pohledu zaměstnanců IT nebyly relevantní nebo nesouvisely s oblastí IT.
- **Vytvořením postupů pro řešení často opakujících se požadavků** - Pro implementaci tohoto způsobu řešení je téměř nutné vytvoření jednotného přehledu požadavků, jelikož bez něj by se manuály pro řešení tvořily pořád opakovaně. Po vytvoření této evidence, by se mohla vytvořit tzv. znalostní databáze, která by poskytovala postup, kterým byl požadavek vyřešen v minulosti. V této databázi by však musela být implementována funkcionality pro vyhledávání požadavků dle klíčových slov. Vytvoření takové znalostní databáze je velmi náročný proces, navíc taková databáze se musí neustále aktualizovat, aby se v ní nacházela pouze aktuální data.

Třetí problém, kterým je - **Nemožnost delegování**, se dá vyřešit (Je důležité podotknout, že níže uvedené způsoby řešení budou odkazovat na způsoby řešení prvního a druhého problému, jelikož všechny tyto problémy úzce souvisí.):

- **Přidělováním požadavků jednotlivým zaměstnancům vedoucím oddělení IT** - Konkrétní řešení spočívá v tom, že by právě vedoucí oddělení IT byl jedinou osobou, zodpovědnou za vyřešení všech zadaných požadavků a všechny požadavky by pouze delegoval. Jak již však bylo uvedeno v způsobech řešení u prvního problému, přistoupení k takovému řešení bude velmi náročné, jelikož vyžaduje spoustu kapacity vedoucího oddělení IT, který má mnoho dalších úkolů a přidělování požadavků svým podřízeným by ho mohlo značně vytížit.
- **Vzájemným informováním o řešeném požadavku** - Ze strany zaměstnanců IT by byla potřeba domluvit se na jednotném způsobu takového informování. Může se například jednat o zřízení společné skupiny na platformě, která poskytuje chatové služby. Nebo by k výměně informací mohlo docházet skrze email. Toto informování by však bylo možné také uskutečnit skrze již zmíněný společný soubor, do kterého by měli všichni zaměstnanci IT přístup.

3.1.3 Shrnutí aktuálního stavu

Aktuální stav má mnoho nedostatků, které jsou uvedeny v sekci 3.1.1, jež mají vážné dopady na organizaci. Abych zjistil, jak dopady pocítuje management organizace, uskutečnil jsem rozhovor s náměstkem pro technicko-ekonomickou část, kterého jsem se zeptal na několik níže uvedených otázek. Tyto otázky byly zvoleny tak, aby z odpovědí bylo možné navnímat názor vrcholového managementu na uvedenou problematiku řešení uživatelských požadavků.

1. Otázka:

- Jaký dopad má absence systematizovaného řešení pro zadávání požadavků na management organizace?
- Odpověď: *"První věc je ta, že management nevidí, jestli požadavky, které jdou směrem k IT jsou relevantní nebo ne. Další problém je ten, že chybí jakékoliv pořadí důležitosti požadavků a za třetí chybí kontrola a zpětná vazba na práci IT oddělení."*

2. Otázka:

- Byla vůbec někdy snaha implementovat v organizaci helpdesk?
- Odpověď: *"Ze strany managementu tato snaha byla, ale narazilo to na personální a technické překážky na straně bývalých zaměstnanců oddělení IT."*

3. Otázka:

- Přáli byste si dělat reporting?
- Odpověď: *"Ano určitě ano, kolik požadavků bylo obdrženo apod. Bylo by také fajn mít možnost na tyto reporty nahlížet do většího detailu."*

4. Otázka:

- Byl by to problém pokud by implementovaný helpdesk měl cizojazyčné rozhraní?
- Odpověď: *"Pokud by prostředí helpdesku bylo cizojazyčné jednalo by se o problém, helpdesk by v takovém případě nebyl používán."*

5. Otázka:

- Myslíte si, že by lékaři měli problém s používáním takového nového softwaru?
- Odpověď: *"Ano, problém by to ze začátku rozhodně byl, myslím si, že lékaři jsou již tak přetěžováni administrativními povinnostmi. Pokud by to však v dlouhodobém horizontu mělo přinést zlepšení v procesu odbavování požadavků, domnívám se, že by se to doktoři a doktorky mohli naučit. Případně by obsluhu tohoto softwaru mohli delegovat na sestry."*

6. Otázka:

- Preferoval byste nasazení nějakého již hotového řešení, nebo řešení které bude vyvinuté specificky pro potřeby tohoto zařízení?
- Odpověď: *"Preferoval bych cestu vlastního vývoje nejen z pohledu ekonomické výhodnosti, ale i z toho důvodu, že systém bude možné kdykoliv upravit dle potřeb zaměstnanců. Navíc v aktuální chvíli musíme více investovat do přístrojů, vybavení ambulancí a také do personálu."*

Dle výše zodpovězených otázek můžeme vidět, že i management společnosti pocítuje nutnost změny a zavedení systematizovaného řešení v této oblasti. Vzhledem k provedené analýze, identifikovaným problémům a způsobům jejich řešení je tímto systematizovaným řešením software, který je označován jako helpdesk. Z jednotlivých odpovědí pak vyplývají kritéria, která budou důležitá při hodnocení jednotlivých systémů. Z poslední otázky je také patrné, že by management velmi ocenil implementaci helpdesku, která by byla na míru ušitá této konkrétní organizaci. Takové řešení má výhodu v tom, že data, která budou v helpdesku uchovávaná budou uložena v úložišti organizace a nemusela by tak být posílána do cloudu ¹. Teoreticky by se v případě helpdesku nemělo jednat o citlivá data, avšak mohou nastat situace, kdy například zdravotnický personál bude potřebovat pomoci s nahráním lékařské zprávy někam a tuto lékařskou zprávu omylem připojí jako přílohu k požadavku. Toto je pouze hypotetická situace, která ani nemusí nastat. Pokud by se však taková situace stala je lepší, aby tato data byla uložena lokálně a organizace tak měla jistotu, že se tyto informace nedostanou mimo organizaci.

3.2 Existující platformy pro implementaci helpdesku

Softwarový trh nabízí mnoho dostupných řešení pro jakoukoliv oblast. Domnívám se, že tomu tak bude i v případě helpdesku. Právě z tohoto důvodu jsem se rozhodl prozkoumat nabídku stávající řešení a posoudit vhodnost implementace jednotlivých produktů v zdravotnickém zařízení. Jak již bylo zmíněno, prostředí zdravotnického zařízení je velmi specifické a implementace nového systému má spoustu úskalí. Právě z tohoto důvodu musí systém nabízet uživatelské rozhraní, které zdravotnickému personálu usnadní řešení požadavků a bude uživatelsky přívětivé.

S vedoucím této práce jsme se domluvili, že provedu analýzu čtyř dostupných produktů. Takový počet produktů byl vybrán z toho důvodu, že pokud bychom se rozhodli porovnávat více než právě čtyři kandidáty, informace, které bychom získali, by nám v rámci rešerše nijak významně neprospěly, jelikož ostatní kandidáti by se nejspíše jevíli jako velmi podobní. Tyto čtyři kandidáti byli vybráni ze seznamu dostupných helpdesk řešení na webové

¹Cloudem nazýváme hardware a software datového centra. Cloud computing je pak zastřešujícím výrazem jak pro služby poskytované přes internet, tak i pro hardware a software v datových centrech, které tyto služby poskytují.[10]

stránce GetApp² poté, co dostupná řešení byla seřazena dle uživatelských recenzí a byl aplikován filtr na oblast: "Hospital & Health Care".

Při průzkumu jednotlivých řešení budu zkoumat jaké klíčové funkce nabízí jednotlivá řešení, abych následně mohl posoudit zda splňují definována hodnotící kritéria. Také se zaměřím na uživatelská rozhraní jednotlivých řešení, prozkoumám zda systém nabízí českou jazykovou mutaci, možnost propojení s jinými softwary, dostupnost on-premise řešení a možnost zasílání notifikací. V neposlední řadě provedu analýzu cenové politiky jednotlivých řešení a dostupnost technické podpory.

Provedení této analýzy mi umožní navnímat současné trendy v oboru helpdesk platform a zamyslet se nad tím, zda i přes preferenci vlastního vývoje není možné využít již existující řešení.

3.2.1 Hodnotící kritéria nabízených řešení

Pro hodnocení vybraných kandidátů jsem zvolil několik hodnotících kritérií, na základě kterých následně posoudím, jaké řešení je nejvíce vyhovující. Tato kritéria vyplývají z identifikovaných problémů aktuálního stavu a také z rozhovoru s vedoucím pracovníkem organizace.

Jednotlivá kritéria:

- **Uživatelské rozhraní:** Systém musí být dostupný v českém jazyce.
- **Cena:** Měsíční náklady na systém nesmí měsíčně přesáhnout částku 3 000 Kč. Jelikož většina softwaru v dnešní době je tzv. subscription based³, musíme vzít v potaz také počet řešitelů. V našem konkrétním případě, je požadováno, aby k systému měli přístup čtyři osoby.
- **Možnost propojení s jinými softwary:** Systém musí umožňovat přihlášení uživatele při použití identity z Microsoft Active Directory.⁴ Také musí umožňovat odesílání notifikací prostřednictvím emailů.
- **Uložení dat:** Systém musí umožňovat uložení dat v lokálním úložišti organizace. (Případně musí systém alespoň garantovat, že jeho pracovní data jsou uložena na serverech v rámci EU.)
- **Požadovaná funkčnost:** Systém musí umožňovat zakládat uživatelské požadavky, přiřadit k nim řešitele, vytvářet souhrny za definované časové období a jinou základní funkcionalitu spojenou s problematikou helpdesku. Systém také musí nabízet uživatelskou podporu v pracovní době tedy 5 dní v týdnu x 8 hodin.

²Odkaz na stránku: <https://www.getapp.com>

³Zákazník platí za službu, kterou využívá v konkrétně definovaných pravidelných časových intervalech.

⁴Active Directory (AD) je adresářová služba od firmy Microsoft, která je postavená na protokolu LDAP. Často slouží k autorizaci a autentizaci uživatelů nebo zařízení a využívat ji lze jen v prostředí operačního systému Windows.[3]

- **Webové rozhraní:** Systém musí umožňovat přístup a manipulaci prostřednictvím webového prohlížeče, jehož rok vydání je minimálně 2016.
- **Rozdělení pravomocí:** Systém musí umožňovat víceúrovňové rozdělení pravomocí. Minimálně musí být rozlišeny následující úrovně: obyčejný uživatel, řešitel, správce systému.

Pro větší přehlednost jsem vytvořil svoji stupnici hodnocení, dle které budu udělovat body za jednotlivá kritéria. Stupnice je definována následujícím způsobem: Pokud systém kritérium:

- **Splnil** - je mu přiděleno 10 bodů za toto kritérium.
- **Splnil částečně** - je mu přiděleno 5 bodů za toto kritérium.
- **Nesplnil** - je mu přiděleno 0 bodů za toto kritérium.

Následně při vyhodnocování jednotlivých řešení vytvořím tabulku, kde sečtu body za vybraná kritéria a vyhodnotím, který systém je na základě tohoto hodnocení nejlepší, tj. má nejvíc bodů.

■ 3.2.2 Jednotlivá řešení

V této sekci se zaměříme na důkladnou analýzu níže uvedených řešení. U každého řešení je nejdříve uveden obecný popis, který je převzatý z oficiálních zdrojů společnosti, jež konkrétní řešení vyvíjí. Jelikož texty na oficiálních stránkách jsou uvedeny v anglickém jazyce, rozhodl jsem se je parafrázovat do češtiny. V těchto úvodních textech firmy často uvádí termín zákaznická podpora a spokojenost zákazníků, protože se jedná o texty směřované k podnikům, kteří s pomocí tohoto softwaru budou řešit systém externího helpdesku. V našem případě to však nevadí a můžeme si místo těchto pojmů dosadit, pojmy: uživatelská podpora a spokojenost uživatelů, jelikož my se na takové systémy budeme koukat jako na prostředek pro realizaci helpdesku interního. Rozdíl mezi interním a externím helpdeskem je totiž z pohledu této analýzy bezpředmětný.

■ Freshdesk

- Odkaz na web: <https://www.freshworks.com/freshdesk/>

Freshdesk je online řešení pro práci s uživatelskými požadavky od společnosti Freshworks. Tento software umožňuje zefektivnit zákaznickou podporu vaší společnosti a pomůže vám efektivně komunikovat s uživateli při budoucím rozšiřování.[4]

1. Uživatelské rozhraní

Na oficiálních stránkách výrobce, můžeme vidět, že čeština je přítomná v seznamu podporovaných jazyků:

Here is the complete list of languages supported by Freshdesk:

Arabic	Bosnian	Bulgarian	Catalan
Chinese	Chinese (Traditional)	Croatian	Czech
Danish	Dutch	English	Estonian
Filipino	Finnish	French	German
Greek	Hebrew	Hungarian	Icelandic
Indonesian	Italian	Japanese	Korean
Latvian	Lithuanian	Malay	Norwegian
Polish	Portuguese (BR)	Portuguese/ Portugal	Romanian
Russian	Serbian	Slovak	Slovenian
Spanish	Spanish (Latin America)	Swedish	Thai
Turkish	Ukrainian	Vietnamese	

Obrázek 3.1: Seznam dostupných jazyků softwaru Freshdesk

2. Cena

Cena softwaru závisí na počtu služeb, které firma chce používat. V našem případě by nám mohla stačit základní verze softwaru, která je zdarma do 10 řešitelů, avšak je limitována co se týče funkcionality.

Free Get going for free	Growth Intuitive, industry-leading support for growing businesses	Pro <small>POPULAR</small> Advanced automation for high performance	Enterprise Fully featured with bots for enterprise-level support
\$0	\$15	\$49	\$79
Up to 10 agents	/agent/month, billed annually \$18 /agent/month, billed monthly	/agent/month, billed annually \$59 /agent/month, billed monthly	/agent/month, billed annually \$95 /agent/month, billed monthly
START FREE TRIAL	START FREE TRIAL	START FREE TRIAL	START FREE TRIAL

Obrázek 3.2: Cenová nabídka z oficiálních stránek Freshdesk

3. Možnost propojení s jinými softwary

Freshdesk umožňuje propojení s mnoha komunikačními softwary. V nabízených softwarech nalezneme: Whatsapp, Slack, Microsoft Teams, Mailchimp. Freshdesk disponuje možností propojení s vybraným poštovním serverem pro zasílání emailových notifikací. Software také podporuje přihlášení uživatele pomocí online analogu Microsoft AD, od firmy Microsoft, který se jmenuje Microsoft Entra ID. Z tohoto hlediska je funkcionality dostačující, ale ne 100 % vyhovující, jelikož se nejedná o nativní podporu služby ověřování Microsoft AD.

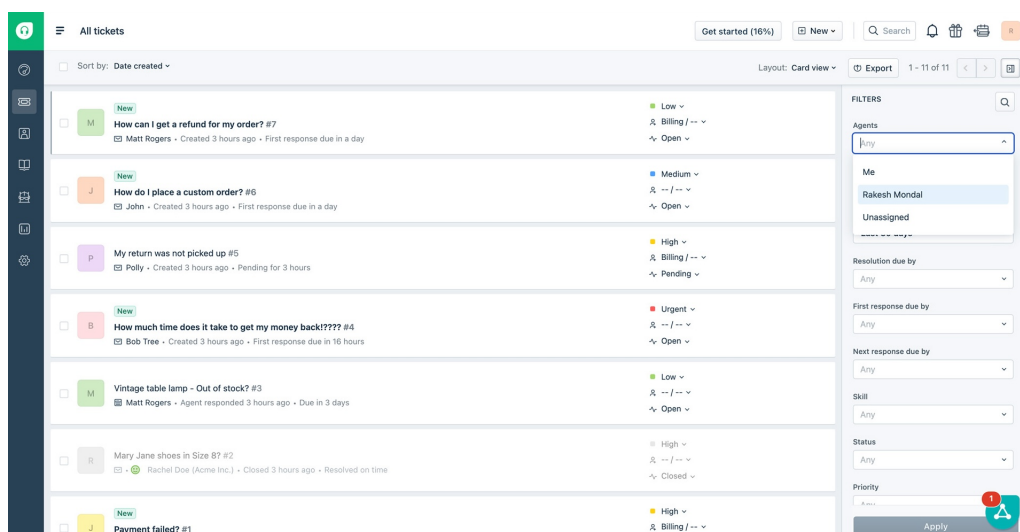
4. Uložení dat

Na webových stránkách podpory výrobce jsem se dočetl, že svůj software provozují na infrastruktuře AWS⁵. Na stránce, kde jsou uvedeny cenové nabídky systému jsem našel, že je možné si vybrat umístění datového centra pro provoz naší instance softwaru. I přes tuto skutečnost, nám však výrobce nedokáže garantovat, že data, se kterými budeme v systému pracovat budou uložena v rámci EU, jelikož infrastruktura AWS je rozprostřena napříč celým světem. Navíc právě kvůli běhu softwaru na platformě AWS, neumožňuje výrobce softwaru lokální uložení dat v rámci úložiště organizace.

5. **Požadovaná funkčnost** Systém nabízí široké spektrum možných funkcionalit a dostupných doplňků. Včetně přiřazování požadavků, znalostní databáze, komplexního analytického řešení a automatizace, avšak je důležité podotknout, že rozšířená funkcionalita se uživateli odemyká pouze při pořízení vyšší úrovně licence.

6. Webové rozhraní

Software Freshdesk je navržen primárně pro přístup prostřednictvím webového prohlížeče a jeho uživatelské rozhraní vypadá dle mého názoru velmi moderně a intuitivně.



Obrázek 3.3: Uživatelské rozhraní softwaru Freshdesk

7. Rozdělení pravomocí

Na webových stránkách podpory od výrobce jsem se dočetl, že systém umožňuje širokou škálu možností jak nastavit pravomoce pro jednotlivé uživatele systému. V základním módu software nabízí následující čtyři role:

⁵Amazon Web Services, zkráceně AWS, je komplexní platforma cloudových služeb poskytovaná americkou společností Amazon. Nabízí on-line přístup k výpočetním zdrojům, úložištím, aplikacím, databázím, analytice a k mnohým dalším službám.[8]

- Řešitelé
- Vedoucí pracovníci
- Administrátoři
- Správci účtů

Navíc systém umožňuje přidávat vlastní role, u kterých je možné nastavit si přístupová oprávnění specificky podle potřeb. V tomto ohledu tedy systém také splňuje naše požadavky.

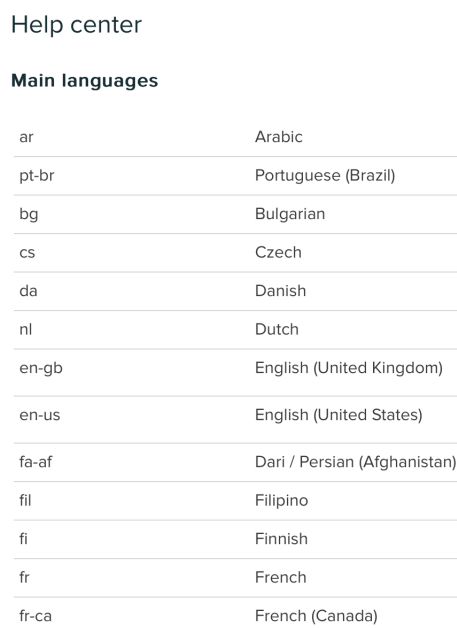
■ Zendesk

- Odkaz na web: <https://www.zendesk.com>

Zendesk nabízí jednoduché, spolehlivé a intuitivní řešení IT helpdesku, které poskytuje inteligentní a vždy dostupnou interní podporu, která zlepšuje zkušenosti zaměstnanců a zákazníků.[5]

1. Uživatelské rozhraní

Na oficiálních stránkách podpory výrobce jsem dohledal, že vybraný software je nabízený v českém jazyce:



Help center	
Main languages	
ar	Arabic
pt-br	Portuguese (Brazil)
bg	Bulgarian
cs	Czech
da	Danish
nl	Dutch
en-gb	English (United Kingdom)
en-us	English (United States)
fa-af	Dari / Persian (Afghanistan)
fil	Filipino
fi	Finnish
fr	French
fr-ca	French (Canada)

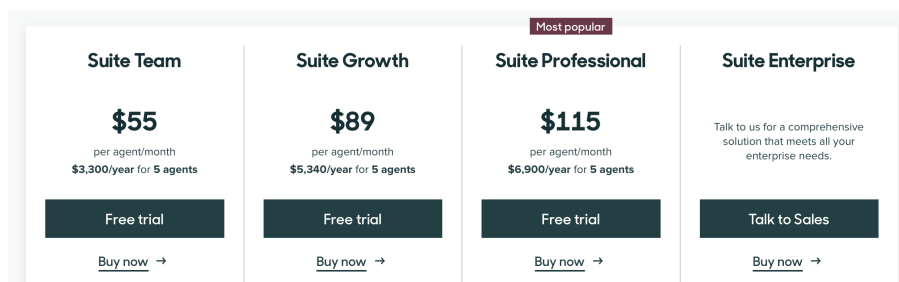
Obrázek 3.4: Seznam dostupných jazyků softwaru Zendesk

2. Cena

Obdobně jako u předešlého řešení závisí cena na počtu funkcionalit, které chce organizace využívat. Pro náš konkrétní případ opět bude stačit základní (nejlevnější) verze softwaru. Ve srovnání s předešlým softwarem

3. Analýza současného stavu řešení požadavků

je však cena výrazně vyšší, a navíc je počet řešitelů pro níže uvedené ceny limitován na 5 osob.



Obrázek 3.5: Cenová nabídka z oficiálních stránek Zendesk

3. Možnost propojení s jinými softwary

Software Zendesk nabízí široké spektrum dostupných propojení nejen s komunikačními softwary. Navíc platforma nabízí mnoho API endpointů, které mohou být užitečné, jak pro odesílání notifikací pomocí emailů, tak i pro ověření uživatele pomocí služby Microsoft AD a mohou sloužit jako výborný podpůrný nástroj pro budoucí rozšiřování funkcionality.

4. Uložení dat

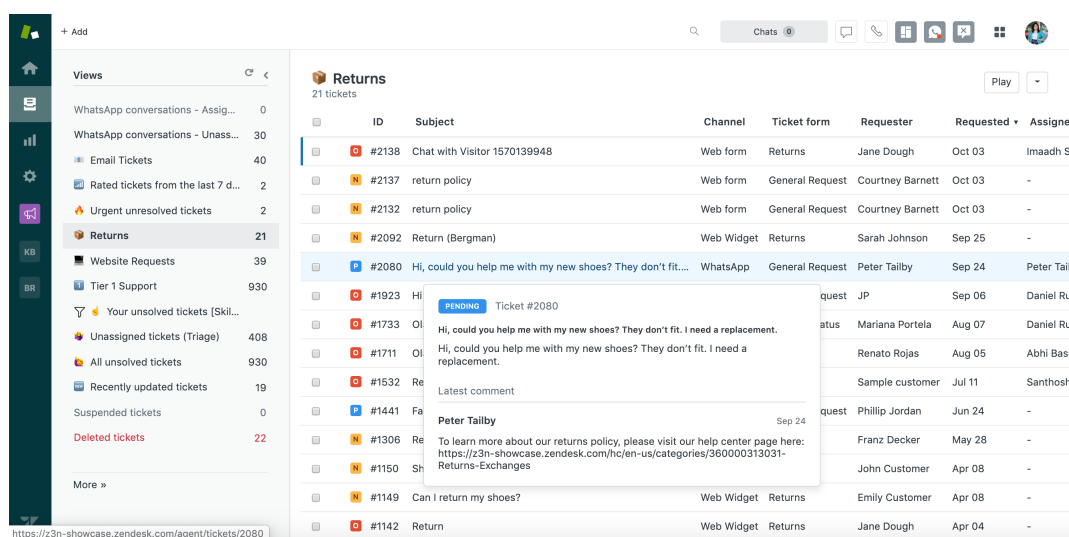
Na oficiálním blogu společnosti Zendesk jsem se dočetl, že celou svoji infrastrukturu provozují stejně tak jako předešlý uvedený software na platformě AWS. Opět se tedy potýkáme s problémem fyzického uložení dat a absencí jistoty, že data jsou uložena na území EU. Stejně tak jako u předešlého řešení existuje možnost výběru lokace výpočetních zdrojů a úložiště, avšak 100 % garance opět není poskytnutá. Navíc se v případě Zendesku jedná o příplatkovou možnost, která není obsažena v základním balíčku. Jelikož, software Zendesk je provozován na infrastruktuře AWS, jeho výrobce nenabízí ukládání dat lokálně v rámci úložiště organizace.

5. Požadovaná funkčnost

Řešení od společnosti Zendesk je velmi robustní a s přesahem pokrývá veškerou potřebnou základní funkcionalitu helpdesku. V základní cenové verzi také nabízí online podporu, což se dá považovat za velký benefit.

6. Webové rozhraní

Zendesk byl primárně navržen pro zobrazení ve webovém prohlížeči. V tomto ohledu tedy software splňuje uvedené kritérium.



Obrázek 3.6: Uživatelské rozhraní softwaru Zendesk

7. Rozdělení pravomocí

Na webu podpory výrobce jsem dohledal, že software umožňuje uživatelům přidělovat různé role. Konkrétně jsem se dočetl, že software nabízí následné rozdělení rolí:

- Koncoví uživatelé
- Řešitelé požadavků
- Administrátoři
- Správci účtů

Každá výše uvedená role má různá oprávnění a povoluje uživateli přístup do jiných částí systému. Takové řešení rozdělení rolí je pro naši potřebu dostačující a umožňuje nastavit uživatelská práva dle konkrétní potřeby.

Zoho Desk

- Odkaz na web: <https://www.zoho.com/desk/>

Zoho Desk je první kontextově orientovaný helpdesk software v oboru, který pomáhá firmám soustředit se na zákazníka. Když říkáme kontext, máme na mysli důležité informace, které všechny zúčastněné strany v oblasti služeb zákazníkům potřebují k poskytování skvělé uživatelské podpory.[6]

1. Uživatelské rozhraní

Na oficiálním webu podpory Zoho Desk, jsem našel tabulku, ve které jsou uvedené softwarem podporované jazyky. Dle níže uvedeného seznamu, můžeme vidět, že čeština se v něm vyskytuje.

Language	Code
Afrikaans	Af
Arabic	Ar
Bulgarian	Bg
Bengali	Bn
Burmese - Myanmar	My
Chinese - Simplified	Cn
Chinese - Traditional	Tw
Croatian	Hr
Catalan	Ca
Czech	Cs
Danish	Da
Dutch - Netherlands	Nl
English	En
English - United Kingdom	En

Obrázek 3.7: Seznam dostupných jazyků softwaru Zoho Desk

2. Cena

Stejně tak jako u předešlých dvou programů je cena závislá na počtu služeb, které budou v rámci softwaru dostupné. Co se cen týče jsou v rámci tohoto průzkumu průměrné a dle mého názoru adekvátní vzhledem k poskytované funkcionalitě.

Plan	Price	Features
Standard	€14 /user/month billed annually	All the essential support features to augment your customer service team.
Professional	€23 /user/month billed annually	Simplified collaboration and process automation functionalities for fast-growing teams.
Enterprise	€40 /user/month billed annually	Advanced AI and customization capabilities to enable enterprise-grade support.

Obrázek 3.8: Cenová nabídka z oficiálních stránek Zoho Desk

3. Možnost propojení s jinými softwary

Toto řešení nabízí propojení s mnoha komunikačními softwary včetně emailových služeb. Nejvíce vyzdvihnu již zmiňovaný Microsoft Teams a Slack. Na oficiálních stránkách podpory výrobce jsem také našel informaci, která potvrzuje, že tento software je možné propojit s Microsoft AD tak, aby se uživatelé mohli přihlásit pomocí identity, kterou běžně používají pro přístup do svých pracovních stanic.

4. Uložení dat

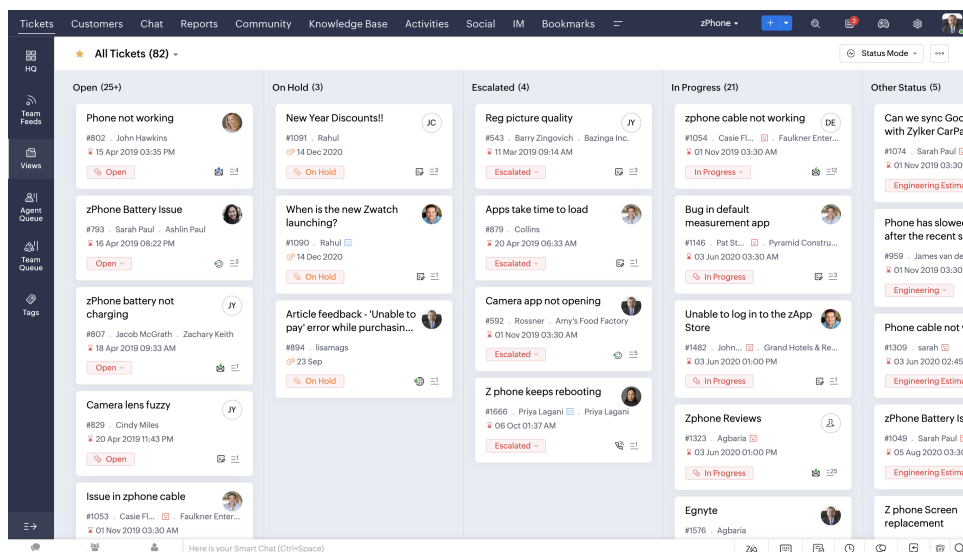
Po důkladné rešerši webových stránek výrobce a několika článků na internetu jsem nebyl schopen dohledat na jaké konkrétní infrastruktuře Zoho Desk běží. Jediná informace, která se průřezově uvádí, zmiňuje, že tento software je tzv. cloud helpdesk software, což znamená, že nejspíše běží na nějaké cloudové infrastruktuře, avšak není možné dohledat kde konkrétně. Na rozdíl od výše uvedených řešení, Zoho Desk neumožňuje zákazníkovi vybrat lokaci jeho instance softwaru. Jako zákazník si tedy nemůžeme být jistí, zda naše data jsou uložena na území EU. Tím, že je infrastruktura softwaru provozována v cloudu, ani tento výrobce nenabízí lokální ukládání dat v rámci úložiště organizace.

5. Požadovaná funkčnost

Uvedený software zcela naplňuje požadavky v tomto kritériu. V nejlevnější cenové edici, software také nabízí telefonickou i emailovou podporu, v požadované dostupnosti.

6. Webové rozhraní

Systém byl primárně vyvíjen pro zobrazení ve webovém prohlížeči, toto kritérium je naplněno.



Obrázek 3.9: Uživatelské rozhraní softwaru Zoho Desk

7. Rozdělení pravomocí

Dle webu podpory výrobce v základním nastavení systému jsou přítomny následující role:

- Koncoví uživatelé
- Řešitelé pouze s právem čtení
- Řešitelé požadavků

■ Administrátoři podpory

Seznam rolí je však upravitelný a umožňuje nastavit pravomoci dle konkrétní potřeby. Počet rolí lze také rozšířit a tím pádem si vytvořit svoji vlastní roli, která bude mít specifická oprávnění. Z těchto informací tak vyplývá, že kritérium definující rozdělení pravomocí je naplněno.

■ Intercom

- Odkaz na web: <https://www.intercom.com>

Intercom je jediné řešení, které v sobě kombinuje hned několik pokročilých funkcí, díky kterým minimalizujete provozní náklady a zároveň držíte vysoký standard služby. Mezi tyto pokročilé funkce patří např. Chatbot s umělou inteligencí a helpdesk s proaktivní podporou.[7]

1. Uživatelské rozhraní

Na stránkách výrobce jsem se dočetl, že Helpdesk od firmy Intercom má hodně omezený seznam jazyků, do kterých byl přeložen. Konkrétní seznam můžete vidět níže. Z hlediska uživatelského rozhraní tento systém nespĺňuje definované kritérium.

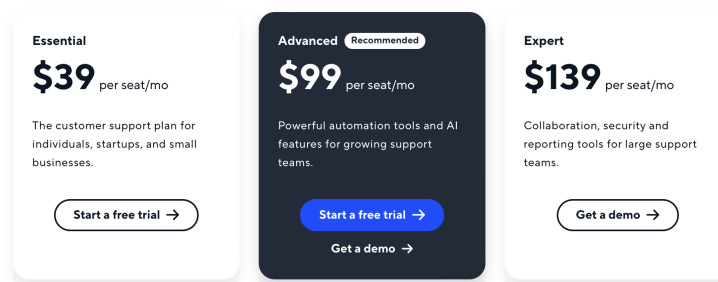
We currently support the following languages:

- English
- German
- Spanish
- French
- Portuguese

Obrázek 3.10: Seznam dostupných jazyků softwaru Intercom

2. Cena

Z pohledu ceny se tento software také jeví jako nejméně vhodný kandidát, jelikož cena za uživatele je nejvyšší z těchto čtyř kandidátů. Stejně tak jako u předešlých kandidátů zde existuje několik cenových plánů, pro naše účely bychom si bohatě vystačili s tím nejlevnějším z nich.



Obrázek 3.11: Cenová nabídka z oficiálních stránek Intercom

3. Možnost propojení s jinými softwary

Software od firmy Intercom umožňuje propojení s mnohými softwary od různých výrobců. V rámci své funkcionality systém nabízí možnost zaslání emailových notifikací s pomocí vybraného emailového serveru. Dle řešerše, kterou jsem provedl tento software však nativně neumožňuje přihlášení uživatele pomocí služby Microsoft AD. Kvůli tomu, je toto kritérium splněno jen částečně.

4. Uložení dat

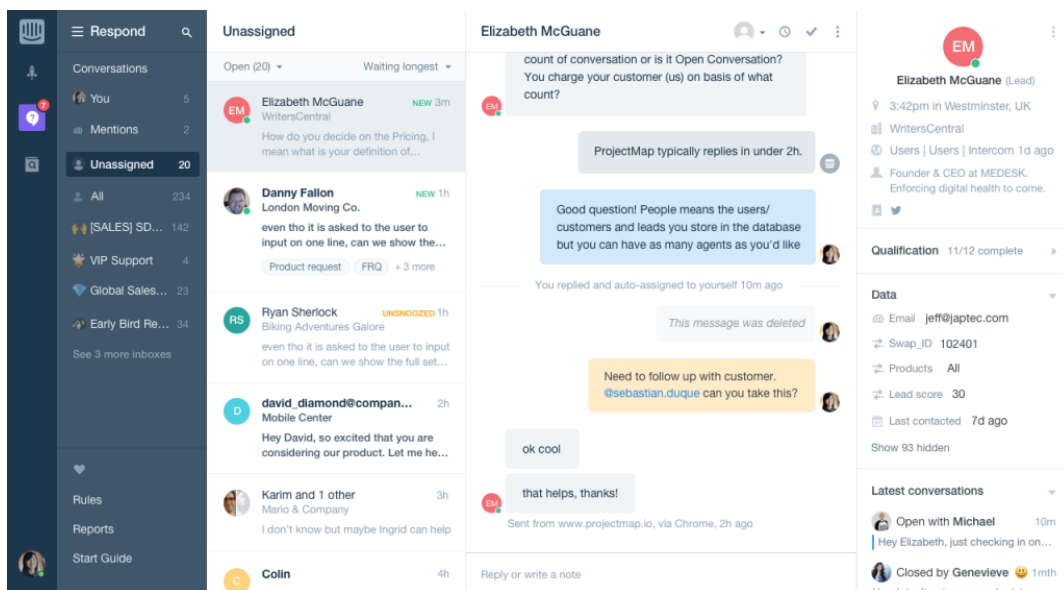
Stejně tak jako již několik uvedených alternativ, i Intecom provozuje svoji infrastrukturu na serverech AWS. V aktuální chvíli disponují data-centry v Americe, Evropě a Austrálii. Na vyžádání zákazníka je možné, aby konkrétní instance zákazníka byla provozovaná ve vybrané lokaci. Opět nám však nikdo nezaručí, že tato data budou uchovávána pouze v tomto vybraném datovém centru. Stejně tak jako předešlé alternativy, ani Intercom nenabízí možnost lokálního uložení dat v rámci úložiště organizace.

5. Požadovaná funkčnost

Systém zcela naplňuje základní požadavky, definované pro tento druh systému. Uživatelská podpora je dostupná prostřednictvím emailové komunikace a je k dispozici 24 hodin 7 dní v týdnu.

6. Webové rozhraní

Software Intercom byl vyvíjený s předpokladem, že primární platforma pro jeho použití bude webový prohlížeč. Toto kritérium je naplněno.



Obrázek 3.12: Uživatelské rozhraní softwaru Intercom

7. **Rozdělení pravomocí** Tento software umožňuje přidělit uživatelům různé role, které disponují odlišnými oprávněními. Na webu výrobce jsem dohledal, že v základu lze vybrat z následujících rolí:

- Superadmin
- Řešitel požadavků
- Manažer podpory

Systém však také umožňuje vytvoření vlastních rolí a definici specifických oprávnění pro každou z nich. Funkcionalita softwaru je tak zcela vyhovující tomuto kritériu.

3.2.3 Závěr hodnocení

V rámci porovnávání jednotlivých řešení a definovaného bodového hodnocení jsem dospěl k následující tabulce, která nám umožňuje provést kvalitativní srovnání jednotlivých softwarů. Dle stanoveného bodového hodnocení tak platí, že čím vyšší je součet bodů v posledním řádku, tím lépe daný software naplnil definovaná kritéria.

Kritérium	Freshdesk	Zendesk	Zoho Desk	Intercom
1. Uživatelské rozhraní	10	10	10	0
2. Cena	10	0	10	0
3. Možnost propojení s jednotlivými softwary	5	10	10	5
4. Uložení dat	0	0	0	0
5. Požadovaná funkčnost	10	10	10	10
6. Webové rozhraní	10	10	10	10
7. Rozdělení pravomocí	10	10	10	10
Součet	55	50	60	35

Tabulka 3.1: Bodové hodnocení jednotlivých kandidátů

Z tabulky vyplývá, že nejlepším kandidátem se jeví software Zoho Desk. Disponuje přiměřenou cenovou politikou, jeho uživatelské rozhraní je přeloženo do češtiny a plně splňuje další definovaná kritéria s výjimkou kritéria číslo čtyři (Uložení dat). Toto kritérium však nesplňuje žádný z posuzovaných softwarů. Dovolím se říct, že u výrobců softwaru, kteří svůj produkt poskytují napříč celým světem je obvyklé, že svůj software provozují na cloudové infrastruktuře, jelikož je to výhodné a poskytovatelé cloudových služeb garantují firmám vysokou dostupnost⁶ a spolehlivost.

V našem konkrétním případě však nesplnění tohoto kritéria je velmi závažným problémem, protože v helpdesku mohou kolovat patientská data, která se nesmí dostat mimo infrastrukturu organizace.

Jelikož, u všech zmíněných kandidátů se vyskytuje tento problém s uložením dat, tak se ještě více nabízí vývoj vlastního řešení. Tato možnost se mi zdá rozumná a to nejen z důvodu problematiky ukládání dat, ale také kvůli tomu,

⁶Pojem vysoká dostupnost (HA = High Availability) udává úroveň kvality služby z hlediska její uživatelské dostupnosti. Cílem služeb s vysokou dostupností je zajistit stálý provoz s minimálními výpadky. [9]

že všechna posuzovaná řešení poskytují často velmi mnoho funkcionalit, které by zaměstnanci organizace neměli v podstatě šanci využít. Všichni posuzovaní kandidáti implementují v rámci svého řešení umělou inteligenci, jelikož je to dnes moderní a mnoho dalších funkcionalit, které jsou pro účely organizace zbytečné. Zmíněná organizace, však nepotřebuje řešení, kde bude mnoho tlačítek a uživatel se v nich bude ztrácet. Tato organizace naopak potřebuje jednoduché a funkční řešení, které bude splňovat všechny potřebné definované požadavky. Navíc pokud bude tento systém vyvinutý přímo na míru pro tuto organizaci, bude velmi snadné ho později upravovat, tak aby vyhovoval i budoucím požadavkům.

Kvůli zmíněným důvodům v této kapitole a preferenci ze strany managementu bylo rozhodnuto dále pokračovat cestou vlastního vývoje a implementovat systém helpdesku pro vybrané zdravotnické zařízení na míru, dle potřeb a požadavků této organizace. Vědomosti nabyté v této analýze mi poslouží jako inspirace, kterou při vývoji vlastního řešení využiji.



Část II

Tvorba vlastního helpdesk řešení



Úvod druhé části

Zatímco první část práce byla věnována důkladné analýze současného stavu a návrhu řešení identifikovaných problémů, je druhá část práce zaměřena na implementaci vlastního helpdesk řešení.

V kapitole č. 4 jsou specifikovány funkční a nefunkční požadavky, společně s definicí uživatelských rolí a oprávnění, a také jsou zde popsány případy užití. Kapitola č. 5 se věnuje popisu vybraných technologií zvolených pro implementaci vlastního řešení. Následně v kapitole č. 6 je popsán návrh systému, včetně tvorby a testování prototypu uživatelského rozhraní. Načež pak kapitola č. 7 je věnována popisu implementace jednotlivých částí systému a nasazení aplikace. Kapitola č. 8 se věnuje uživatelskému testování a jeho výsledkům. V kapitole č. 9 je uvedeno vyhodnocení nasazení aplikace a je zde popsán její další vývoj.

Kapitola 4

Požadované vlastnosti systému

Vycházejíc z navržených způsobů řešení jednotlivých problémů byly v této kapitole specifikovány následující čtyři body, kdy každému z těchto bodů je věnována samostatná sekce v rámci této kapitoly:

- Funkční požadavky (anglicky *Functional requirements*, zkráceně FR) - FR popisují chování produktu, výsledku nebo služby, např. formou popsání procesu nebo interakce s okolím.[17]
- Nefunkční požadavky (anglicky *Non functional requirements*, zkráceně NFR) - NFR jsou doplňkem funkčních požadavků. Popisují další nezbytné vlastnosti potřebné vzhledem k prostředí a kontextu. Např. se jedná o požadavky na spolehlivost, bezpečnost, výkonost, podporu během provozu atp.[18]
- Uživatelské role a oprávnění.
- Případy užití (anglicky *Use cases*, zkráceně UC), - UC popisují (alespoň nepřímo) uživatelsky viditelné funkce a vlastnosti, které jsou základními požadavky na systém. Případy užití jsou nezávislé na programovacím jazyce. [16]

4.1 FR - Funkční požadavky

- **FR01 - Administrace uživatelů** Uživatelé v rámci systému budou mít rozdělená oprávnění, která jim zpřístupní různé funkcionality v rámci systému. Vybraní uživatelé budou moci nastavovat oprávnění jiným uživatelům.
- **FR02 - Zpracování požadavků** Uživatelé v rámci systému budou moci zadávat a měnit své požadavky. Další vybraní uživatelé budou moci přiřazovat k těmto požadavkům řešitele.
- **FR03 - Filtrování požadavků** Uživatelé budou moci nastavovat filtr, který bude ovlivňovat jaké požadavky následně uživatel uvidí.
- **FR04 - Odesílání notifikací** V definovaných krocích bude systém odesílat notifikace uživatelům.

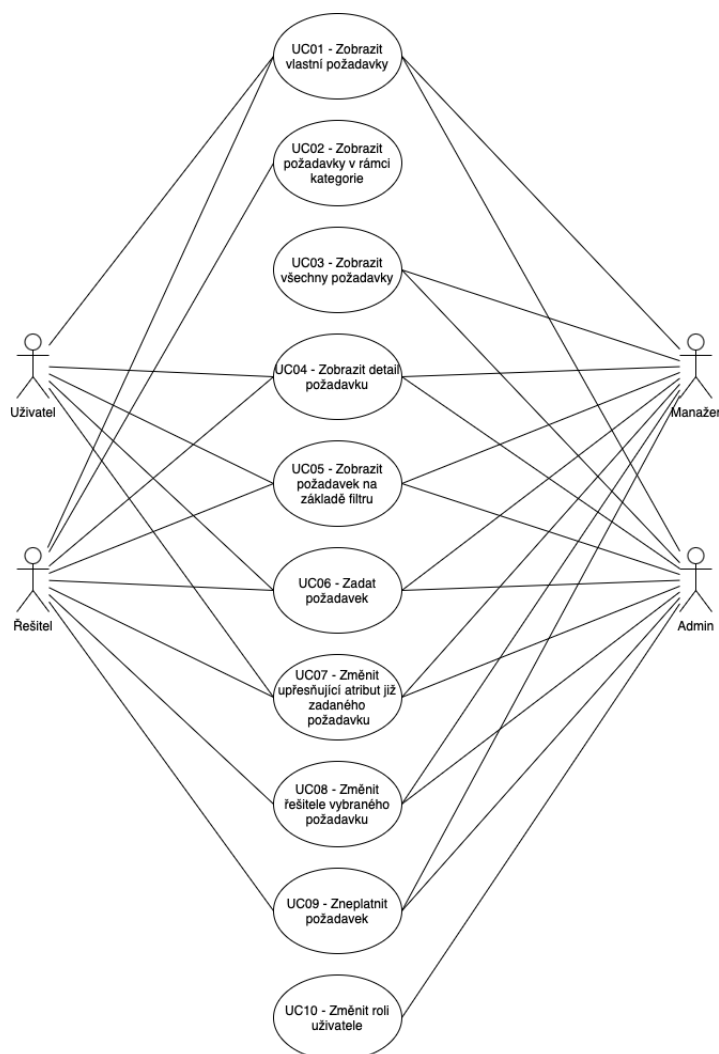
4.4 UC - Případy užití

Na začátek bych chtěl zmínit, že po dohodě s vedoucím práce vynechám případy užití, které obsahuje téměř každý informační systém, konkrétně mám na mysli:

- Registrace
- Obnova hesla
- Přihlášení
- Odhlášení

Důvodem k vynechání těchto případů užití je fakt, že jejich průběh je velmi obvyklý a veřejně známý.

Všechny případy užití jsou znázorněny na níže uvedeném obrázku č. 4.1.



Obrázek 4.1: Diagram případů užití

■ UC05 - Zobrazit požadavek na základě filtru

Uživatel s jakoukoliv rolí uvidí požadavky na základě zvoleného filtru

Scénář:

1. Uživatel se dostane na domovskou stránku
2. Systém ukáže uživateli domovskou stránku s přehledem požadavků na základě role uživatele
3. Uživatel zvolí požadovaný filtr
4. Systém zobrazí přehled požadavků na základě zvoleného filtru

■ 4.4.2 Práce s požadavkem

■ UC06 - Zadat požadavek

Uživatel s jakoukoliv rolí založí nový požadavek

Scénář:

1. Uživatel se dostane na domovskou stránku
2. Systém ukáže uživateli domovskou stránku s tlačítkem pro založení nového požadavku
3. Uživatel klikne na tlačítko pro založení nového požadavku
4. Systém zobrazí uživateli formulář pro zadání nového požadavku
5. Uživatel vyplní formulář s požadovanými údaji
6. Uživatel potvrdí správnost zadaných údajů kliknutím na tlačítko *Zadat*
7. **IF** Údaje jsou zadány správně a systém zvaliduje formulář
 - a. **THEN** Systém uloží požadavek
 - b. **ELSE** Systém vyzve uživatele k opravě formuláře a následně se postup ve scénáři vrátí do kroku č. 4
8. Na základě kategorie požadavku odešle systém notifikaci vybraným řešitelům.

■ UC07 - Změnit upřesňující atribut již zadaného požadavku

Uživatel s jakoukoliv rolí změní upřesňující atribut požadavku

Scénář:

1. Uživatel se dostane na domovskou stránku
2. Systém uživateli ukáže domovskou stránku s přehledem požadavků na základě role uživatele
3. Uživatel klikne na vybraný požadavek
4. Systém zobrazí detail požadavku, na který uživatel kliknul v kroku číslo 3.
5. Uživatel klikne na požadovaný atribut a zadá požadovaný údaj

6. Uživatel potvrdí správnost zadaného údaje kliknutím na tlačítko *Změnit*
7. **IF** Údaj je zadán správně a systém zvaliduje vybraný atribut
 - a. **THEN** Systém uloží požadavek
 - b. **ELSE** Systém vyzve uživatele k opravě údaje a následně se postup ve scénáři vrátí do kroku č. 4

■ **UC08 - Změnit řešitele vybraného požadavku**

Uživatel s rolí **Řešitel/Manažer/Admin** změní řešitele vybraného požadavku

Scénář:

1. Uživatel se dostane na domovskou stránku
2. Systém uživateli ukáže domovskou stránku s přehledem požadavků na základě role uživatele
3. Uživatel klikne na vybraný požadavek
4. Systém zobrazí detail požadavku, na který uživatel kliknul v kroku číslo 3.
5. Uživatel klikne na atribut přiřazeného řešitele a vybere nového řešitele
6. Uživatel potvrdí správnost vybraného řešitele kliknutím na tlačítko *Zadat*
7. **IF** Řešitel je zvolen správně a systém zvaliduje vybraný atribut
 - a. **THEN** Systém uloží požadavek
 - b. **ELSE** Systém vyzve uživatele k opravě řešitele a následně se postup ve scénáři vrátí do kroku č. 4
8. Systém odešle přiřazenému řešiteli notifikaci o přiřazení požadavku.

■ **UC09 - Zneplatnit požadavek**

Uživatel s rolí **Řešitel/Manažer/Admin** změní stav požadavku na *Neplatný*

Scénář:

1. Uživatel se dostane na domovskou stránku
2. Systém uživateli ukáže domovskou stránku s přehledem požadavků na základě role uživatele
3. Uživatel klikne na vybraný požadavek
4. Systém zobrazí detail požadavku, na který uživatel kliknul v kroku číslo 3.
5. Uživatel klikne na atribut stavu a změní jej na stav *Neplatný*
6. Uživatel potvrdí správnost vybraného stavu kliknutím na tlačítko *Zadat*
7. Systém skryje požadavek a nebude ho zobrazovat v žádném z přehledů

■ 4.4.3 Správa uživatelů

■ UC10 - Změnit roli uživatele

Uživatel s rolí **Admin** změní roli vybraného uživatele

Scénář:

1. Uživatel se dostane na stránku s přehledem uživatelů
2. Systém uživateli ukáže stránku se všemi uživateli v systému
3. Uživatel klikne na vybraného uživatele
4. Systém zobrazí detail uživatele, na který uživatel kliknul v kroku číslo 3.
5. Uživatel klikne na atribut role u vybraného uživatele a zadá požadovaný údaj
6. Uživatel potvrdí správnost zadaného údaje kliknutím na tlačítko *Změnit*
7. **IF** Role je zvolena správně a systém zvaliduje vybraný atribut
 - a. **THEN** Systém uloží změnu role
 - b. **ELSE** Systém vyzve uživatele k opravě role a následně se postup ve scénáři vrátí do kroku č. 4

Kapitola 5

Výběr technologií

Při vývoji vlastní aplikace je nutné se zamyslet nad technologiemi, které budeme používat. Tento výběr může být pro každého velmi individuální, avšak v této kapitole bych chtěl vysvětlit proč jsem zvolil právě tyto vybrané technologie a jak jsem při výběru postupoval. Je velmi důležité zmínit, že helpdesk aplikace, již v rámci této bakalářské práce implementuji bude webovou aplikací, která bude mít rozdělený Backend a Frontend.

5.1 Backend aplikace

Začneme backendem aplikace, který bude zodpovědný za zpracování a ukládání dat a také za tzv. logiku aplikace. V mém případě rozdělím backend na dvě části:

1. Ukládání dat
2. Zpracování uživatelských akcí a logika aplikace

5.1.1 Ukládání dat

Za účelem ukládání dat se typicky používají databáze, v případě této aplikace půjde řeč o relační databázi. Relační databáze představují typ databáze, který ukládá a uspořádává datové body s definovanými relacemi pro rychlý přístup. [11]. Také je důležité zdůraznit, že do relačních databází není možné ukládat objekty, např.: soubory obrázky apod. Pro tyto objekty slouží speciálně navržené objektové databáze. V rámci implementace řešení helpdesku však takovou databázi potřebovat nebudeme a bude nám stačit již zmíněná relační databáze. Přirozeně zde může vzniknout dotaz jak se tedy vypořádáme se soubory v rámci naší aplikace. Odpověď bude znít následovně: v případě potřeby, soubory budeme ukládat do dedikované složky, na serveru, kde poběží naše aplikace a v relační databázi budeme ukládat pouze cestu k těmto souborům. Vycházejíc z těchto předpokladů jsem pro potřeby ukládání dat v helpdesk aplikaci zvolil databázi, která se jmenuje PostgreSQL¹. Vím, že také existují jiné databáze jako jsou: MySQL, nebo SQLite, avšak kvůli níže uvedeným důvodům jsem zvolil právě PostgreSQL:

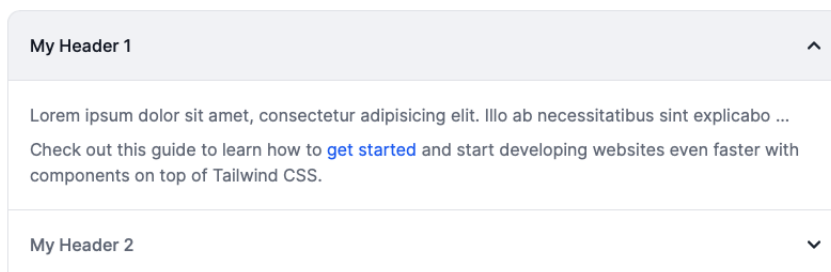
¹Odkaz na stránky databáze: <https://www.postgresql.org>

1. Hypertext Markup Language (zkráceně HTML) - značkovací jazyk, který se používá pro tvorbu webových stránek
2. Cascading Style Sheets (zkráceně CSS) - jazyk, který se používá pro změnu stylu zobrazení elementů, které se vyskytují ve značkovacích jazycích
3. JavaScript (zkráceně JS) - skriptovací jazyk, který se používá pro tvorbu dynamických webů

První dvě položky jsou více méně standardem napříč všemi weby. Situace je zajímavější u třetí technologie, čili u JavaScriptu. Zatímco spousta webů používá tzv. Vanilla JavaScript, což je slovní spojení, které označuje čistý JavaScript[14], existuje také mnoho webů, které používají ať už webové knihovny, či frameworky jako jsou: React, Angular, Vue a další. Jedná se o mnoho tzv. nadstaveb čistého JavaScriptu, které práci s tímto programovacím jazykem zpřijemňují. Diskuze na téma, která knihovna, či framework je lepší by se klidně dala pojmout formou další bakalářské práce a proto se do ní nebudu pouštět. Důvod proč jsem se o těchto frameworkcích a knihovných zmínil je však jednoduchý, jeden takový framework budu pro implementaci Frontendu používat, konkrétně se jedná o framework Svelte⁴. Důvody pro jeho použití jsou následující:

1. Svelte je open-source framework
2. Podporuje komponentové knihovny
3. Má rozsáhlou dokumentaci
4. Setkal jsem se s ním v rámci studia

Bod číslo dva vyžaduje vysvětlení co znamená pojem: "Komponentová knihovna". Framework Svelte umožňuje použití tzv. komponent, což jsou již připravené prvky, které se často používají v rámci webových aplikací. Ukázkou takového prvku může být následující Akordeon, což je rozevírací prvek viz obrázek č: 5.1:



Obrázek 5.1: Prvek Akordeon

⁴Odkaz na framework: <https://svelte.dev>

Právě dostupnost těchto prvků vývojářům často umožňuje vytvářet weby, které vypadají uživatelsky přívětivě. Další výhodou těchto komponentových knihoven je také fakt, že jsou často vyvinuty s ohledem na standardy v oblasti UI designu ⁵. Dodržování těchto standardů při tvorbě vlastních komponent může být v určitých případech velmi náročné. V rámci své implementaci jsem zvolil komponentovou knihovnu Flowbite Svelte⁶, od které si slibuji, že mi umožní implementaci uživatelského rozhraní s dodržáním všech UI standardů.

⁵Standardy a pokyny pro návrh uživatelského rozhraní jsou soubory zásad a osvědčených postupů, které pomáhají vytvářet konzistentní, použitelná a přístupná uživatelská rozhraní. Mohou být obecné, jako například Pokyny pro přístupnost webového obsahu (WCAG), nebo specifické, jako například systém Material Design společnosti Google.[15]

⁶Odkaz na stránky komponentové knihovny: <https://flowbite-svelte.com>

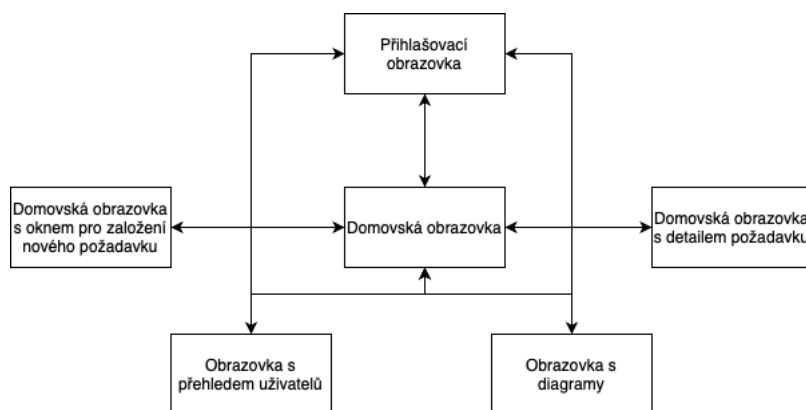
Kapitola 6

Návrh systému

Účelem této kapitoly je popis přípravy na implementaci systému. Pro představu, jak systém bude vypadat jsem vypracoval Prototyp uživatelského rozhraní, následně jsem definoval požadované entity a poté jsem tyto entity propojil vztahy, čímž mi vznikl class diagram.

6.1 Prototyp uživatelského rozhraní

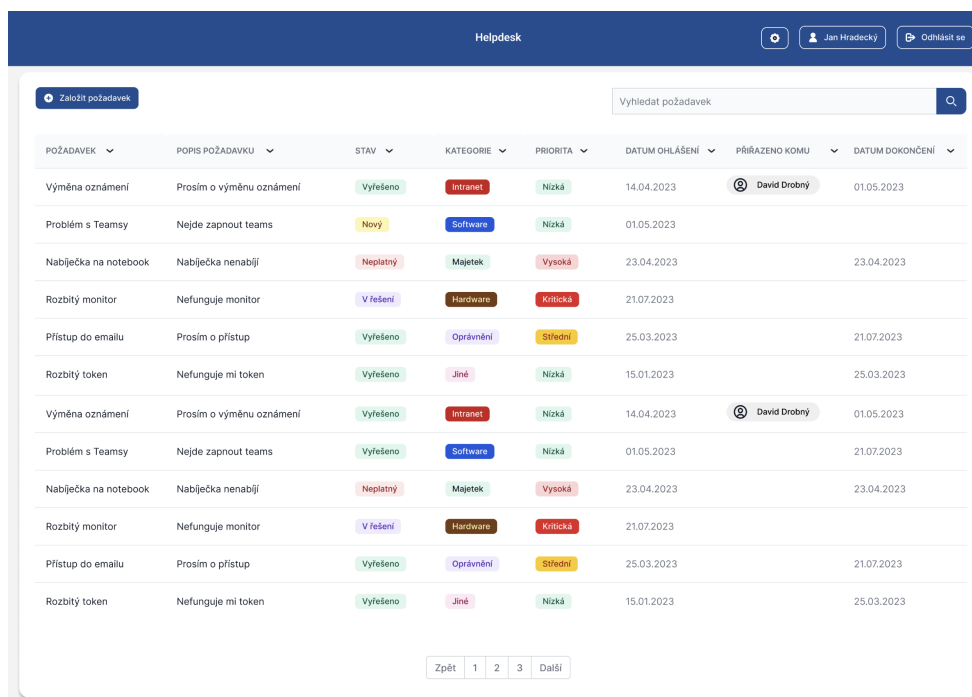
Abych mohl otestovat jak se uživatelům bude líbit uživatelské rozhraní, které jsem navrhl na základě analýzy provedené v kapitole č. 3, vytvořil jsem high-fidelity prototyp v online nástroji Figma¹ za použitím grafických komponent z již zmíněné komponentové knihovny Flowbite. Na obrázku č. 6.1 můžete vidět diagram znázorňující propojení jednotlivých obrazovek v systému.



Obrázek 6.1: Diagram propojení jednotlivých obrazovek

Obrázek č. 6.2 pak znázorňuje vzhled domovské obrazovky, která je hlavní komponentou uživatelského rozhraní.

¹Odkaz na web: <https://www.figma.com/>



Obrázek 6.2: Domovská obrazovka navrženého uživatelského rozhraní

Úplný prototyp je dostupný k nahlédnutí na následujícím odkazu².

6.1.1 Uživatelské testování prototypu

Testování prototypu se zaměstnanci vybrané organizace bylo provedeno formou rozhovoru, kdy jsem jim prvně představil prototyp, vysvětlil jsem jim co to prototyp je a následně jsem je nechal samostatně prototyp projít. Během toho co uživatelé prototyp procházeli jsem jim kladl následující připravené otázky:

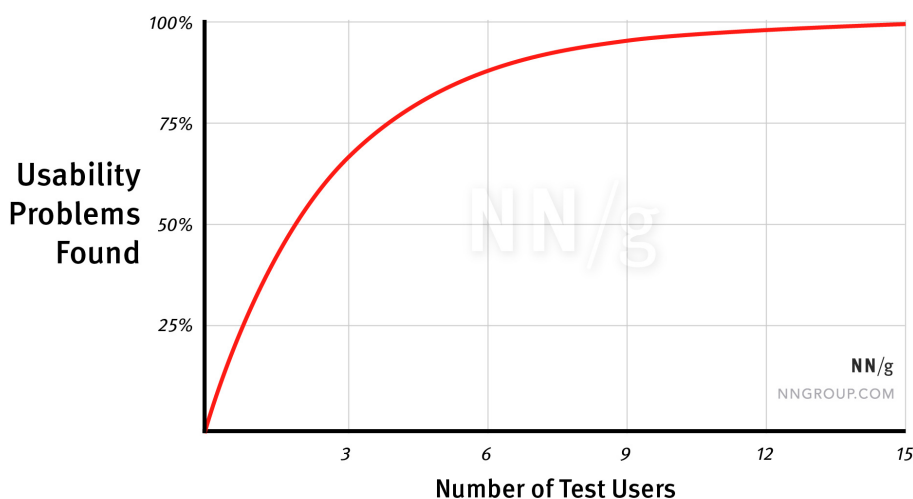
1. Přejde Vám prototyp intuitivní?
2. Byl/a jste schopen/a bez problému najít tlačítko pro založení požadavku?
3. Vyhovuje Vám rozložení na domovské stránce?
4. Vyhovuje Vám zobrazení detailu požadavku?
5. Co byste v prototypu změnil/a?

Sadu těchto otázek jsem zvolil proto, abych si potvrdil, že návrh uživatelského rozhraní je uživatelsky přívětivý, nebo naopak zjistil, že se v návrhu vyskytuje zásadní problém.

²Odkaz na prototyp uživatelského rozhraní: <https://www.figma.com/design/Ls12LeK1gBXVnv6ald145G/Prototyp-helpdesku-pro-zdravotnick%C3%A9-za%C5%99%C3%ADzen%C3%AD?node-id=0%3A1&t=VglpnTNZIDcxVd4C-1>

V rámci svého testování jsem se dotázal 5 zaměstnanců, kteří v rámci zdravotnického zařízení zastupují různé zaměstnanecké pozice. Testování se účastnil jak management organizace, tak i lékaři a další zdravotnický personál. Počet pěti zaměstnanců jsem volil na základě doporučení pana Nielsena, který jej publikuje v článku "How Many Test Users in a Usability Study?"³. Zároveň také z obrázku č. 6.3, na kterém je zobrazen diagram vypracovaný na základě výzkumné práce pana Nielsena a pana Landauera, je patrné že právě počet pěti uživatelů je ideální pro kvalitativní testování, jelikož každý další uživatel má jen velmi omezený přínos v rámci testování.

5 Users: The Optimal Sample Size for Qualitative Usability Studies



Obrázek 6.3: Diagram znázorňující závislost počtu nalezených problémů na počtu testovacích uživatelů[21]

6.1.2 Výsledky testování

Respondenti poskytli následující odpovědi na připravené otázky. Odpovědi jsou uvedeny v pořadí, které odpovídá pořadí připravených otázek.

1. respondent

1. Odpověď: "Ano, prototyp mi přijde intuitivní."
2. Odpověď: "Tlačítko je dobře umístěné a bylo snadné ho najít."
3. Odpověď: "Rozložení odpovídá současným grafickým trendům."
4. Odpověď: "Toto okno je informativně dostačující a nenavozuje přehlcení informacemi a úkony."

³Odkaz na článek <https://www.nngroup.com/articles/how-many-test-users/>

5. Odpověď: "Uvítal bych přítomnost předpřipravených šablon požadavku, dle zvolené kategorie."

2. respondent

1. Odpověď: "Ano, prototyp je intuitivní."
2. Odpověď: "Ano, byl jsem schopen tlačítko najít."
3. Odpověď: "Rozložení na domovské stránce je jednoduché a přehledné."
4. Odpověď: "Okno požadavku je přehledné a obsahuje veškeré potřebné informace."
5. Odpověď: "V aktuální chvíli mi v prototypu chybí funkcionalita komunikace v rámci požadavku, mám tím na mysli nějaké komentáře k požadavku nebo něco podobného. Dále mě nic nenapadá, ale až se systémem budu více pracovat, mohou se vyskytnout připomínky na vylepšení."

3. respondent

1. Odpověď: "Myslím si, že ano."
2. Odpověď: "S trochou rady ano, pokud člověk dostane návod jak s tím má pracovat, tak je to jednoduché."
3. Odpověď: "Rozložení mi vyhovuje."
4. Odpověď: "Vyhovuje."
5. Odpověď: "Aktuálně mě nic nenapadá, je to pro mě nová věc, ale vypadá jednoduše a použitelně."

4. respondent

1. Odpověď: "Ano prototyp je intuitivní."
2. Odpověď: "Je to jasné, tlačítko je umístěno na dobře viditelném místě."
3. Odpověď: "Ano, rozložení je přehledné."
4. Odpověď: "Ano, je to v pořádku."
5. Odpověď: "Uvítal bych popis jednotlivých kategorií a klidně i více nápovědy v rámci formuláře pro zadávání požadavku."

5. respondent

1. Odpověď: "Ano, je vidět, že tvůrce sleduje nejnovější trendy z oblasti grafického návrhu."
2. Odpověď: "Byl jsem schopen poměrně rychle najít tlačítko pro založení nového požadavku."
3. Odpověď: "Ano, rozložení je optimální"
4. Odpověď: "Ano, detail požadavku je přehledný."
5. Odpověď: "Nic bych neměnil, návrh odpovídá požadavkům organizace."

6.1.3 Vyhodnocení testování prototypu

Odpovědi, které jsem sesbíral během testování prototypu mi potvrdily, že mé navržené řešení se uživatelům zamlouvá, a je uživatelsky přívětivé. Také jsem od respondentů získal cenné informace a připomínky, týkající se uživatelské nápovědy a funkcionality komentářů, které budu v rámci implementace systému reflektovat.

6.2 Entity a class diagram

Po ověření vytvořeného prototypu jsem mohl přistoupit k návrhu samotné aplikace. Pro správnou implementaci systému je nutné si předem definovat entity, které budou v systému vystupovat. Následně je potřeba tyto entity propojit vazbami a přehledně je zobrazit formou diagramu. K těmto účelům použiji diagram tříd (anglicky *class diagram*).

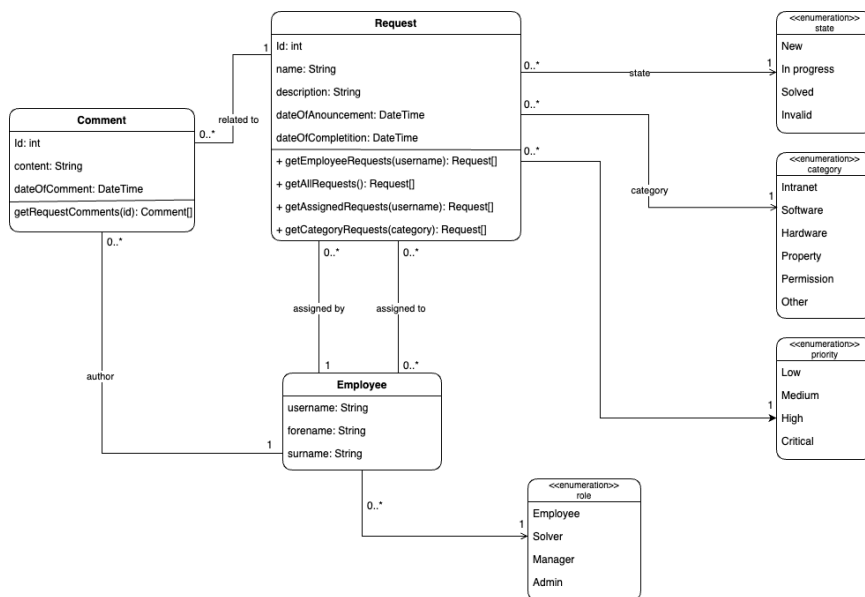
Přehled entit

Názvy entit budu níže uvádět jak česky tak anglicky, jelikož class diagram je vytvořen v anglickém jazyce.

1. Požadavek (anglicky *Request*)
2. Zaměstnanec (anglicky *Employee*)
3. Komentář (anglicky *Comment*)

Přehled atributů, které budu u každé z entit evidovat je patrný z níže uvedeného class diagramu.

Class diagram



Obrázek 6.4: Navržený class diagram systému

V rámci class diagramu je také patrné použití výčtů (anglicky *enumeration*) za účelem standardizace některých z atributů.

6.3 Autentizace a autorizace

Proces autentizace a autorizace je obvyklý a často v mnoha aplikacích stejný. V této kapitole se však nebudu věnovat přímo procesu, jak autentizace a autorizace probíhá, ale definuji specifika implementované aplikace. Nejprve je však nutné definovat co znamenají tato dvě slova:

- Autentizace - Proces ověření identity uživatele.[19]
- Autorizace - Proces ověření oprávnění uživatele.[19]

Při přihlášení do aplikace helpdesku bude uživatel autentizován vůči Active Directory organizace, pro kterou je helpdesk vyvíjen. Následně po úspěšné autentizaci, provede systém přidělení práv uživateli na základě informací, které budou uloženy přímo v databázi vyvíjeného systému. Důvodem, proč práva nebudou přidělovány přímo z Active Directory je požadavek managementu organizace, který si přeje, aby veškerá manipulace s uživatelskými oprávněními byla prováděna přímo v aplikaci.

6.4 Závěr návrhu systému

Výsledkem této kapitoly je zpětná vazba od uživatelů na vytvořený prototyp uživatelského rozhraní, která mi potvrdila, že takový prototyp je vhodný pro implementaci. Také byly v rámci kapitoly definovány entity, class diagram a byl specifikován proces autentizace a autorizace, což jsou komponenty, které jsou nezbytně nutné pro faktickou implementaci softwarového řešení

Kapitola 7

Implementace

V této kapitole se nejprve zaměřím na frontend aplikace a detaily, které v něm přibyly oproti prototypu. Dále v sekci backendu popíšu strukturu tříd aplikace. V závěru kapitoly se pak budu věnovat samotnému nasazení aplikace do produkčního prostředí a problémům, se kterými jsem se v rámci implementace potýkal.

Pro vývoj frontendu a backendu aplikace jsem používal verzovací nástroj git, což je distribuovaný verzovací systém určený pro sledování změn v souborech a koordinaci práce mezi lidmi na různých verzích těchto souborů.[20] Konkrétně jsem pro verzování své aplikace používal platformu GitHub. Odkazy na jednotlivé repositáře s kódem jednotlivých částí najdete dále v příslušných podkapitolách.

7.1 Frontend aplikace

Implementace uživatelského rozhraní byla prováděná tak aby se vzhled aplikace maximálním možným způsobem podobal prototypu z kapitoly č. 6, jelikož uživatelské názory na tento prototyp byly velmi pozitivní.

Jak již bylo zmíněno pro implementaci frontendu jsem používal framework Svelte společně s knihovnou komponent, která se jmenuje Flowbite. Právě tato skutečnost mi umožnila nabídnout koncovému uživateli vzhledově atraktivní řešení, a také mi velmi pomohla při implementaci.

Velké množství komponent je totiž v této knihově již přítomno a z toho důvodu pro finální implementaci vyžadovaly jen vzhledovou úpravu za účelem dodržení jednotné barevnosti v rámci aplikace.

Tato skutečnost však neznamená, že všechny komponenty se pouze použily a sestavily dohromady tak aby daly výsledný vzhled aplikace. V kódu frontendové části, který je dostupný v GitHub repositáři¹, je provedeno spoustu úprav tak, aby aplikace reagovala na uživatelskou interakci a zobrazovala údaje, které jsou uloženy v databázi aplikace.

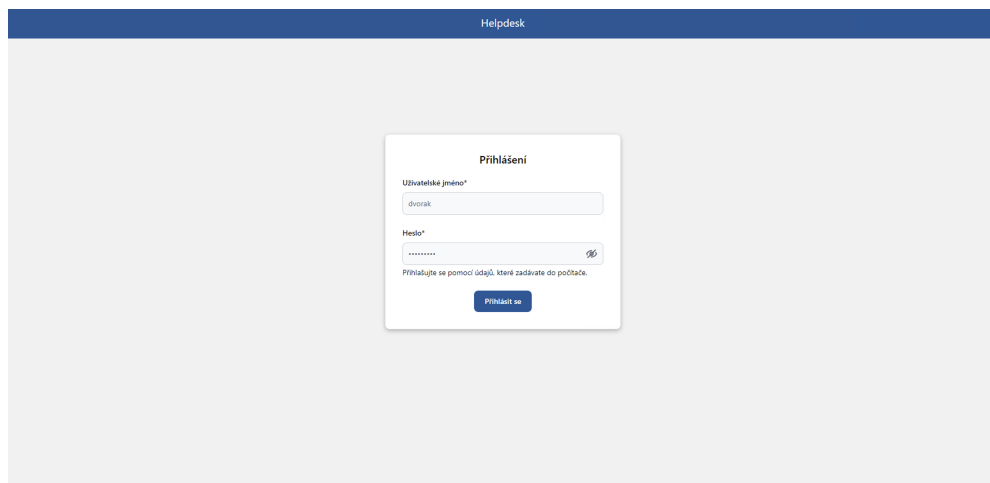
Pro čtenáře, který by nebyl seznámen s vývojem frontendu bych tedy využití komponentové knihovny popsal ve zkratce takto: "Komponenty z knihovny Flowbite výrazným způsobem zrychlují vytvoření vzhledu aplikace, tak aby

¹Odkaz na repositář (frontend): <https://github.com/Doridos/fe-helpdesk>

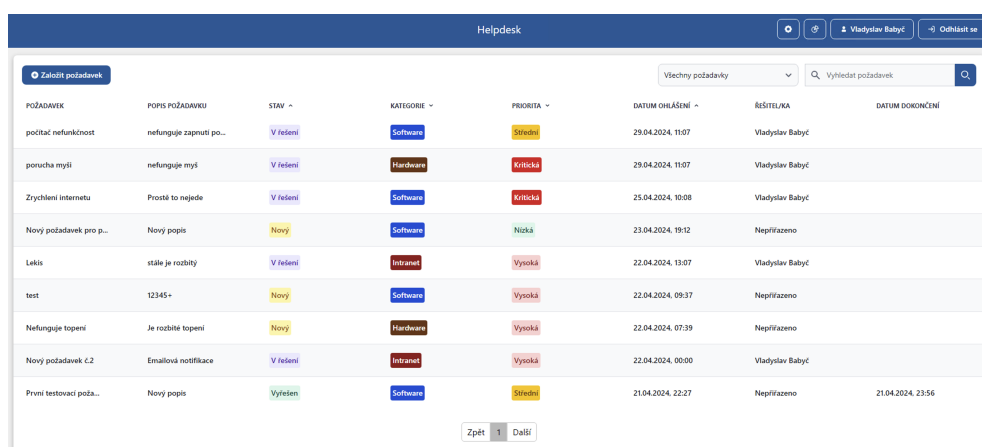
se maximálně podobala prototypu. Funkčnost této aplikace je však plně v rukou vývojáře a její implementace vyžaduje mnoho úsilí a také řádků kódu pro korektní fungování."

Níže jsou uvedeny snímky obrazovky implementované aplikace, poté co bylo provedeno její nasazení v prostředí organizace.

Obrázek č. 7.1 znázorňuje přihlašovací obrazovku aplikace.



Obrázek 7.1: Obrazovka přihlášení

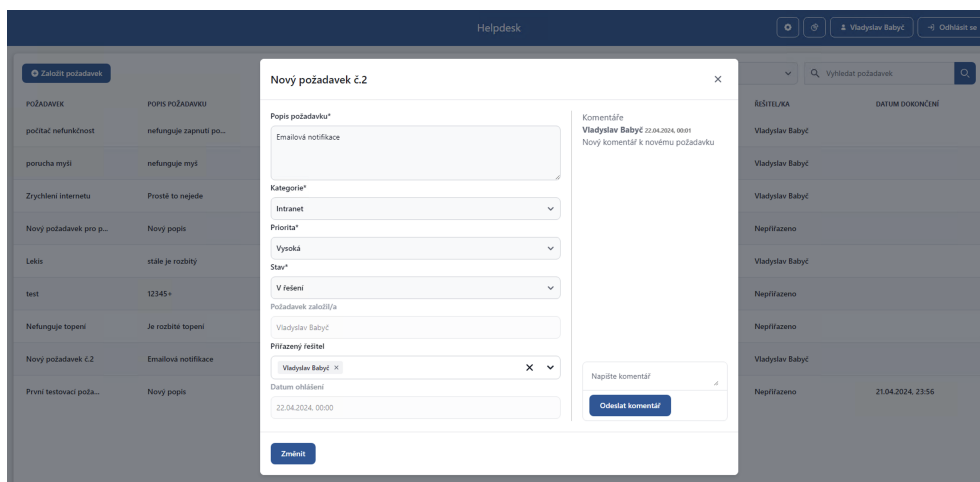


Obrázek 7.2: Domovská obrazovka s požadavky

Na obrázku č. 7.2 oproti prototypu přibyl filtr v pravé horní části, který umožňuje filtrovat požadavky na základě toho zda se mají uživatelé zobrazit:

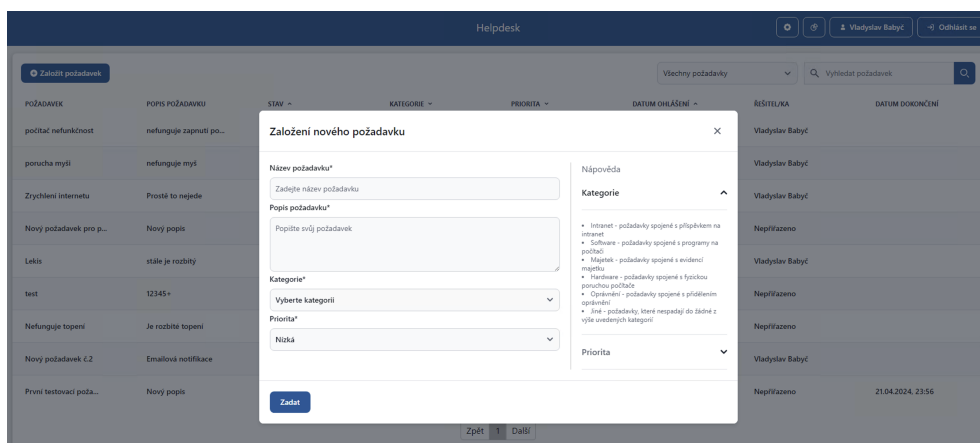
- Mé zadané požadavky
- Požadavky, které řeším
- Mé přiřazené kategorie
- Všechny požadavky

Tato zobrazení jsou také vázaná na role uživatele, což znamená, že pouze role - Admin a Manažer mohou vybrat položku: Všechny požadavky, dále pak mohou tyto role společně s rolí - Řešitel vybrat: Požadavky, které řeším a Mé přiřazené kategorie. Položku Mé zadané požadavky mohou vybrat všichni uživatelé. Filtrace těchto požadavků probíhá na backendu aplikace, čímž je zajištěna optimalizace objemu zasílaných dat přes síť.



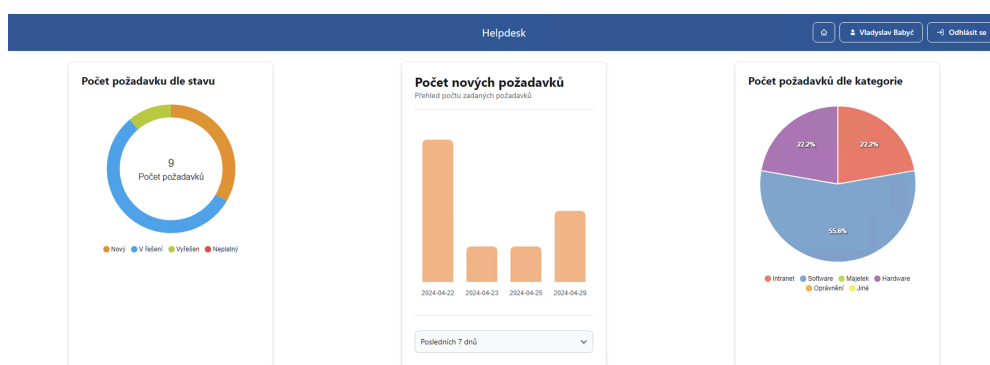
Obrázek 7.3: Obrazovka s detailem požadavku

Na obrázku č. 7.3 je vidět sloupec s komentáři, které byly přidány na základě uživatelského testování prototypu.



Obrázek 7.4: Obrazovka pro založení požadavku

Na obrázku č. 7.4 je znázorněno jak vypadá okno pro založení požadavku, a také je zde vidět sloupec s nápovědou, která taktéž jako komentáře byla přidána na základě zpětné vazby z testování prototypu.



Obrázek 7.5: Obrazovka s přehledy pro vedení organizace

Obrázek č. 7.5 znázorňuje obrazovku s přehledy, která byla přidána na základě dodatečného požadavku od vedení organizace. Tato obrazovka umožňuje vedení organizace na první pohled vidět důležité metriky ze systému. Dle mého osobního názoru se domnívám, že v budoucnu bude vedení organizace chtít, aby na této obrazovce byly zobrazeny ještě další metriky na základě jejich požadavků.

The user management interface displays a list of users with the following columns:

- uživatel:** User name (e.g., Běžný uživatel, Vladislav Babyc).
- ROLE:** Role (e.g., Admin, Ředitel, Manažer).
- PŘÍRAZENÉ KATEGORIE:** Assigned categories (e.g., Intranet, Software, Majetek, Hardware, Oprávnění, Jiné).

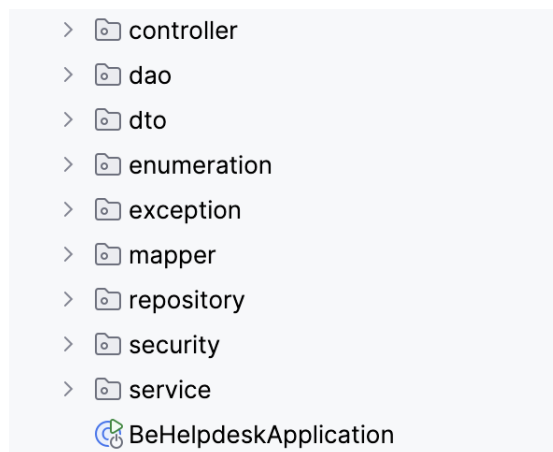
Some rows are partially obscured by a white redaction box.

Obrázek 7.6: Obrazovka s přehledem uživatelů

Na obrázku č. 7.6 je vidět obrazovka s přehledem uživatelů v systému, jeho část byla z důvodu ochrany osobních údajů zakrytá bílou barvou.

7.2 Backend aplikace

Část backendu aplikace bych chtěl uvést příložením obrázku, konkrétně níže uvedený obrázek č. 7.7 popisuje strukturu tříd aplikace.



Obrázek 7.7: Struktura tříd aplikace

Pozn.: To co vypadá jako složka se v programovací jazyce Java jmenuje anglicky - package, neboli česky balík. Dále v textu se z tohoto důvodu nebudu zmiňovat o složkách ale o balících.

Taková struktura tříd vychází z používaných praktik v návrhu backendu a nejen dle mého názoru je relativně jednoduchá pro pochopení. Pro osobu, která již někdy ve svém životě vyvíjela aplikaci s pomocí Spring frameworku bude tato struktura na první pohled pochopitelná, pro ostatní si dovoluji tuto strukturu vysvětlit:

- controller - V tomto balíku jsou přítomné třídy, které mají za úkol zpracovávat a odpovídat na uživatelské požadavky, které přijdou na uvedený aplikační endpoint.
- dao - V tomto balíku jsou přítomné Data access objects, které jsou dá se říct projekcí entit, které byly uvedeny v class diagramu, do aplikace.
- dto - V tomto balíku jsou uvedeny Data transfer objects, které jsou jakýmsi šablonami pro přenos dat ke koncovému uživateli. Při dotazu na controller by se uživateli totiž neměl vracet konkrétní objekt nýbrž právě DTO a to z mnoha důvodů, jedním z nich je třeba bezpečnost aplikace. Uvedu na příkladu: uvažujme, že máme uživatele, který má ve svém objektu uložené heslo. Při dotazu na konkrétního uživatele však nechceme dotazovateli prozradit uživatelské heslo, ale chceme mu poslat jméno a příjmení vybraného uživatele. Právě za tímto účelem je nutné vytvořit DTO, ve kterém bude chybět právě již zmíněné heslo a uvést tam pouze jméno a příjmení uživatele. Přirozenou otázkou může být: "Jak se z objektu stane DTO?" Pro tyto účely existují tzv. mappery, které objekt uživatele namapují na vybrané DTO. Ve své implementaci používám pro implementaci mapperů knihovnu MapStruct².
- enumeration - V tomto balíku jsou obsaženy číselníky definované v class diagramu aplikace.

²Odkaz na Java knihovnu: <https://mapstruct.org>

- exception - V tomto balíku jsou obsaženy vlastní třídy výjimek, které jsou potřeba v rámci implementace systému, například pro vytvoření vlastních chybových hlášek apod.
- mapper - V tomto balíku jsou obsaženy již zmíněné mappery, které mají za úkol z objektů tvořit DTO.
- repository - V tomto balíku jsou obsaženy třídy, které mají na starosti manipulaci s databází aplikace.
- security - V tomto balíku jsou obsaženy třídy, které jsou spojené s bezpečnostní konfigurací systému.
- service - V tomto balíku jsou obsaženy třídy, které provádějí veškeré manipulace s objekty, v rámci logiky aplikace.

V kódu aplikace jsou také obsaženy anotace, se kterými framework Spring pracuje. Pro příklad uvedu několik z nich:

- Běžné anotace jako jsou: `@Entity`, `@Service` a `@RestController`, které slouží pro označení jednotlivých částí aplikace.
- Anotace z knihovny Lombok³, jako jsou `@Getter`, `@Setter` a `@RequiredArgsConstructor` pro zmenšení částí opakujícího se kódu.
- Anotace jako `@Transactional`, pro znázornění, která metoda má být tzv. transakční, což je důležité při manipulaci s daty v databázi.
- Anotace `@Async`, která je použita v `EmailService`, za účelem, aby operace odesílání notifikačních emailů, která je časově náročná, nebrzdila chod aplikace.

Je důležité zmínit, že backend aplikace je pro účely přihlašování napojen na Active Directory organizace a také je napojen na poštovní server organizace za účelem odesílání notifikačních emailů.

Veškerý kód backendu aplikace je dostupný v GitHub repositáři⁴.

Pozn.: Z důvodu bezpečnosti nejsou citlivé informace na GitHubu přítomny.

7.3 Nasazení aplikace

Poté co jsem vyvinul frontend a backend aplikace jsem začal přemýšlet nad otázkou, jak takovou aplikaci v rámci prostředí vybrané organizace nasadit. Během svého studia jsem totiž již několikrát nasazoval opravdu malinké webové aplikace, které byly většinou napsané s použitím programovacího jazyka PHP a vyžadovaly pro své fungování databázi.

Z mých předešlých zkušeností vím, že každé takové nasazení bylo velmi časově náročné a často jsem narážel na nějaké problémy, kdy mé vyvinuté

³Odkaz na Java knihovnu: <https://projectlombok.org>

⁴Odkaz na repositář (backend): <https://github.com/Doridos/be-helpdesk>

řešení fungovalo u mě lokálně na mém lokálním počítači, ale nefungovalo při nasazení na server.

Přemýšlel jsem také nad tím, že má vyvinutá aplikace se nejspíše někdy bude přenasazovat z důvodu aktualizace nebo opravy chyb.

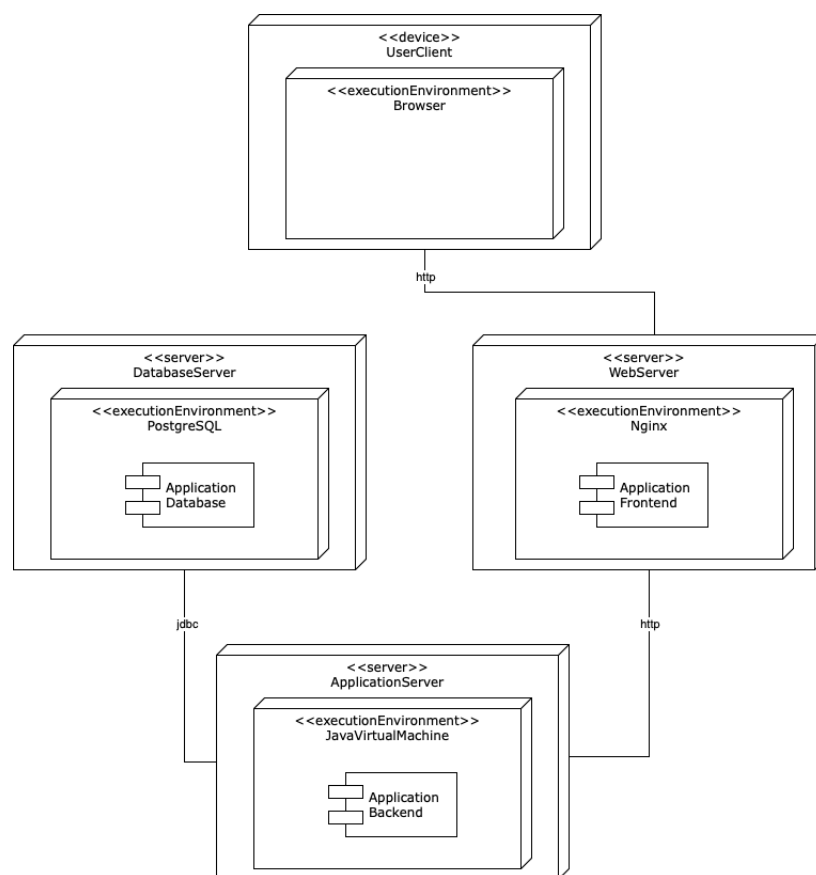
Po zvážení všech těchto skutečností jsem si uvědomil, že již několikrát během studia jsem slyšel, že existují platformy pro kontejnerizaci které mé výše popsané problémy řeší. Rozhodl jsem se proto svojí aplikaci nasazovat s pomocí právě takového řešení, konkrétně s pomocí platformy Docker. Docker je velmi rozsáhlá platforma a popis všech jejich funkcionalit by mohl naplnit mnoho dalších stran textu, z tohoto důvodu je dále pouze ve velmi obecných rysech popsána funkcionalita, kterou jsem použil během nasazení své vyvinuté aplikace.

Pro tuto platformu je potřeba vytvořit tzv. image vyvinutých aplikací. Taková image je, dá se říct, výchozím bodem pro tvorbu kontejneru, který je již to, co fakticky běží a vykonává funkcionalitu vyvinuté aplikace. Velkou výhodou Dockeru je to, že obsahuje spoustu již předpřipravených imagí, jako například PostgreSQL, které se dají spustit fakticky zadáním jednoho příkazu. Jediné co je pro to potřeba, je mít na vybraném počítači (serveru) nainstalovaný Docker, který se následně postará o vše potřebné pro hladký běh nasazované aplikace.

Docker má také výbornou platformu, která se jmenuje Docker Hub, do které se dají nahrát image aplikací a následně se odtud dají stáhnout. Pro mě byla tato skutečnost velikou výhodou jelikož jediné co jsem pro své nasazení aplikace po nahrání jejího image do Docker Hubu musel udělat bylo stažení a spuštění této image na cílovém serveru.

Dockerfily neboli soubory, které jsou potřeba pro vytvoření image, z kódu aplikace jsou uvedené jak v repositáři frontendu tak v repositáři backendu.

Pro správné fungování aplikace je potřebné spustit tři její hlavní části: databázi, backend a frontend. Začneme nejdřív spuštěním databáze, aby backend, který bude spuštěn následně, měl přístup k prostředí pro ukládání dat. Po spuštění databáze a backendu bude potřeba také spustit frontend, aby uživatelé mohli s aplikací interagovat. Diagram nasazení aplikace znázorňuje obrázek č. 7.8.



Obrázek 7.8: Diagram nasazení

V následujících částech 7.3.1 až 7.3.3 jsou uvedeny příkazy nutné pro spuštění jednotlivých komponent aplikace. Tyto příkazy je nutné zadávat do příkazové řádky na stroji, na němž je nainstalován Docker. Software Docker je k dispozici ke stažení z oficiálních stránek výrobce⁵.

7.3.1 Spuštění databáze

Jelikož kontejnery, mají své vlastní souborové systémy, je nutné vytvořit místo, které bude na těchto souborových systémech nezávislé, aby data uložena v tomto místě byla nezávislá na existenci kontejneru.

Vytvoření persistentního úložiště:

```
1 docker volume create <volume-name>
```

Spuštění databáze:

```
1 docker run --name <container-name> -p 5432:5432 -e
  POSTGRES_USER=<postgres-user> -e POSTGRES_PASSWORD=<postgres
  -password> -e POSTGRES_DB=<db-name> -v <volume-name>:/var/
  lib/postgresql/data -d postgres
```

⁵Odkaz na stránky výrobce: <https://www.docker.com>

7.3.2 Spuštění backendu

Pro spuštění backendu je také nutné vytvořit image této části aplikace a nahrát ji do Docker Hubu.

Příkaz nutný pro vytvoření image a její nahrání na Docker Hub:

```
1 docker buildx build --platform linux/amd64,linux/arm64 -t <
  username/repository:tag-name> --push .
```

Příkaz nutný pro spuštění backendu na cílovém serveru:

```
1 docker run --rm --name=<container-name> -p 8080:8080 -d <
  username/repository:tag-name>
```

V příkazech výše používám buildx což je plugin Dockeru, který maximalizuje kompatibilitu imagí napříč architekturami počítačových systémů.

7.3.3 Spuštění frontendu

Pro spuštění frontendu je nutné vytvořit image této části aplikace a nahrát ji do Docker Hubu.

Příkaz nutný pro vytvoření image a její nahrání na Docker Hub:

```
1 docker buildx build --platform linux/amd64,linux/arm64 -t <
  username/repository:tag-name> --push .
```

Příkaz nutný pro spuštění frontendu na cílovém serveru:

```
1 docker run --rm --name=<container-name> -p 80:80 -d <username/
  repository:tag-name>
```

7.4 Problémy při implementaci

Během implementace jsem narazil na několik problémů, jejichž vyřešení mi zabralo poměrně mnoho času. Z toho důvodu bych chtěl tuto sekci pojmout spíše edukační formou, jelikož člověk by se ze svých problémů měl poučit a ideálně poradit ostatním tak, aby ten stejný problém byli schopni vyřešit za kratší dobu.

1.
 - Problém: Webový prohlížeč odmítá zobrazovat nastavené styly u elementů.
 - Rada: Pečlivě si zkontrolujte cestu importu css souborů.
2.
 - Problém: Webový prohlížeč odmítá aktualizovat zobrazení na základě změněné proměnné.
 - Rada: Ujistěte se, že používáte korektní mechanismy, které budou aktualizovat vzhled webové stránky na základě změny proměnné.
3.
 - Problém: Filtry na frontendu nefungují.
 - Rada: Zkontrolujte, zda porovnáváte velikosti písmen. Případně převedte to co hledáte a vyhledávaný pojem na stejnou velikost.

4.
 - Problém: Databáze nejde spustit.
 - Rada: Pečlivě prozkoumejte, zda na stejném portu, na kterém se snažíte spustit databázi Vám neběží něco jiného.
5.
 - Problém: Backend odmítá requesty z frontendu.
 - Rada: Zkontrolujte nastavení CORS - Cross-Origin Resource Sharing a ujistěte se, že doména vašeho frontentu je v povolených doménách.
6.
 - Problém: Backend neodpovídá na zaslané requesty.
 - Rada: Ujistěte se, že v cestě nestojí něco co blokuje komunikaci mezi frontendem a backendem např.: Windows Firewall.
7.
 - Problém: Build image se nedaří kvůli nspecifikovanému erroru.
 - Rada: Zkontrolujte verzi Javy, která se pro tento build používá a ujistěte se, že je tato verze minimálně stejná jako verze Javy, ve které je aplikace napsaná.

Věřím, že výše uvedené rady Vám pomohou předejít problémům, na které jsem v rámci implementace narazil, případně pak zkrátí dobu řešení uvedeného problému.

7.5 Závěr implementace systému

V této kapitole byla popsána implementace systému a byly zde uvedeny obrázky, jež implementovaný systém znázorňují. Následně byl popsán proces nasazení aplikace v prostředí organizace a byly uvedeny problémy, na které jsem v rámci implementace aplikace narazil.

Kapitola 8

Uživatelské testování aplikace

Po nasazení aplikace bylo provedeno uživatelské testování výsledné aplikace. Pro testování aplikace byli vybráni zaměstnanci, kteří již testovali prototyp uživatelského rozhraní. Tito zaměstnanci byli vybráni, jelikož mi jsou schopni poskytnout zpětnou vazbu v souvislosti s již předvedeným prototypem. Nasazenou aplikaci tak testovalo pět zaměstnanců, což znamená, že jsem se opět držel doporučení p. Nielsena, které je podrobně vysvětleno v sekci č. 6.1.1 této práce.

Uživatelské testování probíhalo formou rozhovoru, kdy zaměstnanci bylo na začátku rozhovoru řečeno, jak se má do aplikace přihlásit a poté mu byl zadán jednoduchý úkol, který spočíval ve vytvoření požadavku. Po splnění tohoto úkolu byly zaměstnanci položeny následující otázky:

1. Podobá se dle Vašeho názoru implementace systému prototypu?
2. Vyhovuje Vám způsob přihlašování do systému?
3. Vyhovuje Vám způsob založení požadavku?
4. Jak hodnotíte implementaci funkcionality komentářů k požadavku?
5. Jak hodnotíte emailové notifikace, které systém odesílá?
6. Jsou pro Vás srozumitelné chybové hlášky, které Vám systém zobrazí v případě chybně zadaných údajů?
7. Co byste na první pohled v aktuální implementaci změnil/a?

Sadu těchto otázek jsem zvolil proto, abych zjistil, jak moc je implementace systému z pohledu uživatelů vyhovující, zda se v ní nenachází nějaké významné nedostatky a také abych ověřil svůj návrh aplikace.

8.1 Výsledky testování

Respondenti poskytli následující odpovědi na připravené otázky. Odpovědi jsou uvedeny v pořadí, které odpovídá pořadí připravených otázek.

5. Odpověď: "Emailové notifikace jsou vhodným komunikačním prostředkem pro upozornění na změnu v rámci systému."
6. Odpověď: "Chybové hlášky, které systém zobrazí v případě chybně zadaných údajů jsou výstižné a pro mě srozumitelné."
7. Odpověď: "Takto na první pohled mě nic nenapadá."

4. respondent

1. Odpověď: "Implementace se podobá prototypu."
2. Odpověď: "Způsob přihlašování mi vyhovuje, protože si nemusím pamatovat další přihlašovací údaje."
3. Odpověď: "Způsob založení požadavku mi vyhovuje, a je intuitivní."
4. Odpověď: "Komentáře jsou dobrým doplňkem v systému."
5. Odpověď: "Notifikace hodnotím pozitivně, kvůli tomu že nemusím stále koukat do helpdesku ale budu informován o změnách prostřednictvím emailu. "
6. Odpověď: "Chybové hlášky jsou srozumitelné a napoví mi co mám opravit při chybně zadaných údajích."
7. Odpověď: "Aplikace vypadá na první pohled pěkně, v aktuální chvíli bych nic neměnil."

5. respondent

1. Odpověď: "Ano, systém je s prototypem téměř totožný."
2. Odpověď: "Ano, zvolený způsob je velmi snadný, navíc pro uživatele to nepředstavuje problém v podobě dalšího hesla."
3. Odpověď: "Ano, způsob založení požadavku je intuitivní a není nic, co by bylo potřeba změnit."
4. Odpověď: "Tato funkcionality je velmi přínosná, bude možné zobrazit velmi rychle průběh řešení konkrétního požadavku."
5. Odpověď: "Tato funkcionality je vhodná pro všechny uživatele systému – pro zaměstnance, kteří požadavek zadávají i pro kolegy z oddělení IT, kteří požadavky řeší. Obsah notifikace je také dostačující."
6. Odpověď: "Ano, v této části bych nic neměnil."
7. Odpověď: "V této chvíli jsem neodhalil žádné problémy, které by vedly ke změnám v systému nebo v procesu."

8.2 Závěr uživatelského testování

Odpovědi, které mi respondenti poskytli během uživatelského testování mě ujistily v tom, že implementované řešení uživatelé hodnotí pozitivně, a že aplikace byla navržena správně. V rámci testování prototypu se vyskytla pouze jedna připomínka týkající se přidání předpřipravených šablon k založení požadavku, což je velmi cenný podnět pro přidání funkcionality v další verzi aplikace.

Kapitola 9

Vyhodnocení nasazení aplikace

Týden po nasazení aplikace do testovacího provozu v rámci organizace jsem provedl diskuzi s vedoucím a kolegy v IT oddělení, během které jsem se ptal na jejich názor na aplikaci a na to, zda jim helpdesk pomáhá v každodenní práci. Jejich odpovědi mě příjemně překvapily, jelikož všichni z nich helpdesk chválili a říkali, že nyní mohou uživatelské požadavky řešit efektivněji, protože mají přehled všech požadavků, které mají na starosti k dispozici na jedno kliknutí myši. Vedoucí oddělení si také pochvaloval přítomnost manažerského dashboardu, který poskytuje přehled o počtu a typu zadaných požadavků.

9.1 Další vývoj aplikace

Nasazením aplikace a jejím uvedením do provozu však nekončí její vývoj. Již v rámci uživatelského testování jsem obdržel návrh na přidání funkcionality do aplikace, který je relevantní a mohl by být v dalších verzích aplikace implementován. Nepochybně se také bude rozšiřovat funkcionality dashboardu, kdy do něj budou přibývat další metriky, jež budou na první pohled poskytovat lepší přehled o aktuálním stavu požadavku v helpdesku. V budoucnu také počítám s některými požadavky na úpravu, poté co aplikace již nějakou dobu bude funkční a uživatelé se s ní seznámí.

9.1.1 Škálovatelnost

Dalším tématem k řešení je škálovatelnost aplikace a rychlost její odezvy. Aktuálně aplikace využívá minimum výpočetních prostředků, konkrétně se jedná o dvě virtuální jádra a 2 GB operační paměti. S rostoucím počtem požadavků a uživatelů bude nejspíše nutné výpočetní prostředky naškálovat vertikálně neboli vylepšit výkon serveru, na kterém je aplikace provozována. Vzhledem k tomu, že databáze, která je v aktuální implementaci použita by měla fungovat rychle i při počtu požadavků evidovaných v systému v jednotkách milionů, domnívám se, že není nutné již teď uvažovat nad nějakou alternativou, jelikož očekávaný počet nově vytvořených požadavků měsíčně se pohybuje v jednotkách stovek. Pro případ, že však rychlost databázových operací se stane omezujícím faktorem pro rychlost aplikace, bude možné databázi také škálovat vertikálním způsobem.

Kapitola 10

Závěr

V této bakalářské práci jsem se zaměřil na oblast uživatelské podpory, která dle mého názoru nefungovala zcela efektivně v rámci vybraného zdravotnického zařízení. Poté jsem si svoji hypotézu ověřil provedením analýzy aktuálního stavu, ve které bylo identifikováno několik problémů a byla navržena jejich řešení. Z této analýzy mi vyplynulo, že i management organizace má na tuto vybranou oblast stejný názor a podporuje implementaci helpdesku ve zdravotnickém zařízení.

Následně jsem uskutečnil analýzu existujících platforem, které poskytují řešení helpdesku a tato řešení porovnal. Kvůli tomu, že nebyla nalezena vhodná již existující platforma a také kvůli tomu, že management organizace preferoval cestu vlastního vývoje, jsem navrhl vlastní softwarové řešení, které jsem vyvinul a v prostředí organizace nasadil.

Po nasazení aplikace proběhlo uživatelské testování, jehož výsledkem byly pozitivní reakce od uživatelů a managementu organizace. Cíle této bakalářské práce, které byly definovány v úvodní kapitole tak byly zcela naplněny.

Tato bakalářská práce mi umožnila spojit veškeré, v rámci studia získané, vědomosti, a tyto vědomosti rozšířit o praktické zkušenosti nabyté při tvorbě vlastního helpdesku. Čtenáři by práce měla přiblížit proces vývoje softwaru a prostřednictvím podrobného popisu vysvětlit jeho dílčí části.

Na základě poskytnuté zpětné vazby během uživatelského testování věřím, že implementovaný helpdesk zlepší fungování uživatelské podpory a tím vzroste spokojenost lékařů a zdravotnického personálu. Management organizace pak bude mít dostatečný přehled o tom, co se v organizaci z pohledu uživatelských požadavků děje a tím bude umožněno efektivnější řízení organizace.



Literatura

- [1] Národní zdravotnický informační portál [online]. zdravotnické zařízení | NZIP. dostupné z <https://www.nzip.cz/rejstrikovy-pojem/49>. navštíveno 30.10.2023.
- [2] ALVAO s.r.o. Helpdesk IT: Co je helpdesk a jaké jsou jeho přínosy? | ALVAO. dostupné z <https://www.alvao.com/cs/reseni/helpdesk-it>. navštíveno 30.10.2023.
- [3] MasterDC Active Directory (AD) - Co to je a k čemu slouží? | Slovník IT pojmů. dostupné z <https://www.master.cz/help/slovník/active-directory-ad/>. navštíveno 02.12.2023.
- [4] Freshworks Inc. What is Freshdesk? : Freshdesk. dostupné z <https://support.freshdesk.com/en/support/solutions/articles/228050-what-is-freshdesk->. navštíveno 03.12.2023.
- [5] Zendesk, Inc. What is an IT help desk: A guide for 2023. dostupné z <https://www.zendesk.com/internal-help-desk/it-help-desk-software/>. navštíveno 03.12.2023.
- [6] Zoho Corporation Pvt. Ltd. Get started with Zoho Desk - Enter the trial room now. dostupné z <https://www.zoho.com/desk/trial-room.html>. navštíveno 03.12.2023.
- [7] Intercom, Inc. What is Intercom? | Help Center. dostupné z <https://www.intercom.com/help/en/articles/294-what-is-intercom>. navštíveno 03.12.2023.
- [8] BUSINESS WAY MAGAZINE, Co je AWS? Amazon Web Services v kostce | Business Way Magazine. dostupné z <https://www.bway.cz/co-je-aws-amazon-web-services-v-kostce/>. navštíveno 03.12.2023.
- [9] MasterDC, Vysoká dostupnost co to je a k čemu slouží? | Slovník IT pojmů. dostupné z <https://www.master.cz/help/slovník/vysoka-dostupnost/>. navštíveno 17.12.2023.
- [10] M. ARMBRUST, A. FOX, R. GRIFFITH, A. D. JOSEPH, R. H. KATZ, A. KONWINSKI, G. LEE, D. A. PATTERSON, A. RABKIN, I. STOICA,

- M. ZAHARIA. Above the Clouds: A Berkeley View of Cloud Computing. Technical Report No. UCB/EECS-2009-28 Berkeley: Electrical Engineering and Computer Sciences University of California at Berkeley, 2009, 23 s.
- [11] Microsoft Corporation, Co je systém pro správu relačních databází | Microsoft Azure. dostupné z <https://azure.microsoft.com/cs-cz/resources/cloud-computing-dictionary/what-is-a-relational-database>. navštíveno 03.03.2024.
- [12] MioWeb s.r.o., Co je open source | Mioweb slovníček webových pojmů. dostupné z <https://www.mioweb.cz/slovnicek/open-source/>. navštíveno 03.03.2024.
- [13] Stack Exchange Inc, Stack Overflow Developer Survey 2023. dostupné z <https://survey.stackoverflow.co/2023/#section-most-popular-technologies-databases>. navštíveno 03.03.2024.
- [14] JavaTpoint, What is Vanilla JavaScript - javatpoint. dostupné z <https://www.javatpoint.com/what-is-vanilla-javascript>. navštíveno 03.03.2024.
- [15] LinkedIn, How to Use UI Design Standards and Guidelines. dostupné z <https://www.linkedin.com/advice/0/how-do-you-use-user-interface-design-standards#:~:text=User%20interface%20design%20standards%20and%20guidelines%20are%20sets%20of%20principles,Material%20Design%20system%20by%20Google..> navštíveno 03.03.2024.
- [16] S. PRESSMAN, Roger a MAXIM, Bruce. Software Engineering: A Practitioner's Approach. 8th Edition. McGraw Hill, 2014. ISBN 9780078022128.
- [17] PM Consulting s.r.o., Funkční požadavky - PM consulting. dostupné z <https://www.pmconsulting.cz/slovnikovy-pojem/funkcni-pozadavky/>. navštíveno 09.03.2024.
- [18] PM Consulting s.r.o., Nefunkční požadavky - PM consulting. dostupné z <https://www.pmconsulting.cz/slovnikovy-pojem/nefunkcni-pozadavky/>. navštíveno 09.03.2024.
- [19] ITnetwork.cz, Autentizace jménem a heslem. dostupné z <https://www.itnetwork.cz/programovani/nezarazene/autentizace-jmenem-a-heslem-ukladani-hesel>. navštíveno 17.03.2024.
- [20] ITnetwork.cz, Lekce 1 - Git - Historie a principy. dostupné z <https://www.itnetwork.cz/programovani/git/git-tutorial-historie-a-principy>. navštíveno 24.04.2024.

- [21] Nielsen Norman Group, Why You Only Need to Test with 5 Users. dostupné z <https://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>. navštíveno 11.05.2024.



Přílohy

Příloha A

Obsah přiložených souborů

A.1 Struktura příloh

Obsah přiloženého adresáře s přílohami byl popsán níže uvedenou strukturou. Tato struktura popisuje pouze hierarchické uspořádání složek se základním popisem jejich obsahu.

```
bp_attachments - základní adresář s přílohami
├── tex - zdrojové soubory bakalářské práce
│   └── images - obrázky, které byly použity v bakalářské práci
├── fe - zdrojový kód frontendu aplikace
└── be - zdrojový kód backendu aplikace
```