

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií



Diplomová práce

**Vývoj Web CRM systému a personálního profilu
uživatele založeného na architektuře mikroslužeb s
využitím frameworku Django**

Kochubei Serhii

© 2024 ČZU v Praze

ZADÁNÍ DIPLOMOVÉ PRÁCE

Serhii Kochubei

Informatika

Název práce

Vývoj Web CRM systému a personálního profilu uživatele založeného na architektuře mikroslužeb s využitím frameworku Django

Název anglicky

Development of Web CRM system and personal user account based on microservices architecture using Django framework

Cíle práce

Diplomová práce je tematicky zaměřena na problematiku CRM v prostředí WWW. Hlavním cílem práce je analýza možností nasazení webového CRM včetně pilotní implementace v oblasti zdravotnictví.

Díličními cíli práce jsou:

- vypracování přehledu CRM systémů dostupných na trhu pro lékaře (porovnání cen, funkcí a výhod různých CRM systémů).
- analýza možných přínosů webového CRM,
- vypracování metod a přístupů pro zavedení bezpečnosti při selhání jako kriticky důležité a nezbytné vlastnosti systému, který má být používán ve zdravotnictví.

Metodika

Metodika řešené problematiky diplomové práce je založena na studiu a analýze odborných informačních zdrojů. Vlastní práce je založena na analýze možností nasazení webového řešení CRM včetně pilotní implementace. Na základě syntézy teoretických poznatků a výsledků praktické části budou formulovány závěry diplomové práce.

Doporučený rozsah práce

60 – 80 stran textu.

Klíčová slova

CRM, django, python, MVT databáze, mikroslužby, informační systém, DBMS, klient-server

Doporučené zdroje informací

1. ANSHARI, Muhammad and ALMUNAWAR, Mohammad Nabil. Evaluating CRM implementation in healthcare organization. arXiv.org [online]. 17 April 2012. [Accessed 31 May 2023]. Available from: <https://arxiv.org/abs/1204.3689>
1. BAASHAR, Yahia, ALHUSSIAN, Hitham, PATEL, Ahmed, ALKAWSI, Gamal, ALZHRANI, Ahmed Ibrahim, ALFARRAJ, Osama and HAYDER, Gasim. Customer relationship management systems (CRMS) in the healthcare environment: A systematic literature review. Computer standards & interfaces [online]. August 2020. [Accessed 31 May 2023]. Available from: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7194637/>
1. MASSARO, Alessandro, MARITATI, Vincenzo, SAVINO, Nicola, GALIANO, Angelo, CONVERTINI, Daniele, DE FONTE, Emanuele and DI MURO, Maurizio. A study of a health resources management platform integrating neural networks and DSS telemedicine for homecare assistance. MDPI [online]. 19 July 2018. [Accessed 31 May 2023]. Available from: <https://www.mdpi.com/2078-2489/9/7/176>
1. RICHARDS, Mark and FORD, Neal. Fundamentals of Software Architecture. O'Reilly Online Learning [online]. [Accessed 31 May 2023]. Available from: <https://www.oreilly.com/library/view/fundamentals-of-software/9781492043447/>
1. URQUHART, James. Flow Architectures. Google Books [online]. 6 January 2021. [Accessed 31 May 2023]. Available from: https://books.google.com/books/about/Flow_Architectures.html?id=3X4SEAAAQBAJ

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

doc. Ing. Pavel Šimek, Ph.D.

Garantující pracoviště

Katedra informačních technologií

Elektronicky schváleno dne 31. 1. 2024

doc. Ing. Jiří Vaněk, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 9. 2. 2024

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 20. 03. 2024

Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Vývoj Web CRM systému a personálního profilu uživatele založeného na architektuře mikroslužeb s využitím frameworku Django" jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 30.03.2024

Poděkování

Rád bych touto cestou poděkoval svému školiteli doc. Ing. Pavlu Šimkovi, Ph.D. za podporu a poskytnuté rady při zpracování této diplomové práce.

Vývoj Web CRM systému a personálního profilu uživatele založeného na architektuře mikroslužeb s využitím frameworku Django

Abstrakt

Tato diplomová práce se zabývá vývojem webového CRM systému a personálního profilu uživatele, které jsou postaveny na architektuře mikroslužeb a využívají framework Django. V současné době, kdy digitalizace a personalizovaná péče hrají důležitou roli v poskytování kvalitních služeb, je nezbytné, aby systémy pro řízení vztahů se zákazníky (CRM) byly vybaveny nejnovějšími technologiemi a byly plně přizpůsobeny potřebám uživatelů. Cílem diplomové práce je analýza možností nasazení webového CRM včetně pilotní implementace v oblasti zdravotnictví. Práce se věnuje analýze stávajících řešení CRM, identifikaci klíčových funkcionalit potřebných pro úspěšnou implementaci v zdravotnickém prostředí a navrhuje nový model CRM systému, který využívá mikroslužby pro lepší škálovatelnost, flexibilitu a integraci s dalšími systémy. Dále se práce zaměřuje na praktický vývoj prototypu systému, včetně návrhu databáze, uživatelského rozhraní a bezpečnostních aspektů. Výsledky této práce ukazují potenciál architektury založené na mikroslužbách a Django frameworku pro vytvoření robustního, flexibilního a uživatelsky přívětivého CRM systému, který může efektivně reagovat na dynamické požadavky zdravotnického sektoru a přispět k lepší kvalitě a efektivitě poskytovaných služeb.

Klíčová slova: CRM, Django, informační systém, klient-server, mikroslužby, MVT databáze, optimalizace procesů, python, technologie ve zdravotnictví, WebCRM systémy, webové technologie.

Development of Web CRM system and personal user account based on microservices architecture using Django framework

Abstract

This master thesis deals with the development of a web CRM system and a user's personal profile, which are built on the microservices architecture and use the Django framework. Today, when digitization and personalized care play an important role in providing quality services, it is essential that customer relationship management (CRM) systems are equipped with the latest technologies and are fully adapted to the needs of users. The aim of the thesis is to analyze the possibilities of deploying web CRM, including pilot implementation in the field of healthcare. The thesis analyzes existing CRM solutions, identifies key functionalities needed for successful implementation in the healthcare environment and proposes a new model of CRM system that uses microservices for better scalability, flexibility and integration with other systems. In addition, the work focuses on the practical development of a prototype system, including database design, user interface and security aspects. The results of this work demonstrate the potential of microservices-based architecture and Django framework to create a robust, flexible and user-friendly CRM system that can effectively respond to the dynamic requirements of the healthcare sector and contribute to better quality and efficiency of services provided.

Keywords: CRM, Django, Information System, client-server, microservices, MVT databases, process optimization, python, healthcare technologies, WebCRM systems, web technologies.

Obsah

1 Úvod.....	10
2 Cíl práce a metodika	11
2.1 Cíl práce	11
2.2 Metodika	11
3 Teoretická východiska	13
3.1 Customer relationship management	13
3.1.1 Definice CRM.....	13
3.1.2 Hlavní typy CRM systémů	15
3.1.3 Data, zákony a využití umělé inteligence v kontextu legislativy	18
3.1.4 Integrace aplikací umělé inteligence do systémů CRM.....	20
3.1.5 Vymezení pojmu nemocniční informační systém	22
3.1.6 CRM systémy v medicíně.....	24
3.1.7 Implementace systému.....	26
3.1.8 Integrace umělé inteligence do systémů CRM	29
3.2 Webové technologie.....	30
3.2.1 Obecné informace o webových frameworkcích	32
3.2.2 Vizualizace informací	34
3.3 Výběr nástrojů pro implementaci	36
3.3.1 Django.....	36
3.3.2 Python	38
3.3.3 Výběr nástrojů pro implementaci	39
3.4 Vícekriteriální rozhodování při výběru CRM systému.....	40
4 Vlastní práce.....	44
4.1 Charakteristika nemocnice	44
4.2 Nedostatky v současném systému poskytování služeb	45
4.3 Charakteristika stávajících CRM systémů	45
4.4 Vícekriteriální rozhodování efektivity vývoje CRM systému	48
4.5 Směry vývoje CRM.....	56
4.6 Charakteristika funkcionality vyvinutého CRM systému	56
4.7 Create Script.....	58
4.8 Soubor forms.py	60
4.9 Models.py.....	61
4.10 Urls.py	63
4.11 Views.py.....	65
4.12 Microslužby a CSS	80

5	Výsledky	84
6	Závěr.....	87
7	Seznam použitých zdrojů	88
8	Seznam obrázků, tabulek, grafů a zkratk.....	92
8.1	Seznam obrázků	92
8.2	Seznam tabulek	92
8.3	Seznam použitých zkratk.....	93
Přílohy	94

1 Úvod

V současném světě je nezbytné poskytovat jedinečné a vysoce kvalitní služby, aby bylo možné získat a udrželi si zákazníky. Toho lze dosáhnout vývojem a implementací efektivních systémů řízení vztahů se zákazníky, které integrují pokročilé technologie a moderní přístupy k trhu. Tento přístup umožňuje budovat vzájemně výhodné a dlouhodobé vztahy. Kvalita služeb přímo ovlivňuje loajalitu stávajících zákazníků a získání nových, čímž určuje tržní podíl společnosti. Vysoká úroveň služeb snižuje náklady na poskytování služeb a zvyšuje zisk, ziskovost a konkurenceschopnost podniku. Kvalitní zákaznický servis by proto měl být hlavním prvkem při navrhování a zavádění jakýchkoli systémů a procesů v podniku.

Systémy CRM, které se v poslední době staly rozšířenými a aktivně zaváděnými technologiemi, jsou nástroji pro řízení vztahů se zákazníky. Tyto systémy uplatňují přístup zaměřený na zákazníka, který je klíčem k vytvoření úspěšného a udržitelného podniku. V každém odvětví ekonomiky je spokojenost zákazníků prioritou a poskytování pozitivních prožitků a emocí představuje jeden z nejvýznamnějších a nejnáročnějších úkolů pro všechny současné společnosti. Díky CRM jsou společnosti schopny vytvořit databázi svých zákazníků, která obsahuje informace o jejich preferencích a speciálních požadavcích, což umožňuje poskytovat služby na nejvyšší možné úrovni. Kromě toho systémy CRM pomáhají zlepšit práci prodejních týmů, systémy zpětné vazby a podpory zákazníků a mnoho dalších aspektů organizací.

Hlavní potíž při zavádění CRM systému nespočívá ani tak v nákladech na prostředky nebo čas na jeho zavedení, ale v připravenosti samotné společnosti a jejich zaměstnanců na změnu, stejně jako v jejich schopnosti efektivně využívat novou technologii. Je důležité si uvědomit, že samotný systém CRM nebude univerzálním řešením všech problémů, prosté zavedení systému do chodu organizace automaticky nezlepší procesy a nezvýší prodej. Aby systém prokázal svou efektivitu, je třeba se s ním naučit pracovat a kvalifikovaně využívat jeho možností k optimalizaci obchodních procesů. Zavedení jakékoli inovace vyžaduje, aby se změnám přizpůsobila celá organizace, zejména zaměstnanci, kteří se musejí přizpůsobit novému pracovnímu prostředí, což představuje absolvování odborného školení. Navzdory možným obtížím může být při správném přístupu k plánování a motivaci týmu proces školení úspěšný a trvat relativně krátkou dobu.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem diplomové práce je analýza možností nasazení webového CRM včetně pilotní implementace v oblasti zdravotnictví.

Dílčími cíli práce jsou:

- vypracování přehledu CRM systémů dostupných na trhu pro lékaře (porovnání cen, funkcí a výhod různých CRM systémů),
- analýza možných přínosů webového CRM,
- vypracování metod a přístupů pro zavedení bezpečnosti při selhání jako kriticky důležité a nezbytné vlastnosti systému, který má být používán ve zdravotnictví.

2.2 Metodika

Metodika řešené problematiky diplomové práce je založena na studiu a analýze odborných informačních zdrojů. Vlastní práce je založena na analýze možností nasazení webového řešení CRM včetně pilotní implementace. Diplomová práce započala definováním hlavních cílů a metodiky výzkumu. Po literární rešerši, zaměřené na sběr teoretických poznatků o CRM systémech, architektuře mikroslužeb, frameworku Django a jejich aplikacích ve zdravotnictví, následovala fáze analýzy požadavků. Během této fáze byly shromažďovány informace od potenciálních uživatelů a odborníků v oblasti zdravotnictví a IT, což umožnilo definovat specifické potřeby pro nově vyvíjený CRM systém. Na základě získaných dat byl vytvořen detailní návrh systému, který sloužil jako základ pro následnou fázi vývoje a zavedení prototypu. Vývojová fáze zahrnovala praktické programování a integraci systému, testování jeho funkčnosti a bezpečnosti a nakonec získání zpětné vazby od testovacích uživatelů pro finální úpravy a optimalizaci systému.

Teoretická část práce je věnována komplexnímu přehledu o CRM systémech, s důrazem na jejich význam a aplikace ve zdravotnictví. Zabývá se také architekturou mikroslužeb a výhodami jejího využití pro vývoj flexibilních a škálovatelných aplikací. Další kapitoly jsou zaměřeny na framework Django, jeho klíčové vlastnosti a přínosy pro rychlý vývoj webových aplikací. Teoretická část rovněž obsahuje přehled současných trendů a výzev spojených s implementací CRM systémů ve zdravotnictví, včetně otázek souvisejících s ochranou dat a integrací umělé inteligence.

Praktická část dokumentuje celý proces vývoje CRM systému od návrhu po implementaci prototypu. Charakterizuje je použité technologie, architektonické rozhodnutí, návrh databáze, vývoj uživatelského rozhraní a zabezpečení aplikace. Detailně je zde popsána metodologie vývoje, včetně výběru nástrojů a technik programování. Praktická část také obsahuje výsledky testování systému, analýzu zpětné vazby od testovacích uživatelů a diskuzi o dosažených výsledcích v porovnání s původně stanovenými cíli. Závěr praktické části prezentuje možnosti budoucího rozvoje systému a doporučení pro jeho implementaci v reálném prostředí zdravotnických zařízení.

3 Teoretická východiska

Teoretická část se zabývá vývojem CRM systému, výhodami a nevýhodami, identifikuje vlastnosti CRM systému v medicíně, zkoumá nejvhodnější jazyky pro psaní kódu.

3.1 Customer relationship management

3.1.1 Definice CRM

CRM systém neboli systém řízení vztahů se zákazníky je podnikový informační systém určený k automatizaci strategie společnosti v oblasti vztahů se zákazníky. Jeho cílem je zvýšit prodej, zlepšit marketingové úsilí a zkvalitnit služby zákazníkům prostřednictvím systematizace informací o zákaznících, jejich preferencích a historii komunikace s nimi. Toho je dosaženo vytvořením jednotného datového skladu, který poskytuje okamžitý přístup ke kompletním informacím o interakcích se zákazníky v každém okamžiku. Důležitým aspektem systému CRM je analýza shromážděných informací za účelem získání údajů potřebných pro manažerská rozhodnutí, například pro segmentaci zákazníků podle jejich významu pro společnost. Tento systém zajišťuje, že při jakémkoli kontaktu se zákazníkem prostřednictvím jakéhokoli komunikačního kanálu mají pracovníci společnosti přístup ke všem informacím o předchozích interakcích s tímto zákazníkem, což umožňuje činit informovaná rozhodnutí na základě relevantních údajů. Všechny informace o učiněných rozhodnutích jsou rovněž zaznamenány v systému, což zajišťuje integritu a kontinuitu procesu řízení vztahů.

CRM je inovativní technologie, jejímž cílem je zvýšit spokojenost, loajalitu a ziskovost zákazníků prostřednictvím získávání, rozvíjení a udržování efektivních vztahů a interakcí se zákazníky. Z pohledu zdravotnictví definovali Benz a Paddison CRM jako „přístup k poznávání pacientů za účelem vhodné komunikace a budování dobrých vztahů s cílem poskytovat včasné informace, přičemž výsledky pacientů jsou sledovány za účelem provedení nezbytných úprav“. V této studii přijímáme vyvážený pohled a definujeme CRM ve zdravotnickém prostředí jako manažerský přístup a aplikaci zdravotnických informačních technologií (HIT), která podporuje koncept péče zaměřené na pacienta. To umožňuje nemocnicím více se zaměřit na pacienty a uspokojit jejich potřeby a očekávání, zlepšit loajalitu, kvalitu služeb (SQ) a vybudovat dlouhodobý vztah (Baashar, 2020).

Za CRM systém lze vlastně považovat jakoukoli možnost řízení a účtování, která pomáhá zlepšit interakci se zákazníky. I když je historie hovorů a kontaktů vedena v papírové podobě nebo v Excelu - i to lze považovat za CRM systém, pokud vytvořené schéma evidence a kontroly funguje a umožňuje kontrolovat všechny varianty interakce se zákazníky. Takové způsoby vedení záznamů se samozřejmě stávají minulostí, protože v moderním světě si bez účinné automatizace lze jen těžko představit práci jakéhokoli podniku. A proto, když se mluví o CRM systému, obvykle se myslí speciální software.

Jinými slovy, CRM systém je jakýkoli software, který pomáhá úspěšně řídit, organizovat a plánovat práci s klienty.

CRM lze popsat jako obchodní strategii a ucelený proces, který se zabývá výběrem a řízením zákazníků za účelem dlouhodobé optimalizace hodnoty pro firmu. CRM slouží organizacím k identifikaci, získávání, uspokojování a podpoře zákazníků, k udržování dobrých vztahů se svými partnery a potenciálními i stávajícími zákazníky. Existují výzkumníci, kteří rozdělili CRM do dvou hlavních kategorií, tří skupin kategorií a čtyř typů. Výzkumník uvádí dvě hlavní kategorie a čtyři hlavní typy jako např. komplexní přístup a zastává názor, že existují nalezené typy CRM. První kategorie bude implicitně považována za jeden ze čtyř hlavních typů druhé kategorie (strategický CRM, operativní CRM a CRM založený na spolupráci). Jiní přešli ke kategorizaci CRM do tří odlišných skupin, a to, interaktivní, resp. kolaborativní, transakční neboli operativní a analytické. Buttle (2015), a Iriana a Buttle (2006) rozdělili typy CRM na strategické, operativní a analytické CRM. Buttle (2015) rozdělil typy CRM do čtyř úrovní: na strategický, analytický, kolaborativní a analytický CRM a operativní. Typy CRM lze obecně chápat prostřednictvím sledování dvou různých způsobů.

CRM je nezbytné k tomu, aby potenciální zákazník neodešel hned na začátku. V malých a středních podnicích je konkurence velmi vysoká, proto firmy vynakládají značné úsilí, aby přilákat zákazníky. Na tyto typy nákladů vyčleňují vysoké finanční částky a je důležité, aby tyto náklady byly zdůvodněny.

Dále je CRM důležité k tomu, aby firma mohla kontrolovat své zaměstnance a standardizovala práci se zákazníky. Bez společného standardizovaného systému CRM bude každý zaměstnanec pracovat i nadále tak, jak je zvyklý.

Také je důležité, aby se shromáždila statistická základna, která je nezbytná pro rozvoj každého podniku. Díky používání CRM systému se všechny pracovní informace shromažďují v jedné společné bázi ve standardizované podobě.

Existuje několik typů CRM systémů v závislosti na jejich hlavních funkcích a oblastech použití.

3.1.2 Hlavní typy CRM systémů

Tato subkapitola se zabývá hlavními typy CRM systémů, protože ze správného výběru systému bude v budoucnu postavena efektivita provozu podniku.

Strategické CRM. Strategie je lexikálně definována Grantem (1998) jako komplexní strategie pro nasazení zdrojů k dosažení jejich cílů v dlouhodobém horizontu. Tyto zdroje jsou záměrně určeny k rozvoji a udržování potenciálních vztahů a úsilí je třeba přiřazovat společně na základě ziskovosti firem (Zablah et al., 2004). Cílem je tedy vybudovat kreativní firmu dodávající zákazníkům hodnotu takovým způsobem, který je lepší, než jaký používají ostatní relevantní konkurenti (Buttle, 2015), což implicitně představuje první kategorii. Proto je hlavní důraz strategického CRM kladen na vytváření podnikové kultury, která je zaměřena na zákazníka, jehož prostřednictvím se získává konkurenční výhoda.

Provozní nebo Operativní CRM systémy. Zaměřené na automatizaci a optimalizaci provozních procesů souvisejících se zpracováním požadavků zákazníků, řízením prodeje, marketingem a službami zákazníkům. Hlavním cílem operativních CRM systémů je zvýšení loajality zákazníků v procesu přímého kontaktu se zákazníkem.

Operativní CRM se zabývá automatizací procesů u zákazníků prostřednictvím aplikací, které obsluhují mobilní a front-back office včetně podnikového marketingu, automatizace, zákaznického servisu a podpory a automatizace prodejní síly. Tomu odpovídá i zefektivnění pracovních procesů a automatizace u front office je řešeno prostřednictvím provozního CRM, které zahrnuje sběr dat, transakce a jejich vyhodnocování, zpracování a řízení pracovních postupů v odděleních služeb, marketingu a prodeje (Lun, Jinlin a Yingying, 2008). Z historického hlediska zůstalo provozní CRM hlavní součástí výdajů na provoz podniku, protože organizace neustále zavádějí automatizační systémy pro prodejní síly a vyvíjejí procesní aplikace. Vlastností operativního CRM je integrace technologií, procesů a lidí prostřednictvím systému CRM. Buttle (2015) dále uvedl, že v rámci operativního CRM se automatizují a zdokonalují jak podnikové procesy zaměřené na podporu zákazníků, tak procesy zaměřené na zákazníky.

Tyto systémy zjednodušují interakci se zákazníky, systematizují údaje o požadavcích a transakcích, samy vystavují faktury, připomínají zákazníkovi, aby se ozval a mohou mu zasílat sms-oznámení, zaznamenávat telefonní hovory atd (Lee, 2004).

Všechny transakční CRM jsou schopny:

- registrovat příchozí provoz (hovory, e-maily, požadavky na webové stránky),
- ukládat údaje o zákaznících, požadavcích, transakcích, úkolech atd. do databáze,
- automatizovat tok dokumentů v rámci společnosti,
- fixovat průběh obchodů podél prodejního trychtýře,
- připomínat plánované hovory, dopisy, schůzky,
- nastavovat úkoly a kontrolovat práci zaměstnanců.

Analytické systémy CRM. Zaměřené na analýzu údajů o zákaznících s cílem identifikovat trendy, předvídat chování zákazníků a přijímat informovanější marketingová a prodejní rozhodnutí. Analytické CRM se týká zvýšení hodnoty organizací a zákazníků, základny s využitím dat o zákaznících generovaných provozními nástroji pro řízení obchodu a výkonnosti. Kromě toho se analytické CRM zaměřuje na inteligentní vytěžování dat týkajících se zákazníků pro taktické nebo strategické účely. Analytické CRM navíc zahrnuje shromažďování, vytěžování, integraci, organizaci, zpracování, analýzu, interpretaci a využití dat z oblasti CRM dat generovaných z provozních aspektů podniků za účelem zlepšení kvality a přidané hodnoty společnosti (Buttle, 2015). Proto je integrace provozních a analytických řešení CRM nezbytná pro úspěch a efektivitu CRM. Analytické CRM se stalo nedílnou součástí několika implementací CRM. Proto se v této oblasti uplatňuje tzv., analytické CRM, které může vést společnosti k rozhodnutí, že přístupy k prodeji by měly být odlišné v různých skupinách zákazníků. Kromě toho je třeba přizpůsobit styl a obsah komunikace se zákazníky, čímž se zvyšuje možnost přijetí konkrétní nabídky ze strany zákazníků (Al-Homery, 2019).

Účelem analytických CRM je analyzovat nashromážděné informace o zákaznících a prodeji za účelem vytvoření efektivnější strategie. Na rozdíl od operativních systémů analytické systémy nejen zaznamenávají historii interakce se zákazníkem, ale také pomáhají vysledovat vzorce prodeje: z jakých zdrojů zákazníci nejčastěji nakupují, v jaké fázi je přerušeno nejvíce obchodů, jak jsou zákazníci rozmístěni podél prodejního trychtýře, a všechna tato data jsou aktualizována online, v kontextu jakéhokoli parametru.

Analytické CRM mohou:

- segmentovat zákaznickou základnu,

- určit hodnotu zákazníka,
- analyzovat jejich ziskovost,
- sledovat chování zákazníků v každé fázi transakce,
- zobrazit rozložení transakcí v prodejním trychtýři,
- analyzovat dynamiku prodeje,
- analyzovat účinnost marketingových nástrojů,
- předpovídat objem prodeje.

Systémy CRM pro spolupráci nebo kolaborativní CRM. Poskytují nástroje pro zlepšení interní spolupráce a komunikace ve firmě, což pomáhá efektivněji pracovat se zákazníky.

Dyche (2002) definuje kolaborativní CRM jako specifickou funkci, která umožňuje obousměrnou komunikaci mezi firmou a jejími zákazníky prostřednictvím nejrůznějších kanálů s cílem usnadnit a zkvalitnit interakci se zákazníky. Kolaborativní CRM zahrnuje využití služeb a infrastruktury pro spolupráci, které umožňují interakci mezi zákazníky a společností více kanály. Umožňuje interakci mezi zákazníky, podnikem a jeho zaměstnanci, důraz je kladen na udržování vztahů s partnery s cílem usnadnit koordinaci v podnikání podniku za účelem ziskovější identifikace, získání, udržení a rozvoje zákaznické základny. Kolaborativní CRM umožňuje sdílení cenných informací v rámci dodavatelského řetězce. Někteří dodavatelé technologií CRM proto vyvinuli řízení vztahů s partnery (PRM) aplikace, které umožňují společně spravovat komplexní partnery nebo obchodní kanály, ekosystémy a snížit náklady na řízení partnerů nebo prodejních kanálů. PRM umožňuje společně řídit distribuci finančních prostředků, plánovat a řídit propagační akce a měřit výsledky. Někdy se k charakteristice aplikace těchto systémů používá termín kolaborativní CRM (collaborative CRM) na interní komunikaci, například v rámci prodeje, marketingu a služeb (Buttle, 2015).

Takovéto CRM navazují komunikaci se zákazníky za účelem získání zpětné vazby. Informace získané jejich prostřednictvím pomáhají upravovat sortiment výrobků, cenovou politiku i proces obsluhy zákazníků. Například zaměstnanci call-centra cestovní kanceláře volají zákazníkům, kladou jim otázky týkající se kvality uskutečněného zájezdu do té či oné země a fixací odpovědí upravují ceny za navrhovaný zájezd.

Showroom CRM. Po přezkoumání výše uvedených typů CRM je stále potřeba komplexní CRM, kde tento typ může představovat provozní CRM jako front office pro interakci s obchodními zákazníky. Bude představovat back office jako manipulaci a

analýzu pro nahlédnutí do zákaznické databáze, sdílení těchto znalostí mezi obchodními partnery a zákazníky směrem k obchodním cílům. Lze tedy konstatovat, že tento nový navrhovaný typ CRM pro showroomy je vše v jednom. Součástí tohoto typu jsou automatizace prodeje, automatizace marketingu a automatizace služeb. Prodejní proces je propojen s marketingem, servisem, logistikou a podporou, což znamená, že prodejní procesy jsou ovlivňovány všemi vzájemně propojenými odděleními organizace, stejně jako strategie, politiky a postupy. Tento typ bude zahrnovat všechny čtyři předchozí typy CRM dohromady (strategie, provoz, analýza a spolupráce). Typ Showroom CRM má schopnost integrovat pohled na zákazníka a zároveň interakce probíhá na úrovni transakcí a na výkon společnosti. Nově navržený typ showroom CRM může být rámcem jako kompletní obchodní řešení, které začíná od vícekanálového interaktivního prodejního procesu až po proces prodeje, dodání produktu/služby zákazníkovi a sledování klienta jako celoživotního zákazníka (Al-Homery, 2019).

3.1.3 Data, zákony a využití umělé inteligence v kontextu legislativy

Novela zákona č. 20/1966 Sb. z 26. června 2001 (dále jen novela), provedená zákonem č. 260/2001 Sb., platným od 1. srpna 2001, vložení § 67a až § 67 poprvé kodifikovala všeobecnou povinnost zdravotnických zařízení vést zdravotnickou dokumentaci a současně dosti přesně vymezila pravidla pro její vedení, uchovávání, nakládání s ní a likvidaci. Novela významně sjednocuje legislativu tak, aby požadavky na zdravotnickou dokumentaci pro všechny druhy a formy zdravotnických zařízení byly stejné. V odstavci 1 paragrafu 67b je uloženo povinně všem zdravotnickým zařízením vést zdravotnickou dokumentaci (dále jen dokumentace) – tedy nejen nestátním zdravotnickým zařízením, jak tomu bylo dosud. V praxi to znamená, že vyšetřovat a léčit bez řádného vedení dokumentace je protizákonné. Vyhláška o zdravotnické dokumentaci 98/2012 Sb. VYHLÁŠKA ze dne 22. března 2012, kde Ministerstvo zdravotnictví České republiky stanoví zásady, které musí zdravotnická dokumentace splňovat s ohledem na rozsah poskytovaných zdravotních služeb a obsahuje údaje o zdravotním stavu pacienta a skutečnostech souvisejících s poskytováním zdravotních služeb pacientovi. Tato vyhláška nabyla účinnosti 1. dubna 2012 stejně jako zákon č. 372/2011Sb (Materová, 2013).

Praktické dopady legislativních změn

Standardizace zdravotnické dokumentace vede ke sjednocení požadavků na vedení zdravotnické dokumentace napříč různými typy zdravotnických zařízení, zvyšuje

transparentnost a zjednodušuje procesy, což přináší výhody jak pro zdravotnický personál, tak pro pacienty. Umožňuje lepší sledovatelnost péče a zjednodušuje případný přenos informací mezi zařízeními. Přísnější pravidla pro vedení dokumentace znamenají, že zdravotnická zařízení musejí být důkladnější v zaznamenávání a uchovávání zdravotnických informací. Nedodržení těchto pravidel může vést k právním následkům, což podtrhuje význam správného řízení zdravotnických záznamů.

V současné době se zdravotnická zařízení a lékaři setkávají s neustálým tlakem na zvyšování kvality a efektivity péče, přičemž stále více se obracejí k moderním technologiím, včetně umělé inteligence (UI) jako způsobu, jak těchto cílů dosáhnout. Automatizace záznamů pomocí UI nabízí významnou výhodu v efektivitě správy zdravotnické dokumentace. Místo tradičního manuálního zapisování, které je časově náročné a náchylné k chybám, umělá inteligence umožňuje automatické zaznamenávání dat, například výsledků laboratorních testů, přímo do elektronických zdravotních záznamů. Tento přístup nejen uvolňuje cenný čas zdravotnického personálu, ale také minimalizuje riziko chyb při zapisování (Keegan, 2022).

Kromě zefektivnění procesů je další klíčovou oblastí využití UI analýza dat. Velké objemy zdravotnických informací mohou být pro lékaře obtížně přehledné a umělá inteligence nabízí schopnost rychle a efektivně analyzovat tato data, identifikovat vzory, rizika a poskytnout podporu v diagnostickém procesu. Díky pokročilým algoritmům a schopnosti zpracování přirozeného jazyka může UI poskytovat doporučení a insighty, které by mohly být přehlednější.

Integrace umělé inteligence do zdravotnických informačních systémů také rozšiřuje možnosti podpory rozhodování pro lékaře. Systémy umělé inteligence mohou v reálném čase poskytovat doporučení založená na nejnovějších vědeckých poznatcích a specifických datech o pacientovi, což lékařům umožňuje poskytovat cílenou a efektivní péči.

Nicméně, s těmito výhodami přichází také nutnost pečlivého zvážení otázek soukromí a bezpečnosti dat. Veškeré využití umělé inteligence v zdravotnictví musí být v souladu s legislativou týkající se ochrany osobních údajů a zdravotnické dokumentace. To vyžaduje, aby systémy byly navrženy s ohledem na principy minimalizace dat, zabezpečení dat a zajištění práva pacientů na přístup k jejich informacím.

3.1.4 Integrace aplikací umělé inteligence do systémů CRM

Integrace aplikací umělé inteligence do systémů CRM představuje perspektivní cestu pro podniky, které se snaží zlepšit interakci se zákazníky, zefektivnit provoz a získat konkurenční výhodu (Dwivedi a Wang, 2022). Tato snaha však zdaleka není jednoduchá, u projektů AI napříč odvětvími byla pozorována značná míra selhání (Bughin a kol., 2018). Předchozí studie se zabývaly především problémy, s nimiž se společnosti potýkají při zavádění UI obecně (Ångström et al., 2023), přičemž pouze několik nedávných příspěvků se zaměřilo na kontext B2B marketingu a kontext CRM tak zůstal zcela neprozkoumán. Výzva, která se objevila v souvislosti se zaváděním umělé inteligence, se týká technických předpokladů nezbytných pro efektivní využití. To zahrnuje potřebu přístupu k rozsáhlým a vysoce kvalitním souborům dat a nezbytnou technologickou infrastrukturu pro zpracování dat, což jsou předpoklady, které jsou zásadní pro plné využití potenciálu AI (Keegan et al., 2022). Na rozdíl od samostatných aplikací AI vyžaduje AI v rámci CRM bezproblémovou integraci se stávajícími platformami a databázemi CRM a často zahrnuje komplexní datové prostředí a minimální narušení. CRM navíc vyžaduje důraz na škálovatelnost a požadavky na přizpůsobení (Perna a Baraldi, 2014), což může být složité z hlediska mapování dat, synchronizace, aktualizace v reálném čase a konfigurace, a může to vést ke specifickým problémům.

Integrace aplikací umělé inteligence do systémů CRM přináší s sebou řadu skutečností, které vyžadují nejen technickou připravenost, ale také hluboké pochopení potřeb zákazníků a specifik trhu. Zatímco výhody, jako je zlepšená interakce se zákazníky a efektivnější provoz, jsou lákavé, realizace takových projektů vyžaduje pečlivou přípravu a strategický přístup.

V návaznosti na zmíněné studie a problémy je důležité zdůraznit, že klíčem k úspěchu integrace umělé inteligence do CRM je překonání těchto technických a organizačních bariér. Firmy musejí investovat do rozvoje nebo získání potřebné technologické infrastruktury a zajištění dostupnosti kvalitních dat, která jsou základem pro efektivní využití AI. To představuje nejen shromažďování dat, ale také jejich čištění, kategorizaci a analýzu, aby byla zajištěna jejich relevance a přesnost.

Dalším klíčovým faktorem je integrace AI do stávajících systémů CRM bez zbytečného narušení. To vyžaduje, aby řešení AI byla navržena tak, aby byla plně kompatibilní s existujícími databázemi a platformami CRM, a umožnila bezproblémovou synchronizaci dat. Firmy by měly usilovat o vytvoření flexibilní a škálovatelné

architektury, která umožní rychlou adaptaci na měnící se požadavky trhu a potřeby zákazníků (Keegan, 2022).

Kromě technických aspektů je důležité nezanedbat i lidský faktor. Úspěšná integrace AI do CRM vyžaduje zaškolení zaměstnanců, aby mohli efektivně využívat nové nástroje a technologie. Rovněž je nutné zajistit, aby AI nebyla vnímána jako náhrada lidského úsudku a interakce, ale jako nástroj, který tyto aspekty doplňuje a rozšiřuje.

V konečném důsledku je důležité pochopit, že úspěch integrace AI do systémů CRM nespočívá pouze ve využívání pokročilých technologií, ale také ve schopnosti tyto technologie smysluplně začlenit do obchodních procesů tak, aby přinášely skutečnou hodnotu jak firmám, tak jejich zákazníkům. Strategické plánování, pečlivá implementace a neustálé hodnocení a optimalizace procesů jsou proto nezbytné kroky na cestě k dosažení tohoto cíle.

Akademická obec tvrdí, že umělá inteligence představuje další evoluční skok směrem k pokročilejšímu a efektivnějšímu systému CRM (Mishra a Mukherjee, 2019) a že systémy AI-CRM, poháněné ML, DL a dalšími nadcházejícími technologiemi umožní manažerům zlepšit prediktivní návrhy a návrhy v reálném čase s využitím rozsáhlého sberu dat (Mishra a Mukherjee, 2019). Jak upozorňují Ledro et al. (2022), v poslední době došlo k posunu v přístupu akademické obce k integraci AI-CRM. Namísto podrobného zkoumání konkrétních technologických aplikací vědci nyní praktikují strategickou perspektivu se širším záběrem. Tento posun podněcuje diskuse o výhodách a výzvách integrace AI do procesů CRM, jakož i o nezbytných organizačních, kulturních a strategických změnách. Tento posun v perspektivě, od technologického ke strategickému přístupu, odráží rostoucí zájem o přehodnocení vztahu mezi technologií a strategií CRM.

Výzkumy a studie v akademické obci jasně naznačují, že umělá inteligence představuje důležitý prvek v evoluci systémů CRM, směrem k pokročilejším a efektivnějším řešením. Tento vývoj, podpořený technologiemi jako jsou strojové učení (ML) a hluboké učení (DL), nabízí manažerům nové možnosti pro vylepšení prediktivních modelů a návrhů v reálném čase, což se opírá o rozsáhlé sběry dat. Poslední dobou je zřetelný posun v akademickém přístupu k integraci AI do systémů CRM. Oproti minulosti, kdy se vědecké studie soustředily především na specifické technologické aplikace, v současné době převládá tendence k širšímu strategickému pohledu na tuto problematiku.

Tento strategický přístup neznamena jen zkoumání možností, jak technologie AI mohou přinést přidanou hodnotu do stávajících systémů CRM, ale také pochopení

nezbytných organizačních, kulturních a strategických změn, které jsou pro úspěšnou integraci AI nezbytné. Je zřejmé, že integrace AI do CRM přesahuje čistě technologickou výzvu a stává se klíčovou strategickou iniciativou, která může významně ovlivnit celkovou strategii podnikání a zákaznické vztahy.

V tomto kontextu se zdá být důležité, aby se podniky zaměřily nejen na výběr a implementaci nejnovějších AI technologií, ale také aby pečlivě zvážily, jak tyto technologie nejlépe zapadají do jejich celkové strategie CRM a jak mohou tyto nástroje nejefektivněji využít pro zlepšení vztahů se zákazníky. To zahrnuje také nutnost přizpůsobit organizační struktury a kulturu podniků tak, aby bylo možno podpořit inovativní přístupy a maximálně využít potenciál, který AI nabízí (Ledro 2023).

Klíčem k úspěchu patrně je najít rovnováhu mezi technologickým potenciálem AI a strategickými cíli CRM. Tím, že budou podniky schopny integraci AI vnímat jako součást širší strategické vize, mohou lépe využít její potenciál pro posílení vztahů se zákazníky, zlepšení procesů a získání konkurenční výhody.

3.1.5 Vymezení pojmu nemocniční informační systém

Nemocniční informační systém se často označuje zkratkou NIS, která zde bude nadále používána (anglicky HIS – Hospital Information System). Jde o soubor aplikací, technologií a služeb, dodávaných určitou firmou nebo firmami jako komplexní řešení systémů, které pracují s životně důležitými daty a které tato data bezpečně uchovávají. To představuje provozní řešení pro zdravotnická zařízení a podporuje celý proces léčby pacienta od přijetí, stanovení diagnózy, terapie, propuštění z nemocnice a vyúčtování poskytnuté zdravotní péče (Materová, 2013).

Nemocniční informační systémy (NIS) se staly základním kamenem moderního zdravotnictví. Tato systémová řešení, která zahrnují soubor aplikací, technologií a služeb, jsou klíčová pro zajištění plynulého a efektivního provozu zdravotnických zařízení. Poskytují nejen bezpečné uchování životně důležitých dat, ale také podporují celý proces léčby pacienta od přijetí přes stanovení diagnózy a terapii až po propuštění a vyúčtování poskytnuté péče. Tato integrace procesů přináší mnoho výhod, včetně zvýšené efektivity práce zdravotnického personálu, lepší správy záznamů o pacientech a zlepšení kvality poskytovaných zdravotnických služeb.

Implementace NIS však přináší i výzvy, a jednou z hlavních je zajištění bezpečnosti a ochrany dat, což vyžaduje neustálou pozornost vzhledem k rostoucím kybernetickým

hrozbám. Dalším důležitým aspektem je integrace s dalšími systémy a technologiemi, která vyžaduje flexibilitu a otevřenost NIS. To umožňuje zdravotnickým zařízením využívat inovace v medicínské technologii a zpracování dat, čímž se dále zvyšuje kvalita péče o pacienty.

Navíc, s postupující digitalizací zdravotnictví, roste důraz na uživatelskou přívětivost systémů. Zdravotnický personál, který se každodenně setkává s velkým množstvím informací, potřebuje nástroje, které jsou intuitivní a snadno použitelné, aby mohl efektivně pečovat o pacienty bez zbytečné administrativní zátěže. Toto vede k neustálému vývoji uživatelských rozhraní a funkcionalit NIS, které jsou přizpůsobeny potřebám koncových uživatelů.

Nemocniční informační systém přistupuje k řešení zdravotnické informatiky komplexně. Respektuje pracovní návyky v lékařské praxi a minimalizuje zátěž uživatele administrativními a jinými vedlejšími činnostmi. Respektuje standardy MZČR, VZP a je flexibilní v měnících se podmínkách lékařské praxe, legislativy včetně harmonizace s normami EU. Snahou NIS je ergonomické ovládání (snadná ovladatelnost i pro nepočítačovou obsluhu), snadná orientace v prostředí systému (jednotný vzhled uživatelských obrazovek, obdobný způsob ovládání ve všech modulech), odolnost vůči neodborným uživatelským zásahům (Materová, 2013).

Pokračování tohoto procesu znamená, že se zdravotnické informační systémy neustále vyvíjejí, aby držely krok s rychlým tempem změn ve zdravotnictví. To zahrnuje adaptaci na nové lékařské protokoly, změny v legislativě a integraci s novými technologiemi a zdravotnickým vybavením. V tomto kontextu je důležité, aby se při plánování a vývoji NIS braly v úvahu nejen současné, ale i budoucí potřeby zdravotnických zařízení, pouze tak mohou nemocniční informační systémy efektivně sloužit svému účelu a přispívat k lepší péči o pacienty a větší efektivitě zdravotnických zařízení.

Cílem medicínské části NIS je pečovat o pacienta, zvyšovat kvalitu poskytované zdravotní péče, zvyšovat produktivitu zdravotnického personálu a snižovat administrativní náklady na poskytování a financování zdravotní péče. Dalším cílem je podporovat klinické výzkumy a zkoumání zdravotních služeb, umožnit zapojení případných novinek v lékařské technologii. Strategií poskytování zdravotní péče je její správné financování a záruka, že údaje o pacientovi zůstanou za všech okolností důvěrné (Materová, 2013).

Zajímavým aspektem je také snaha o snížení administrativních nákladů. Integrací inteligentních systémů a automatizace rutinních procesů může NIS výrazně snížit čas strávený administrativou, což umožňuje zdravotnickému personálu více se soustředit na přímou péči o pacienty. To vede k efektivnějšímu využití času a zdrojů a zlepšuje celkovou produktivitu zařízení.

Vzhledem k těmto cílům a výzvám je jasné, že vývoj a implementace NIS musejí být prováděny s ohledem na nejnovější technologické trendy a potřeby zdravotnických pracovníků. Pouze tak lze zajistit, že NIS bude skutečně sloužit svému účelu a přispívat k vysoké kvalitě a efektivitě poskytované zdravotní péče.

3.1.6 CRM systémy v medicíně

Řízení vztahů se zákazníky je inovativní technologie, jejímž cílem je zvýšit spokojenost, loajalitu a ziskovost zákazníků prostřednictvím získávání, rozvíjení a udržování efektivních vztahů a interakcí se zákazníky (Hung, 2010). Z pohledu zdravotnictví definovali Benz a Paddison (2004) CRM jako přístup k poznávání pacientů za účelem vhodné komunikace a budování dobrých vztahů s cílem poskytovat včasné informace, přičemž výsledky pacientů jsou sledovány za účelem provedení nezbytných úprav. V této práci je CRM ve zdravotnickém prostředí vyváženě definováno jako manažerský přístup a aplikaci zdravotnických informačních technologií (HIT), která podporuje koncept péče zaměřené na pacienta. To umožňuje nemocnicím více se zaměřit na pacienty a uspokojit jejich potřeby a očekávání, zvýšit loajalitu, kvalitu služeb (SQ) a vybudovat dlouhodobý vztah. Yina (2010) poukázal na to, že efektivní CRM vyžaduje komplexní sběr dat od hospitalizovaných i ambulantních pacientů prostřednictvím multimediální platformy a také integraci systému CRMS s různými klinickými sítěmi, jako jsou nemocniční informační systémy (HIS), elektronické zdravotní záznamy (EHR), laboratorní informační systémy (LIS), webové platformy nemocnic, call centra a systémy založené na SMS. To vyžaduje, aby poskytovatelé zdravotní péče disponovali IT zdroji, jako je hardware, software a infrastruktura, aby mohli zavést CRMS a efektivněji ukládat záznamy o pacientech. Poskytovatelé zdravotní péče mohou být schopni dosáhnout loajality pacientů, pokud zohlední dva klíčové faktory: (i) počet pacientů a (ii) zisk pacientů. Čím více loajálních pacientů nemocnice má, tím méně investic musí vynaložit, tím větší zisk může mít a naopak (Lau, 2010).

Pokud jde o zdravotnická zařízení, tyto systémy plní funkci automatizace sběru informací, umožňují kombinovat údaje z několika zdrojů a zjednodušují úkoly administrativních pracovníků kliniky. Výsledkem je, že zdravotnická zařízení efektivněji slouží svým klientům, zlepšují úroveň služeb a zjednodušují řídicí postupy při své každodenní činnosti.

Moderní zdravotnické společnosti ocenily možnosti CRM pro zdravotnická centra a aktivně je využívají k organizaci interakce s pacienty. Evropské kliniky si již dávno neuvědomují, jak se bez této automatizované složky při práci s klienty obejdou.

Zavedení CRM systémů jako povinné součásti úspěšného fungování zdravotnické organizace napomáhá nárůst počtu společností poskytujících podobné služby, a tedy i růst konkurence. Zákazníci dávají přednost těm klinikám, kde je úroveň služeb vyšší, a CRM pro zdravotnická střediska se zaměřují na zlepšování služeb.

Taková vlastnost CRM systémů, jako je flexibilita, je čini vhodnými pro podniky jakéhokoli rozsahu, od malé soukromé zubní ordinace až po rozsáhlou síť. Stejně úspěšně je mohou využívat laboratoře, centra psychologické pomoci i různé kliniky, protože úkoly interakce s pacienty jsou téměř stejné.

Systémy CRM v lékařství a zdravotnictví pomáhají řídit vztahy s pacienty, zlepšují kvalitu péče a usnadňují administrativní procesy. Specifické aspekty systémů CRM v kontextu medicíny (Perna, 2014):

Správa pacientů - systémy CRM mohou pomoci sledovat historii interakcí s pacienty, záznamy o schůzkách, lékařské předpisy atd.

Plánování schůzek - systémy mohou poskytovat kalendáře pro plánování lékařských schůzek, zasílání upomínek na nadcházející schůzky a oznámení o změnách.

Analýza dat - analýza CRM může pomoci identifikovat vzorce nemocnosti, předvídat poptávku po určitých službách a optimalizovat procesy na klinice.

Spolupráce při léčbě - systémy CRM mohou podporovat komunikaci mezi lékaři a umožnit jim sdílet údaje o pacientech a koordinovat léčebné plány.

Správa stížností a zpětné vazby - systémy CRM pomáhají řešit stížnosti pacientů, poskytují kanály pro zpětnou vazbu a účinně reagují na dotazy.

Používání systémů CRM v medicíně přispívá k lepší a personalizovanější péči o pacienty, zefektivňuje procesy a zlepšuje celkovou výkonnost klinik a zdravotnických organizací.

CRM v medicíně je mocný nástroj, který může výrazně zlepšit řízení vztahů s pacienty, optimalizovat procesy a zlepšit kvalitu zdravotní péče. Následuje bližší pohled na výhody, nevýhody, škálovatelnost, integraci s AI a dalšími systémy a funkce takového systému CRM.

Výhody systémů CRM v oblasti zdravotnictví zahrnují efektivnější péči o pacienty díky možnosti vést detailní záznamy o pacientech, poskytování personalizovaných služeb, zlepšení plánování schůzek a automatizovanému zasílání upomínek. Automatizaci rutinních úkolů, jako je správa zdravotnických záznamů, plánování schůzek a zasílání upomínek, což umožňuje lékařům a zdravotnickému personálu soustředit se na přímou péči o pacienty. Analýzu údajů o pacientech a prognózování, umožňující identifikovat trendy v chování a potřebách pacientů, předvídat poptávku po zdravotnických službách a efektivněji plánovat využití zdrojů. Podporu komunikace a spolupráce mezi lékaři a zdravotnickým personálem prostřednictvím sdílení informací o pacientech a koordinace péče. Efektivnější správu stížností a zpětné vazby od pacientů, poskytování kanálů pro zpětnou vazbu a rychlé reagování na dotazy (Anshari, 2012).

Na druhou stranu, **implementace CRM v medicíně přináší i určité nevýhody**, například složitost implementace vyžaduje značný čas a zdroje na školení personálu a přizpůsobení systému specifickým potřebám zdravotnického zařízení. Vysoké požadavky na ochranu osobních údajů pacientů vyžadují implementaci pokročilých bezpečnostních opatření. Potřeba revize a změny pracovních postupů, která může vyvolat odpor u zaměstnanců, zvyklých na tradiční způsoby práce.

Možnosti rozšíření a integrace obsahují AI, protože systém CRM lze integrovat s umělou inteligencí, která analyzuje data, předpovídá vzorce nemocnosti, automaticky zpracovává dotazy a dokonce poskytuje doporučení k léčbě. Systém CRM lze integrovat s platformami telemedicíny pro konzultace na dálku, sdílení lékařských informací a sledování pacientů. Systém CRM lze integrovat s lékařským vybavením pro sběr dat pro přesnější sledování pacientů (Mishra, 2019).

3.1.7 Implementace systému

Následuje pokus o charakteristiku standardního modelu práce v nemocnici, dále budou vytvořena pomocná schémata charakterizující procesy a následně bude vytvořen předpoklad týkající se toho, které prvky systému jsou pro jednoho lékaře nejjednodušší a

nejpotřebnější, neboť implementace rozsáhlého univerzálního návrhu podniku by přesáhla rozsah této práce.

Podnikové procesy zdravotnického střediska jako takového zahrnují následující kroky a činnosti (Materová, 2013).

1. Příjem pacientů - prvním krokem v provozu zdravotnického střediska je příjem pacientů. Tento proces začíná, když pacient přijde do zdravotnického střediska, objedná se a vyplní dotazník. Poté je přijat lékařem, který dotazník zkontroluje. Pokud se v něm objeví nějaké zvláštnosti či nejasnosti, lékař se na ně zeptá.

2. Lékařské vyšetření - v rámci lékařského vyšetření lékař pacienta vyšetří a stanoví diagnózu. V závislosti na domluveném termínu může tento proces zahrnovat různé typy vyšetření: vyšetření krve a moči, rentgen, ultrazvuk, CT a další.

3. Předepsání léčby - hlavním úkolem lékařského střediska je předepsat nejlepší možnou léčbu. Lékař vybere léčebný postup a informuje pacienta o jeho délce a zvláštnostech. Pacient si poté nechá předepsat léky a obdrží je.

4. Doprovod pacienta - zdravotní střediska poskytují také služby doprovodu pacienta během léčby. Některá zdravotnická střediska mohou poskytovat služby sledování užívaných léků, monitorování pacientovy stravy a fyzické aktivity.

6. Sběr a analýza údajů - součástí procesu provozování zdravotního střediska je sběr a analýza údajů o pacientech. To může zahrnovat zaznamenávání a uchovávání zdravotnických údajů a také vytváření statistických informací. To je důležité pro plánování, podávání zpráv a zlepšování kvality služeb, které zdravotnické středisko poskytuje.

Systematizace obchodních procesů ve zdravotnickém středisku má řadu důležitých důvodů. Pomáhá zlepšovat kvalitu zdravotnických služeb tím, že optimalizuje pracovní procesy, zjednodušuje tok dokumentů a zvyšuje efektivitu zaměstnanců. Systematizace zdravotnických procesů rovněž pomáhá optimalizovat náklady a využití zdrojů, což je důležitý faktor pro ekonomickou efektivitu zdravotnického střediska. Dále systematizace procesů umožňuje účinnější kontrolu výkonu práce a sledování její kvality, což je důležité pro zlepšení image zdravotnického střediska a spokojenosti zákazníků.

Důležitým opatřením je také optimalizace již existujících podnikových procesů ve zdravotnickém středisku, protože umožňuje snížit náklady a zlepšit kvalitu služeb. Například optimalizace procesů objednávání a poskytování lékařských služeb může zkrátit čekací dobu pro klienty, což zvýší spokojenost klientů a jejich opakované návštěvy. Také

zefektivnění procesů zaznamenávání a uchovávání lékařských informací může zlepšit celkovou efektivitu zdravotnického střediska a snížit náklady na papírování.

Sledování vývoje obchodních procesů je také velmi důležité, protože pomáhá zdravotnickému středisku udržet si konkurenceschopnost a přizpůsobit se měnícím se podmínkám na trhu. Například nové technologie a organizační přístupy mohou výrazně zlepšit kvalitu zdravotnických služeb a zjednodušit práci personálu, jejich implementace do stávajících obchodních procesů může vést k výraznému zvýšení ekonomické efektivity zdravotnického střediska a zlepšení jeho image. Zavedení systému CRM může výrazně zlepšit obchodní procesy ve zdravotnické organizaci (Ledro, 2023).

Možné přínosy:

Platforma, která má být vyvinuta, by měla mít potenciál škálování, aby umožnila ukládání, zpracování a analýzu velkých objemů dat týkajících se pacientů, léčby, plateb za služby a dalších aspektů činnosti kliniky. To může pomoci identifikovat trendy a vzorce chování pacientů, rozpoznat faktory ovlivňující jejich zdraví a určit účinnost různých léčebných postupů (Anshari, 2012).

Architektura mikroslužeb naznačí, že systém by mohl být v budoucnu rozšířen tak, aby vytvářel automatizované zprávy, informační panely a analytické nástroje, které zpracovávají data a poskytují informace pro rozhodování. To by mohlo pomoci týmu vedení kliniky rychle přistupovat ke klíčovým ukazatelům výkonnosti a rozhodovat se na základě přesnějších a relevantnějších informací.

Systém by měl mít možnosti vzájemné integrace s dalšími službami a servisí, které mohou po propojení vytvářet personalizovaná doporučení pro pacienty s ohledem na jejich léčebnou historii, potřeby a zájmy. Kromě toho může systém pomoci automatizovat oznámení o schůzkách a platbách, cožlepší úroveň služeb a pohodlí pacientů.

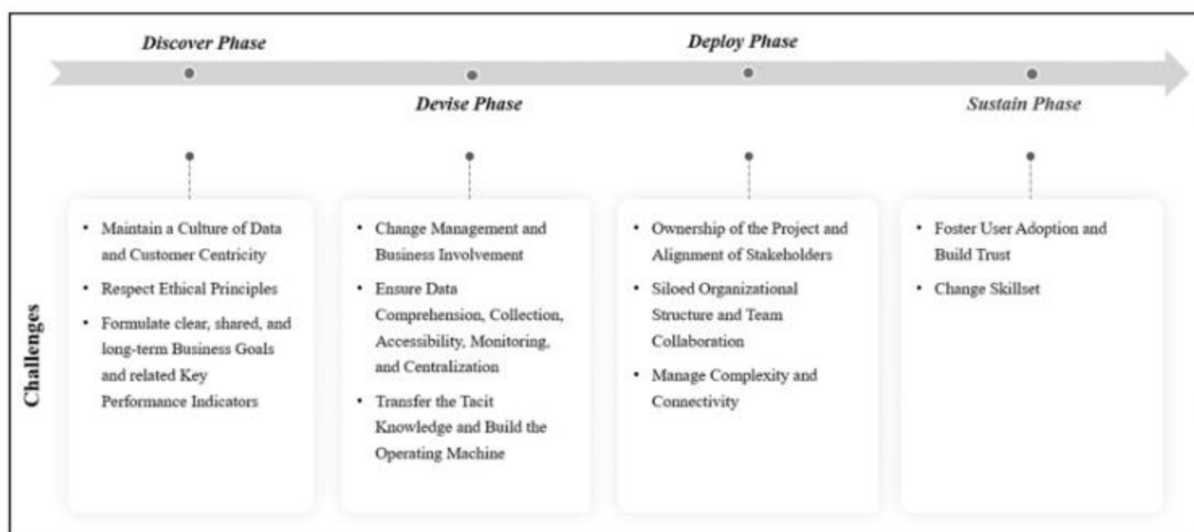
Systém bude muset umožnit automatizaci mnoha procesů, jako je rezervace termínů, kontrola zásob léků a spotřebního materiálu. Tím se zkrátí čas strávený rutinní prací a sníží pravděpodobnost chyb.

Připojení dalších služeb a rozšíření by potenciálně mohlo poskytnout různé nástroje pro komunikaci s pacienty a personálem, jako jsou chaty, videohovory a e-maily. To by zkrátilo čas strávený komunikací a umožnilo rychlejší a efektivnější interakce.

3.1.8 Integrace umělé inteligence do systémů CRM

Integrace umělé inteligence do systémů CRM představuje nejen technologickou výzvu, ale zahrnuje také širokou škálu strategických, etických a organizačních otázek. Tyto aspekty jsou zásadní pro úspěšnou implementaci a dlouhodobé využití AI v rámci řízení vztahů se zákazníky.

Obrázek 1. Integrace umělé inteligence do systémů CRM



Zdroj: Ledro, 2023

Obecně se předpokládá, že přechod do další fáze bez splnění požadavků v aktuální fázi zvyšuje riziko neúspěchu. Prezentované poznatky připraví cestu k hlubšímu pochopení integrace AI do prostředí CRM a konkrétních činností a schopností potřebných k překonání překážek, které jsou vyvolány specifickými vlastnostmi integrace AI-CRM (Ledro, 2023).

V rámci této diplomové práce nejsou podrobně řešeny technické a teoretické aspekty implementace umělé inteligence. Místo toho se práce zaměřuje na zkoumání důležitých prvků, které budou v nadcházejících 10 až 20 letech představovat výzvy pro lidstvo a průmysl v kontextu integrace AI, zejména v systémech pro CRM. Toto zaměření otevírá diskusi o tom, jak přijetí AI ovlivňuje praxi podniků a reakce lidí na tato nová technologická řešení (Anshari, 2012).

Jedním z klíčových aspektů, na které se práce snaží upozornit, je, jak vývoj a nasazení AI v CRM systémech vyvolávají potřebu pro organizační, kulturní a strategické změny. Zvláštní důraz je kladen na význam porozumění emocím zákazníků, což

představuje unikátní výzvu, neboť tradiční aplikace AI nejsou typicky navrženy tak, aby tyto typy dat efektivně zaznamenávaly a interpretovaly.

Dále práce zdůrazňuje význam sjednocení všech zainteresovaných stran a zajištění vlastnictví projektů v rámci CRM systémů, které zahrnují různá oddělení, jako jsou marketing, prodej a zákaznický servis. Je zdůrazněna nutnost jasné integrační strategie, která umožňuje bezproblémovou komunikaci a výměnu dat mezi AI-CRM systémy a stávajícími platformami.

Tato práce rovněž upozorňuje na důležitost nalezení rovnováhy mezi úlohami a odpovědnostmi lidí a AI v rámci interakcí CRM. AI může sice automatizovat úkoly a uvolnit čas lidských pracovníků pro jinou práci, ale nemůže nahradit lidský úsudek a empatii.

Úspěšná integrace AI do systémů CRM nevyžaduje pouze technologické inovace, ale také hluboké pochopení a adaptaci na organizační, strategickou a etickou problematiku. Toto poznání nejenže zvyšuje informační hodnotu diplomové práce, ale také ukazuje na potenciální směry pro budoucí výzkum, který by mohl být tématem doktorské práce, především v oblasti etiky, spolupráce člověka s AI a dlouhodobého dopadu AI na řízení vztahů se zákazníky (pacienty).

3.2 Webové technologie

V této fázi je vhodné zdůvodnit používání webových technologií pro co nejefektivnější a hlavně pohodlný způsob řešení svých úkolů. Vývoj systému Web-CRM. Toto téma zahrnuje pojem web.

Web (nebo přesněji webová aplikace) je aplikační software, který běží na webovém serveru, na rozdíl od počítačových programů, které běží lokálně v operačním systému (OS) zařízení. K webovým aplikacím přistupuje uživatel prostřednictvím webového prohlížeče s aktivním síťovým připojením. Tyto aplikace jsou naprogramovány pomocí struktury modelované jako klient-server: uživateli (klientovi) jsou poskytovány služby prostřednictvím externího serveru hostovaného třetí stranou. Příklady běžně používaných webových aplikací jsou: webmail, online maloobchod, online bankovníctví a online aukce (Lee, 2004).

Webová technologie jako nedílná součást internetu zcela změnila způsob přemýšlení o práci s informacemi a s počítači obecně. Ukázalo se, že tradiční parametry vývoje výpočetní techniky, tedy výkon, šířka pásma, kapacita paměti, nezohledňovaly hlavní

„úzké hrdlo“ systému - rozhraní s člověkem. Zastaralý mechanismus interakce člověka s informačním systémem bránil zavádění nových technologií a snižoval přínosy jejich používání, teprve když se rozhraní mezi člověkem a počítačem zjednodušilo na přirozené vnímání běžného člověka, následoval nebyvalý rozmach zájmu o možnosti výpočetní techniky.

Ve vývoji webových technologií lze rozlišit následující etapy:

- Web 1.0
- Web 2.0
- Web 3.0
- potenciálně Web 4.0

Hlavní rozdíl mezi Webem 1.0, Webem 2.0 a Webem 3.0 spočívá v tom, že Web 1.0 je chápán jako webový cíl určený pouze pro čtení a kreativitu tvůrců obsahu. Web 2.0 je zaměřen na kreativitu uživatelů a producentů obsahu, zatímco Web 3.0 je zaměřen na propojené datové soubory. Níže je uvedeno velmi málo srovnávacích rozdílů mezi Webem 1.0, Webem 2.0 a Webem 3.0 (Choudhury, 2014). Tabulka 1 ukazuje hlavní rozdíly mezi jednotlivými generacemi webových technologií.

Tabulka 1 Rozdíly mezi jednotlivými generacemi webových technologií

WEB 1.0	WEB 2.0	WEB 3.0
1996 - 2004	2004-2016	2016+
The Hypertext Web	The Social Web	The Semantic Web
Tim Berners Lee	Tim O'Reilly, Dale Dougherty	Tim Berners Lee
Read Only	Read and Write Web	Executable Web
Millions of User	Billions of User	Trillions+ of Users
Echo System	Participation and Interaction	Understanding self
One Directional	Bi-Directional	Multi-user Virtual environment
Companies Publish Content	People Publish Content	People build application through which people interact and publish content.
Static content	Dynamic content	Web 3.0 is curiously undefined. AI and 3D, The

		web learning
Personal Websites	Blog and Social Profile	Semi Blog, Haystack
Message Board	Community portals	Semantic Forums
Buddy List, Address Book	Online Social networks	Semantic Social Information

Zdroj: McNaughton, 2010

Hlavní výhody webových aplikací:

1. Běží na více platformách bez ohledu na operační systém nebo zařízení.
2. Všichni uživatelé mají přístup ke stejné verzi projektu, což eliminuje jakékoli problémy s kompatibilitou.
3. Neinstaluje se na pevný disk, čímž se eliminují prostorová omezení.
4. Serverová část aplikace je na vaší straně, takže uživatelé nemohou porušit licenční omezení, což zabraňuje softwarovému pirátství.
5. Snižuje náklady podniku i koncového uživatele, protože je zapotřebí méně podpory a údržby a snižují se nároky koncového uživatele na počítač nebo chytrý telefon.

3.2.1 Obecné informace o webových frameworkcích

V počítačovém programování je softwarový framework abstrakce, ve které lze software poskytující běžné funkce selektivně upravovat dalším kódem napsaným uživatelem, a poskytovat tak software specifický pro danou aplikaci. Poskytuje standardní způsob vytváření a nasazování aplikací a je obecným, opakovaně použitelným softwarovým prostředím, které poskytuje specifické funkce jako součást větší softwarové platformy usnadňující vývoj softwarových aplikací, produktů a řešení. Softwarové platformy mohou zahrnovat pomocné programy, překladače, knihovny kódu, sady nástrojů a rozhraní pro programování aplikací (API), které sdružují všechny komponenty umožňující vývoj projektu nebo systému (McNaughton, 2010).

Na počátku éry vývoje webových aplikací se všechny aplikace psaly ručně a pouze vývojář aplikace mohl aplikaci upravit nebo nasadit. Webové frameworky umožnily vyhnout se tomu. Od roku 1995 byly všechny problémy spojené se změnou struktury aplikace do jisté míry odstraněny vznikem společného přístupu k vývoji webových aplikací, v této době se objevily jazyky pro tvorbu webu. Jejich rozmanitost umožňuje vybrat si vhodný jazyk jak pro statické, tak pro dynamické stránky a v závislosti na daném

úkolu je možné zvolit jeden framework, který pokryje všechny potřeby, nebo jich několik zkombinovat.

Typy webových frameworků

Webové frameworky mají dvě hlavní funkce: práci na straně serveru (backend) a práci na straně klienta (frontend). Frontendové frameworky se týkají přední části aplikace. Zjednodušeně řečeno jsou zodpovědné za vzhled aplikace. Backend je zodpovědný za obsah aplikace.

Webové frameworky na straně serveru

Pravidla a architektura těchto webových frameworků neumožňují vytvořit webovou aplikaci s bohatým rozhraním. Jejich funkčnost je omezená, ale přesto lze použít k vytvoření jednoduchých stránek a formulářů. Mohou také generovat výstupní data a odpovídat za bezpečnost v případě útoků a to vše může zjednodušit proces vývoje. Webové frameworky na straně serveru jsou většinou odpovědné za samostatné, ale kritické části aplikace, bez kterých aplikace nebude správně fungovat. Několik nejoblíbenějších frameworků na straně serveru a jazyky:

- Django - Python,
- Express.js - JavaScript,
- Ruby on Rails - Ruby.

Webové frameworky na straně klienta

Na rozdíl od webových frameworků na straně serveru nemají webové frameworky na straně klienta nic společného s logikou aplikace. Tento typ webových rámců běží v prohlížeči. Lze je použít k vylepšení a implementaci nových uživatelských rozhraní. Frontendové frameworky umožňují vytvářet různé animace a jednostránkové aplikace. Všechny frameworky na straně klienta se liší funkčností a použitím. Zde jsou některé z nich:

- Angular,
- Vue.js.

Všechny tyto webové frameworky používají JavaScript.

Funkce webových frameworků

Webové ukládání do mezipaměti.

Ukládání do mezipaměti jednoduše pomáhá ukládat různé dokumenty a zabraňuje přetížení serveru, uživatelé ji mohou za určitých podmínek používat na různých systémech.

Funguje také na straně serveru. Cachovaných webových stránek si lze všimnout například na stránce s výsledky vyhledávání Google.

Scaffolding

Jedná se o další technologii podporovanou některými frameworky MVC. Framework může automaticky generovat typické části aplikace nebo dokonce celou strukturu projektu (pokud se jedná o inicializaci), což výrazně zvyšuje rychlost vývoje a standardizuje základnu kódu.

System webových šablon

System webových šablon je soubor různých metodik a softwaru implementovaných pro vytváření a nasazování webových stránek. Ke zpracování webových šablon se používají šablonovací nástroje. Jsou základním nástrojem zodpovědným za publikování webových stránek.

Mapování adres URL

Pokud je třeba zjednodušit indexaci webové stránky vyhledávači a zároveň vytvořit atraktivní název webové stránky, pak se využívá tato funkce frameworků. Také mapování URL může usnadnit přístup ke stránkám webu.

Aplikace

Webové frameworky podporují mnoho typů webových aplikací. Používají se především k vytváření aplikací, jako jsou blogy, fóra, CMS atd.

Všechny tyto funkce jsou vlastní všem webovým rámcům. Při tak široké nabídce se však vývojář paradoxně ztrácí a nemůže si nic vybrat, proto je třeba mít kritéria, podle kterých lze vybrat ten nejlepší nástroj usnadňující vývoj. Výběr může záviset například na preferovaném programovacím jazyce a kromě toho je třeba věnovat pozornost také rozsahu schopností sady nástrojů daného frameworky a k preferencím je třeba poznamenat, že mohou sloužit pozitivně i negativně. Samozřejmě je lepší používat frameworky, které se snáze učí.

3.2.2 Vizualizace informací

Dá se říci, že vizualizace informací je činnost zaměřená na hledání a zavádění forem a metod vizuální reprezentace abstraktních dat s cílem usnadnit jejich vnímání člověkem.

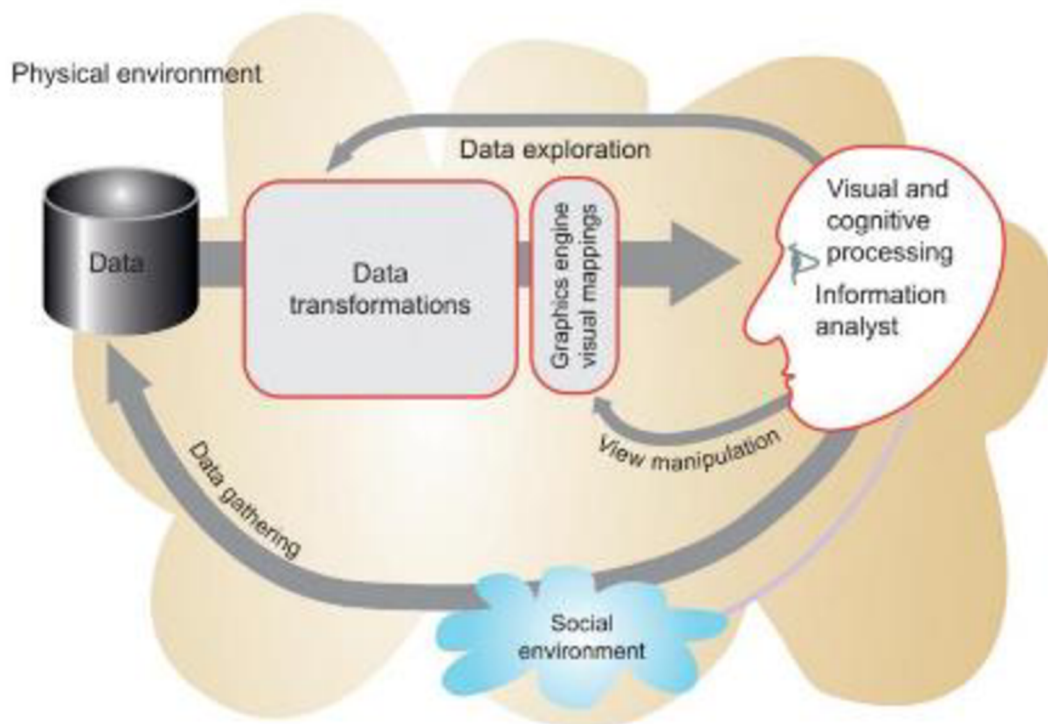
Vizualizaci informací lze definovat jako využití počítačem podporovaných interaktivních vizuálních reprezentací abstraktních dat pro zlepšení poznávání.

Fyzická data, která s největší pravděpodobností leží na nějakém fyzickém médiu, propojená a uspořádaná, jako jsou objektově-relační databáze, procházejí cestou transformace dat a tato transformace obvykle zahrnuje určité programování. Programování je založeno na použití různých algoritmů, podprogramů a zahrnuje:

- extrakci dat,
- posouzení požadovaného formátu,
- určení cílového formátu,
- konverzi extrahovaných dat,
- uložení převedených dat do nového souboru.

Proces vizualizace jako schéma bude vypadat následovně:

Obrázek 2. Proces vizualizace



Zdroj: Card, 1999

3.3 Výběr nástrojů pro implementaci

3.3.1 Django

Je třeba se zaměřit na to, jaký nástroj z celého seznamu, který webový trh nabízí, je třeba použít a také na to, jak jej použít. Pro projekt byl zvolen programovací jazyk Python a framework Django ve verzi 4.* a jazyk Python verze 3.* nejnovější verze.

Django byl původně vytvořen v roce 2005 pro správu zpravodajských webů LJWorld.com, lawrence.com a KUsports.com, které vlastnila společnost The World Company. V roce 2008 byla zveřejněna první verze tohoto frameworku, která se stále vyvíjí. Průběžně se také vydávají nové verze, které framework doplňují a opravují drobné chyby.

Django se také používá jako webová komponenta v různých projektech, například Graphite - systém pro tvorbu grafů a pozorování, FreeNAS - svobodná implementace systému pro ukládání a sdílení souborů a na mnoha webových stránkách, jako je Instagram, Disqus, Mozilla, The Washington Times, Pinterest, YouTube, Google atd.

Hlavní prvky vzoru: **Dispečer URL** - při přijetí požadavku na základě požadované adresy URL určí, který zdroj má požadavek zpracovat. **Zobrazení** přijímá požadavek, zpracovává jej a odesílá odpověď uživateli. Pokud je ke zpracování požadavku třeba přistupovat k modelu a databázi, View s nimi komunikuje. K vytvoření odpovědi může použít šablonu nebo šablony. **Model** charakterizuje data používaná v aplikaci. Jednotlivé třídy obvykle odpovídají tabulkám v databázi. **Šablona** reprezentuje logiku pohledu ve formě generovaného html značení. V MVC této komponentě odpovídá View, tedy pohledy.

Když do aplikace přijde požadavek, dispečer URL určí, na který prostředek se tento požadavek mapuje, a předá tento požadavek vybranému prostředku. Tento prostředek vlastně představuje funkci nebo View, která požadavek přijme a určitým způsobem jej zpracuje. Během zpracování může View přistupovat k modelům a databázi, získávat z ní data nebo naopak data do ní ukládat. Výsledek zpracování dotazu je odeslán zpět a tento výsledek vidí uživatel ve svém prohlížeči. Výsledkem zpracování dotazu je zpravidla vygenerovaný html kód, pro jehož generování se používají šablony.

Důvody, proč je nejlepší volbou Django

Poskytuje rychlý vývojový framework pro vytváření komplexních webových aplikací. Rychlost zajišťuje modulární a jednoduchá filozofie neopakování návrhů a logiky v celé struktuře projektu (známá také jako princip DRY nebo princip Don't Repeat

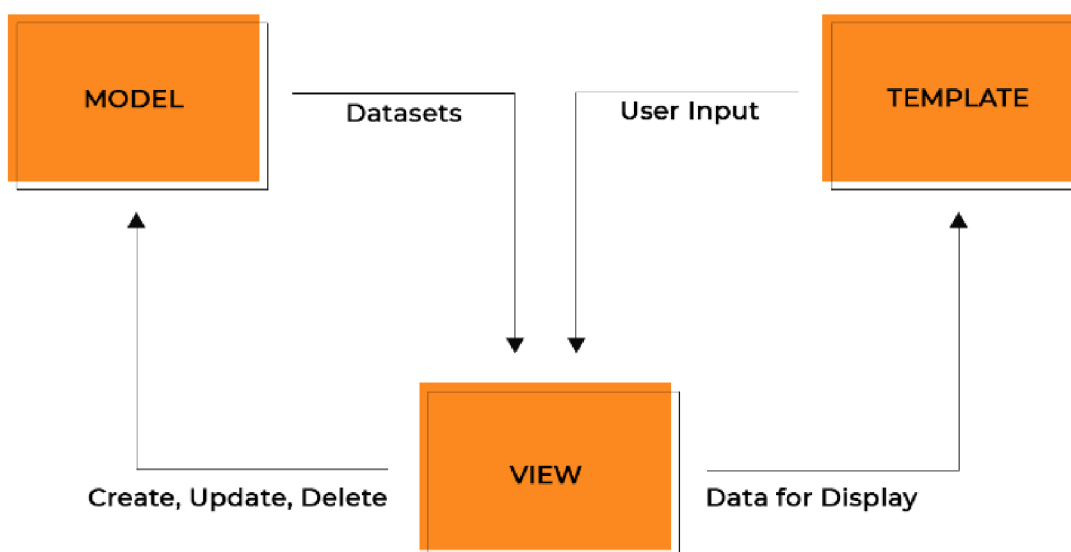
Yourself). Právě tento princip DRY dal vzniknout prosperující komunitě a množství balíčků a dalších frameworků založených na frameworku Django. Po více než 10 letech od prvního vydání je nyní k dispozici plnohodnotný CMS (systém pro správu obsahu), hotová platforma pro elektronické obchodování a více než 3 000 balíčků vytvořených nebo navržených pro práci s platformou Django. Nemluvě o dvou každoročních konferencích v USA a Evropě, na kterých se prezentují inovace Djanga (Rubio, 2017).

Kolem Djanga existuje aktivní komunita. Téměř jakýkoli problém, na který lze při vývoji narazit, je s 99 % pravděpodobností již vyřešen a není nutné ztrácet drahocenný čas.

Využívá architekturu MVC, která vývojáři umožňuje zpracovávat vizuální reprezentaci a obchodní logiku aplikace odděleně. Při práci s Djangem odborníci často používají termín MVT - Model-Visual-View-Template nebo Model-Visual-Template. MVC je schéma rozdělení dat aplikace, uživatelského rozhraní a řídicí logiky do tří samostatných komponent tak, aby bylo možné provádět úpravy každé komponenty nezávisle.

Schematicky jej lze znázornit takto:

Obrázek 3. MVT



Zdroj: vlastní zpracování

Další důležitou vlastností je rozšiřitelnost, Django lze rozšířit pomocí zásuvných modulů. Jedná se o softwarové moduly, které umožňují rychle přidat na web požadovanou funkci, což je při vývoji webové aplikace pro multifunkční zobrazení chemicky orientovaných dokumentů pro snadné čtení určitě potřeba. Kromě toho se při vytváření

aplikace automaticky vygeneruje panel správce Django a vývojář tedy nemusí panel správce vytvářet ručně.

ORM Django implementuje objektově-relační mapování (ORM), které umožňuje aplikaci komunikovat s databázemi (DB). ORM automaticky mapuje data z databáze, například PostgreSQL nebo MySQL, na objekty, které se používají v kódu aplikace. Stručně řečeno, tento framework je skvělý pro vývoj webových aplikací.

3.3.2 Python

Dalším důležitým bodem je, že framework Django je napsán v programovacím jazyce Python, takže jeho struktura odpovídá zvláštnostem tohoto jazyka a Python je podle mnoha názorů nejlepší programovací jazyk.

Letos se podle autoritativního mezinárodního hodnocení poprvé po dvaceti letech podařilo Pythonu obsadit první příčku hodnocení a překonat tak Javu a C. TIOBE každý měsíc zveřejňuje hodnocení oblíbenosti programovacích jazyků.

U výhod a nevýhod Pythonu jako programovacího jazyka je třeba se zastavit podrobněji. Python se stal lídrem mezi dobře zavedenými a optimalizovanými jazyky, včetně jazyků C, C++ a Java, a to z velmi jednoduchých důvodů. Zahrnuje principy filozofie, že složité problémy lze řešit jednoduchými způsoby. Python využil sílu hnutí open source, které mu pomohlo nashromáždit obrovskou základnu uživatelů z téměř všech oblastí života. Filozofii jazyka Python lze definovat jednou větou: Python je víceúčelový, přenosný, vysokoúrovňový objektově orientovaný programovací jazyk, který umožňuje v interaktivním prostředí psát kód minimalistickým způsobem (Nagar, 2017).

Tento jazyk má také dynamické typování: proměnné nemají tzv. předdefinovaný typ. Další z mnoha výhod:

- Bohaté vestavěné typy kolekcí
- Seznamy
- Tuples
- Slovníky (mapy)
- Množiny
- Dobrý příklad skriptovacího jazyka
- Jednoduchý objektový systém
- Iterátory (jako v Javě) a generátory
- Výjimky jako v Javě

- Výkonné indexování (slicing)
- Odsazování místo závorek
- Přenositelnost

Co se týká nevýhod, není jich mnoho. Je to pomalý jazyk, a C implementuje kritické knihovny pro práci s poli a čísly, jako je NUMPY. Tento jazyk je však tak jednoduchý a dobrý, že po zkušenostech s ním chce jen málokdo psát v jiných jazycích.

3.3.3 Výběr nástrojů pro implementaci

Je třeba vyvinout systém CRM pro lékaře a osobní ordinaci pomocí webového frameworku Django. Software by měl poskytovat následující funkce:

1. Přidávání dat do databáze,
2. Čtení dat z databáze,
3. Stylování šablon pomocí CSS,
4. Dynamická úprava dat
5. Zobrazování dat v osobní skříni,
6. Zobrazení údajů o klientech v administrativní části.

Za zmínku také stojí, že pro práci s tímto webovým rámcem není třeba instalovat editory IDE. Stačí nainstalovat interpret jazyka Python a do něj nainstalovat Django. K instalaci Djanga je zapotřebí správce balíčků pip. Pip je systém pro správu balíčků, který slouží k instalaci a správě softwarových balíčků napsaných v jazyce Python.

Od verzi Python 2.7.9 Python obsahuje balíček pip (nebo pip3 pro Python 3) ve výchozím nastavení.

K ukládání dat je obvykle zapotřebí databáze. Databáze je soubor dat uspořádaný podle konceptuální struktury, která charakterizuje vlastnosti těchto dat a vztahy mezi nimi. Existuje velké množství typů databází, které se liší různými kritérii. Klasifikace podle datového modelu:

- Hierarchické,
- Objektové a objektově orientované,
- Objektově-relační,
- Relační,
- Síťové,
- Funkční.

Django bylo navrženo tak, aby běželo pod Apachem s modulem python mod a jako databázi používalo SQLite ale v tomto případě bude využíván PostgreSQL.

V současné době může Django kromě databáze SQLite a PostgreSQL pracovat i s jinými DBMS: MySQL a Oracle.

3.4 Vícekriteriální rozhodování při výběru CRM systému

Rozhodovací proces je nedílnou součástí řízení a jeho hlavní součástí je řešení problémů řízení, například výběr nebo vývoj optimálního CRM systému pro oblast zdravotnictví. Při řešení takových problémů se využívá kvalitativní analýza založená na zkušenostech a znalostech manažera, kdy se rozhodnutí hodnotí podle jejich možných dopadů na systém bez použití numerických výpočtů. Naproti tomu kvantitativní analýza využívá číselné údaje, což umožňuje vytvořit kvantitativní model zkoumaného systému a na jeho základě přijímat informovaná rozhodnutí. Tento model je aproximací skutečného systému a volba správného modelu je rozhodující.

V praxi se často kombinují oba přístupy, tedy kvalitativní a kvantitativní, a výsledkem této integrace je konečné rozhodnutí. Podle Kalčevové (2008) zahrnuje struktura rozhodovacího procesu následující kroky: identifikace problému, analýza a formulace problému, stanovení hodnotících kritérií, vypracování možných řešení, stanovení jejich důsledků, vyhodnocení těchto důsledků a výběr nejvhodnějšího řešení k realizaci, vlastní realizace a následná kontrola účinnosti realizovaného řešení.

Před zahájením hledání možných variant řešení problému je nezbytné definovat cíle, které by tyto varianty měly splnit. Uvědomělé stanovení cílů umožňuje následně posuzovat, do jaké míry jednotlivé varianty tyto cíle naplňují, co se provádí na základě určených kritérií. Tímto způsobem se vyhledává nejvhodnější varianta řešení daného problému.

Vytváření variant je klíčovým krokem v rozhodovacím procesu, protože kvalitní a mnohostranné varianty jsou základem pro informovaná a účinná rozhodnutí. Podle Fotra a kol. (2016) je důležité zdůraznit, že kvalita a kvantita variant spolu souvisejí - větší počet variant nabízí širší škálu možností, což zvyšuje šanci na nalezení optimálního řešení.

Při tvorbě variant se obvykle používají buď intuitivní nebo systematicko-analytické přístupy. V tomto konkrétním případě byla k nalezení vhodných kritérií použita metoda brainstormingu a metoda analogie (systematicko-analytická), přičemž kritéria byla porovnávána s existujícími CRM systémy.

Při výběru a hodnocení různých možností řešení je klíčovým faktorem určení cílů, kterých má být v rozhodovacím procesu dosaženo. Kritéria hodnocení slouží především k určení míry, do jaké různé varianty tyto cíle splňují. Podle Fotra (2016) je důležité zdůraznit multikriteriální povahu rozhodovacích problémů, což znamená, že k posouzení a hodnocení problému a jeho možných řešení je třeba využít více hledisek. Často se stává, že jednotlivá kritéria nemusejí být navzájem aditivní, a to ani v případě, že jsou vyjádřena ve stejných měrných jednotkách (např. rentabilita kapitálu a rentabilita tržeb v procentech). Dalším specifickým aspektem je smíšená povaha kritérií, kdy některá jsou kvalitativní a vyjádřena slovně, zatímco jiná jsou kvantitativní a vyjádřena číselně. Důležitým prvkem hodnocení je počet použitých kritérií - čím více kritérií, tím složitější je proces hodnocení, a totéž platí pro počet hodnocených variant.

Kritéria hodnocení tedy představují různá hlediska, která si rozhodovatel vybírá, aby posoudil, jak výhodné jsou různé možnosti z hlediska dosažení nebo míry dosažení konkrétních cílů řešeného problému. Tato kritéria jsou obvykle odvozena přímo z cílů, které lze vyjádřit jako snahu o maximalizaci (např. obratu, peněžních toků), minimalizaci (např. nákladů, ztrát, času) nebo dosažení konkrétních hodnot stanovených veličin.

Podle Fotra a kol. (2016) musejí kritéria hodnocení splňovat určité základní požadavky, aby bylo možné adekvátně posoudit a vyhodnotit všechny důsledky navrhovaných řešení problému:

- Úplnost - zajišťuje, aby byly zváženy a vyhodnoceny všechny relevantní přímé a nepřímé důsledky každé z navrhovaných alternativ.
- Operativnost - každé kritérium musí být jasné a srozumitelné a musí umožňovat praktické využití při hodnocení.
- Neredundance - zajišťuje, že každý aspekt je zahrnut pouze jednou, aby se zabránilo duplicitě v hodnocení.
- Minimální rozsah - počet kritérií by měl být co nejnižší, bez toho, aby bylo ohroženo jejich komplexní pokrytí posuzovaných aspektů.
- Nezávislost - kritéria by měla být na sobě nezávislá, aby nedocházelo ke zbytečnému překrývání nebo vzájemné závislosti.

Tyto požadavky pomáhají zajistit, aby posouzení možností bylo komplexní, přesné a užitečné pro rozhodovací proces. Možnosti nebo alternativy představují různé způsoby řešení problému, z nichž každá je hodnocena podle následujících kritérií. Často mohou být kritéria v rozporu, například když je jedna alternativa lepší z hlediska jednoho kritéria, ale

horší podle jiného. Cílem je najít alternativu, která nejlépe vyhovuje kombinaci všech relevantních kritérií.

Kritéria se obecně dělí následujícím způsobem:

- Výnosová - vyšší hodnota je upřednostňována před nižší. Například vyšší tržby nebo zisk.
- Nákladová - nižší hodnota je upřednostňována před vyšší. Může jít o snahu minimalizovat náklady nebo čas.
- Smíšená - kombinace výnosových a nákladových kritérií. Často zahrnují hodnocení, které vyvažuje mezi ziskovostí a efektivitou.

Úvodním krokem multikriteriální analýzy je sestavení hodnotící matice M , která je tvořena prvky S_{ij} , kde $i = 1, \dots, I$ varianta a $j = 1, \dots, J$ kritéria. Příklad matice je uveden v následující tabulce.

Tabulka 2 Hodnotící matice

	Kritérium 1	Kritérium 2	Kritérium 3	...	Kritérium j
Varianta 1 (S1)					
Varianta 1 (S2)					
Varianta 1 (S3)					
...					
Varianta i (Sj)					

Zdroj: Kalčevová (2008)

Při analýze a výběru mezi různými variantami řešení je efektivní rozdělit rozhodovací proces do několika fází, a tím postupně redukovat počet variant na ty nejvhodnější. V první fázi se z celkového souboru vyřadí varianty, které nesplňují základní požadavky nebo jsou zjevně horší než ostatní, čehož se dosáhne vyřazením tzv. dominovaných variant.

Dominovaná varianta je taková, pro kterou existuje jiná varianta s lepšími nebo alespoň stejně dobrými hodnotami ve všech kritériích a alespoň v jednom kritériu je tato varianta lepší. Naopak nedominovaná varianta je taková, která nemá lepší ekvivalent - není možné zlepšit žádnou hodnotu kritéria, aniž by se zhoršila jiná. Pokud je v souboru více nedominovaných možností, přechází proces výběru do fáze, kdy je jedna z těchto možností vybrána jako nejvhodnější kompromisní možnost.

Kalčevová (2008) také rozlišuje mezi ideální a bazální variantou, kde:

- Ideální varianta je teoreticky nejlepší možností, které lze dosáhnout. Rozlišuje se relativní ideální varianta (nejlepší podle existujících možností pro dané kritérium) a absolutní ideální varianta (teoreticky nejlepší možná hodnota).

- Bazální varianta představuje opak ideální varianty, tj. nejméně příznivou variantu, které lze teoreticky i prakticky dosáhnout, přičemž se rozlišuje relativní (nejhorší z reálných variant pro dané kritérium) a absolutní bazální varianta (nejhorší možná hodnota).

Rozhodnutím se v tomto kontextu rozumí výběr jedné nebo více přípustných možností z redukovaného souboru a doporučení jejich dalšího řešení nebo realizace.

4 Vlastní práce

4.1 Charakteristika nemocnice

Klinika „ABC“ v Praze byla založena jako klinická základna s kompletními službami. Cílem kliniky je poskytovat kvalitní lékařské služby v oblasti prevence, diagnostiky a léčby různých onemocnění a zlepšovat funkce a kvalitu života pacientů.

Hlavní náplní činnosti kliniky je vědecký přístup, nemocnice poskytuje širokou škálu zdravotnických služeb pacientům s různými potřebami.

Využití moderních technologií a zařízení pro diagnostiku a léčbu zajišťují vysokou přesnost a minimální rizika. Podílí se na vědeckém výzkumu a vývoji nových metod léčby a diagnostiky chorob.

Klinika „ABC“ je moderní klinika, nabízející vysoce kvalifikovanou pomoc při léčbě a diagnostice vad a také nabízí:

- kompletní vyšetření a diagnostiku v den ošetření,
- přijímání pacientů s náhlými zdravotními problémy nebo v důsledku úrazů, které vyžadují okamžitou lékařskou pozornost,
- nabídka služeb specializovaných oddělení, jako jsou kardiologie, neurologie, ortopedie, pediatrie a mnoho dalších, které se zaměřují na specifické zdravotní problémy nebo oblasti těla,
- provádění operací, včetně akutních, plánovaných, a minimálně invazivních zákroků, v různých medicínských oborech,
- zahrnují laboratorní testy, rentgen, magnetickou rezonanci (MRI), počítačovou tomografii (CT) a další obrazové metody používané pro diagnostiku a sledování průběhu léčby,
- služby zaměřené na fyzickou a psychologickou rehabilitaci, které pomáhají pacientům zotavit se po zranění nebo operacích, nebo se vyrovnat s chronickými onemocněními,
- lékařská péče poskytovaná pacientům, kteří nepotřebují hospitalizaci, včetně konzultací, drobných zákroků a sledování,
- poskytování neodkladné lékařské pomoci v případě urgentních zdravotních stavů.

Zaměstnanci pravidelně procházejí kurzy a pokročilým školením, účastní se specializovaných programů a kurzů pro udržení a rozšíření jejich odborných znalostí a dovedností. Lékaři a další zdravotničtí pracovníci se aktivně účastní národních a

mezinárodních lékařských konferencí, kde sdílejí své zkušenosti a seznamují se s nejnovějšími pokroky v medicíně. Všichni zaměstnanci kliniky jsou zaměřeni na empatický a pozorný přístup ke každému pacientovi a snaží se mu poskytnout maximální pohodlí a porozumění ve všech fázích léčby.

4.2 Nedostatky v současném systému poskytování služeb

Po analýze modelů obchodních procesů poskytování zdravotnických služeb na příkladu kliniky lze konstatovat, že je třeba automatizovat evidenci klientů, evidenci a následné poskytování zdravotnických služeb.

Po přezkoumání současného stavu služeb byly identifikovány hlavní nedostatky procesu poskytování služeb kliniky:

- neefektivní správa dokumentů, která nutí každého zaměstnance kliniky vést seznamy pacientů a služeb, které jim byly poskytnuty,
- velké množství změn v průběhu poskytování služeb, které vyžadují neustálé zaznamenávání, brání efektivitě zpracování dat a jejich relevanci,
- neefektivní využívání času zaměstnanců kliniky,
- nízká efektivita používání zdravotnických prostředků,
- chyby v diagnostice,
- při opravě nalezených chyb se mohou objevit další chyby, které spolu souvisejí, a systém může selhat,
- existuje možnost ztráty dat.

V důsledku toho musí klinika implementovat CRM systém, který zefektivní její činnost a zlepší služby zákazníkům.

4.3 Charakteristika stávajících CRM systémů

Mezi pět největších firem na trhu CRM dnes patří gigant cloud computingu Salesforce, Microsoft, Oracle, Adobe Systems a 1C, ale to jsou přední vývojáři univerzálních systémů, nás zajímají konkrétně systémy pro lékařské kliniky. Podívejme se na tyto systémy blíže.

CRM pro zdravotnictví přesahuje rámec správy pacientů a kontaktů. Silný důraz je kladen také na marketing a prodej. Prodejem se v tomto případě rozumí pomoc klinikám nebo jiným zdravotnickým organizacím při hledání nových pacientů a rozšiřování jejich praxe.

V moderních klinikách se automatizace stala samozřejmostí: online plánování, elektronické lékařské záznamy, účetnictví. Vztahy s pacienty jsou však stále zanedbávány. Klinika využívající CRM má možnost vést databázi pacientů, sledovat všechny fáze jejich interakce se zdravotnickým střediskem a může zanechávat štítky a upomínky pro registrátory.

Tento druh CRM se může podílet i na takových věcech, jako je zákaznická podpora. Vytváření online center zdrojů, generování požadavků a sledování problémů, poskytování jednotného panelu pro veškerou komunikaci od potenciálních i opakovaných pacientů a mnoho dalších záležitostí je možné pomocí CRM.

Systém CRM v medicíně pokrývá celou řadu aspektů, od správy pacientů až po analýzu dat a zpětnou vazbu. Systém by měl být schopen přizpůsobit se specifickým potřebám zdravotnické organizace a jejím procesům. Je důležité, aby byl systém CRM schopen integrace s dalšími systémy, jako jsou systémy elektronických lékařských záznamů (EMR) nebo finanční platformy. Systém by měl být schopen škálovat se s rostoucím počtem pacientů a potřebami kliniky. Systém CRM by měl poskytovat pohodlné nástroje pro sjednávání schůzek, získávání informací a komunikaci se zdravotnickým personálem. Je důležité, aby systém fungoval spolehlivě, poskytoval rychlé zpracování dotazů a měl vysokou dostupnost. Cílem systému CRM je zvýšit efektivitu zařízení, zkrátit čekací doby, optimalizovat zdroje a zlepšit kvalitu péče o pacienty. CRM v medicíně nabízí mnoho možností, jak zlepšit správu a interakci s pacienty, pokud je správně implementován, přizpůsoben a je zajištěna bezpečnost dat.

Tabulka 3 uvádí seznam nemocničních informačních systémů, které jsou v Česku nejčastější.

Tabulka 3 Nemocniční informační systémy (HIS) v České republice

Počet instalací	Podniky	Název HIS
43	AKORD SOFTWARE s.r.o.	AKORD PRO
3	DATA-PLAN Bohemia s.r.o.	Net WinMed
6 ?	GUBI	ISZP
20	HiComp Systems CZ s.r.o.	NIS HiComp
14	HIPPO spol. s r.o.	ISpP HIPPO
27 (2) ?	ICZ a.s.	AMIS*H AMIS*H/2000 SAMED
0 ?	JRS data s.r.o.	MedIS
1	LOGIS s.r.o.	HERMES
13	MEDICALC software s.r.o.	WinMedicalc 2000
1		WinMedicalc FN Plzeň

24 (3)	MEDICON a.s.	GreyFox
9	MedisoArt s.r.o.	IS L3
5	PCS Systems s.r.o.	PCS*CARE
4 ?	PROSOFT Kroměříž s.r.o.	TREE
7 32 ?	SMS, spol. s r.o.	CLINICOM CLINICOM - PL
41 (8)	STAPRO s.r.o.	MEDEA
10	STEINER	UNIS
1 ?	SYSTEMA Integral s.r.o.	NIS Systema
9 (1)	VIP Most s.r.o.	
-	WEBCOM a.s.	
9 (1)	WMZ v.o.s.	MEDIX

Zdroj: Medinfo, 2024

CompuGroup Medical ČR, dříve známá jako SMS spol. s r.o., se vyprofilovala v implementaci NIS CLINICOM a široké paletě systémů pro zdravotnické sektory. Nyní nabízí komplexní softwarová a komunikační řešení podporující efektivní péči o pacienta (CGM, 2024).

Datalan a.s., se svými 24 lety na trhu, představuje lídra v poskytování inovativních IT služeb a řešení pro zdravotnictví i další odvětví, s působením od Slovenska po mezinárodní scénu (Datalan, 2024).

MEDAX Systems s.r.o. (2024, založená v roce 2008, vyniká v oblasti zdravotnictví díky svému hlavnímu produktu, modulárnímu informačnímu systému SmartMEDIX®, určenému pro zdravotnická a sociální zařízení, což reflektuje jejich dlouholeté zkušenosti a odbornost v IT a zdravotnickém sektoru.

Společnost STAPRO s.r.o. (2024), významný účastník na trhu s informačními systémy pro zdravotnictví od roku 1990, nabízí širokou paletu vlastního softwaru. Od roku 1993 expanduje také na Slovensko prostřednictvím dceřiné firmy STAPRO Slovensko s.r.o. a vlastní další společnosti, jako jsou MLAB software s.r.o. a HiComp a.s. Nabídka zahrnuje řadu FONS, která představuje nejnovější a tradiční nemocniční informační systémy, laboratorní a manažerské systémy, flexibilní formuláře a nástroje pro integraci a komunikaci.

ICZ a.s., založená v roce 1997 sloučením vedoucích IT společností z České a Slovenské republiky, působí globálně a nabízí rozsáhlé služby v mnoha sektorech včetně

zdravotnictví. Společnost se pyšní vysokými standardy kvality a bezpečnosti, což potvrzují její certifikace ISO a bezpečnostní certifikace NATO. Pro zdravotnické zařízení poskytuje systémy AMISHD, AMISH, obrazové systémy, elektronickou ošetrovatelskou dokumentaci, převod řeči na text a další (Skupina ICZ, 2024).

Od roku 1997 poskytuje společnost Prosoft s.r.o. služby v oblasti nemocničních informačních systémů, s využitím pokročilých databázových technologií CACHE od INTERSYSTEMS. Nabízí komplexní řešení pro laboratoře a bezpečnou výměnu dat v zdravotnictví, včetně systému NIS TREE s PACS a manažerským systémem pro analýzu dat (Prosoft, 2024).

AGEL a.s., největší soukromý poskytovatel zdravotní péče ve střední Evropě, působí od roku 2006 také na Slovensku. Nabízí unikátní NIS IKIS®, který integruje data z celé sítě zdravotnických zařízení, podporující sdílení a využívání informací mezi různými pracovišti. Systém, vyvinutý týmem Medical Systems a.s., znamená revoluční přístup v propojení zdravotnických informací (Agel, 2024).

Společnost Medicalc software s.r.o. poskytuje od roku 2002 nemocniční informační systémy, které podporují činnost a správu nemocnic. Spolupracuje s významnými domácími i zahraničními partnery v oblasti medicíny a informatiky. Nabízí moderní systémy medicalc4 a WinMedicalc, doplněné o produkty pro laboratorní práci a další specifické aplikace (Medicalc, 2024).

MEDICON a.s. patří mezi vedoucí poskytovatele ambulantní péče v Praze, nabízející širokou škálu specializovaných služeb včetně radiodiagnostiky a screeningu karcinomu prsu. Jejich služby zahrnují také jednodenní chirurgii a gastroenterologii (Mediconas, 2024).

Společnost Steiner s.r.o. se od roku 1991 specializuje na vývoj softwaru pro zdravotnictví. Jejich produkty, certifikované dle ISO norm, zahrnují nemocniční a laboratorní informační systémy, systémy pro transfuzní služby a registr dárců krevetvorných buněk. Plánují rozvoj v oblasti komplexních řešení pro darování a zpracování krevetvorných buněk (Steiner, 2024).

4.4 Vícekriteriální rozhodování efektivity vývoje CRM systému

V této práci je použita metoda hodnotící tabulky pro výběr nejlepšího z alternativních produktů pro nákup - CRM systému pro kliniku. Podstatou této metody je

převést hodnocení mnoha kritérií na zobecněné skóre a vybrat alternativu, která získala maximální počet bodů.

Úkol: Je třeba vybrat ze čtyř alternativ - různých modelů systémů CRM a také vývoj vlastního systému CRM. Důležitými ukazateli pro každou z alternativ jsou:

- 1) Použitelnost (K1),
- 2) Integrace se stávajícími systémy (K2),
- 3) cena (K3),
- 4) doba realizace (K4),
- 5) podpora a údržba (K5).

Použitelnost (K1)

Použitelnost CRM systému je klíčovým faktorem, který ovlivňuje jeho efektivitu a přijetí uživateli ve zdravotnických zařízeních. To je důležité zejména ve zdravotnictví, kde pracují specialisté s různou úrovní technických znalostí a kde jakékoli zpoždění nebo chyby v provozu systému mohou vést k vážným následkům. CRM by měl mít dobře strukturované a logicky uspořádané rozhraní, aby uživatelé mohli snadno najít funkce a informace, které potřebují. Rozhraní by mělo být čisté a ne přehlcené, aby nedocházelo k nejasnostem a aby se noví uživatelé snáze učili. Schopnost přizpůsobit rozhraní konkrétním úkolům a preferencím uživatele může výrazně zlepšit uživatelskou zkušenost se systémem. Uživatelé by měli mít možnost přizpůsobit sestavy a řídicí panely tak, aby vyhovovaly jejich konkrétním potřebám.

Je důležité, aby byl CRM přístupný a snadno použitelný na chytrých telefonech a tabletech, aby zaměstnanci mohli mít aktuální informace i mimo kancelář. Vestavěné funkce pro spolupráci, jako je zaslání zpráv a komentářů k záznamům, mohou zjednodušit komunikaci v rámci týmu. Možnost konfigurovat úroveň přístupu pro různé skupiny uživatelů pomáhá zabezpečit citlivé informace a řídit práva k prohlížení a úpravám.

Integrace se stávajícími systémy (K2)

Integrace CRM se stávajícími systémy v nemocnici je klíčem k zajištění efektivního provozu zdravotnického zařízení, zlepšení zkušeností pacientů a zefektivnění interních procesů. Integrace CRM s jinými systémy umožňuje vytvořit jednotný pracovní prostor pro zaměstnance, což snižuje potřebu přepínat mezi různými aplikacemi a platformami. To zvyšuje efektivitu práce tím, že zkracuje čas strávený každodenními úkoly.

Propojení CRM s elektronickými zdravotními záznamy (EMR), účetními a dalšími systémy řízení umožňuje automatizovanou výměnu dat. Například informace o pacientech

registrované v CRM mohou automaticky aktualizovat zdravotní profil pacienta v EMR a údaje o poskytnutých službách mohou být přeneseny do účetního systému pro účely fakturace a účetnictví.

Integrace zabraňuje nesrovnalostem a duplicitě dat v různých systémech. Díky tomu jsou informace o pacientech, jejich léčbě a finančních transakcích aktuální a přesné, což je pro kvalitní zdravotní péči a řízení nemocnice zásadní.

Synchronizace dat mezi CRM a dalšími systémy umožňuje poskytovat pacientům individuální přístup a včasné informace o nadcházejících návštěvách, zákrocích a výsledcích testů. To přispívá ke zvýšení spokojenosti a loajality pacientů.

Integrovaný systém usnadňuje plánování zdravotnického personálu, přidělování zdrojů a analýzu výkonnosti nemocnice. Analytické nástroje CRM, které využívají data z různých systémů, mohou poskytnout cenné poznatky pro optimalizaci procesů a zlepšení kvality péče.

Integrace různých systémů může být technicky náročná kvůli rozdílům v jejich architektuře a protokolech pro výměnu dat. Řešením může být zapojení specializovaných IT konzultantů nebo výběr CRM s předintegrovanými integracemi..

Cena (K3)

Cena systému CRM hraje důležitou roli v rozhodovacím procesu každé organizace, včetně poskytovatelů zdravotní péče. Výběr vhodného systému pro řízení vztahů se zákazníky by měl zohledňovat nejen funkčnost a obchodní potřeby, ale také finanční omezení.

Při výběru CRM je důležité zhodnotit potenciální návratnost investice. Systém by se měl nejen vejít do rozpočtu, ale také přispět ke zvýšení efektivity, lepší péči o pacienty a v důsledku toho i zvýšit příjmy. Kromě počátečních nákladů na licenci nebo předplatné je důležité vzít v úvahu i možné skryté náklady, spojené s implementací a údržbou CRM. Ty mohou zahrnovat náklady na školení zaměstnanců, přizpůsobení systému konkrétním potřebám instituce a náklady na pravidelnou podporu a upgrade. Úplné pochopení všech souvisejících nákladů pomůže předejít nečekaným finančním potížím.

Při výběru CRM je důležité porovnat nejen cenu, ale také funkčnost různých systémů. Některá řešení mohou nabízet širokou škálu funkcí za vyšší cenu, zatímco jiná mohou být cenově dostupnější, ale s omezenou funkčností. Nalezení rovnováhy mezi náklady a potřebami je klíčem k zajištění efektivity investic.

Doba realizace (K4)

Časový rámeček pro zavedení systému CRM ve zdravotnickém zařízení je důležitým aspektem, který je třeba při výběru řešení pečlivě naplánovat a zvážit. Tato doba se může lišit v závislosti na mnoha faktorech, včetně složitosti zvoleného systému, velikosti a specifčnosti zdravotnického zařízení a úrovně školení a motivace personálu. Složitější systémy s širokou škálou funkcí obvykle vyžadují více času na implementaci a přizpůsobení. Doba potřebná k integraci nového CRM se softwarovými produkty, které se v nemocnici již používají, může významně ovlivnit celkovou dobu implementace. Přizpůsobení CRM specifickým procesům a potřebám konkrétního zdravotnického zařízení může dobu implementace prodloužit. Před plným zprovozněním systému by mělo být provedeno testování, aby se zjistily a opravily případné chyby a nedostatky.

Podpora a údržba (K5)

Pravidelná údržba a aktualizace CRM zajišťují dlouhodobou životaschopnost systému a chrání investice do jeho nákupu a implementace. Proto jsou podpora a údržba systému CRM důležitými součástmi pro zajištění jeho stabilního a efektivního provozu ve zdravotnickém zařízení. Správně organizovaná podpora a údržba pomáhá nejen řešit problémy a záležitosti, které se objeví během provozu systému, ale také zajišťuje jeho aktuálnost a soulad s měnícími se potřebami organizace. Pro vyhodnocení nejvhodnějšího systému byly pro srovnání nabídnuty tři CRM systémy, které jsou na českém trhu nejrozšířenější a možnost vyvinout CRM systém na míru. Každá ze čtyř dostupných variant CRM systému byla hodnocena na desetibodové stupnici, přičemž nejvyšší počet bodů získaly alternativy vyznačující se největším komfortem, nejlepší integrací se stávajícími systémy a nejnižší cenou.

Hodnoty zobecněného kritéria S pro každou alternativu pak byly vypočteny prostým sečtením výsledných skóre kritérií.

Tabulka 4 Výsledky hodnocení alternativ podle kritérií

Alternativy (A) / Kritéria (K)	Použitelnost (K1)	Integrace se stávajícími systémy (K2)	Cena (K3)	Doba realizace (K4)	Podpora a údržba (K5)	Součet (S)
Akord Pro (A1)	9	8	6	9	8	40
STAPRO Medea (A2)	7	7	5	6	7	32
WinMedicalc (A3)	9	8	6	5	8	36
UNIS (A4)	7	7	8	6	8	38
AMIS*H (A5)	8	9	6	7	7	37
IKIS® (A6)	8	10	6	6	8	38
CGM CLINICOM (A7)	8	10	7	6	8	39
Vlastní systém (A8)	8	9	10	7	9	43

Zdroj: vlastní zpracování

Výsledkem bylo několik nejlepších možností - možnost vývoje vlastního systému, která získala maximální hodnotu zobecněného kritéria S rovnou 43 bodů a nákup hotového CRM systému AKORD PRO se 40 body. Tyto výsledky jsou však v rozporu s mou intuitivní volbou. Je to způsobeno tím, že jsem nezohlednil rozdílnou důležitost jednotlivých kritérií pro mě.

Dalším krokem je proto zavedení váhových koeficientů V, které zohledňují relativní důležitost každého z těchto pěti kritérií. Protože optimální hodnota zobecněného kritéria S je v našem problému jeho maximální hodnotou, mělo by mít důležitější kritérium vyšší hodnotu váhového koeficientu.

V tomto případě jsou pro toto zdravotnické zařízení nejdůležitějšími kritérii pohodlnost CRM systému, cena a integrace se stávajícími systémy, méně důležitá je doba implementace a podpora a údržba, protože CRM systém ovlivňuje kvalitu a pohodlí plnohodnotné práce personálu a vedení profilů pacientů, je pořizován na mnoho let a rozpočet na nákup je omezený (nepřiměřeně drahé možnosti jsou z mnoha alternativ jednoduše vyloučeny).

Pro výpočet hodnot váhových koeficientů V byla použita metoda Saatiho (Analytický hierarchický proces, AHP), která umožňuje kvantifikovat preference mezi kritérii prostřednictvím matice párového porovnání. Tato metoda je založena na stanovení vah kritérií na základě expertních odhadů o jejich relativní důležitosti.

V rámci metody Saatiho jsou hodnoty koeficientů na diagonále matice párového porovnání vždy rovny jedné, jelikož každé kritérium je považováno za rovnocenné samo sobě. Hodnoty mimo diagonálu jsou určeny na základě subjektivního odhadu důležitosti jednoho kritéria ve vztahu k druhému. Hodnoty jsou obvykle v rozmezí od 1/9 (kritérium i je mnohem méně důležité než kritérium j) až 9 (kritérium i je mnohem důležitější než kritérium j), kde hodnoty mezi těmito extrémy vyjadřují různé stupně preference.

Pro každý pár kritérií i a j se tedy určí hodnota v_{ij} takto:

- $v_{ij} > 1$, pokud je kritérium i důležitější než kritérium j ,
- $v_{ij} = 1$, pokud jsou kritéria i a j stejně důležitá,
- $v_{ij} < 1$, pokud je kritérium i méně důležité než kritérium j .

Tabulka 5 Výsledky párového porovnání kritérií V

Kritéria	Pohodlnost (V1)	Cena (V2)	Integrace (V3)	Implementace (V4)	Podpora (V5)	Součet	Význam
Pohodlnost (V1)	1	0.333	0.2	5	3	9.533	0.164
Cena (V2)	5	1	0.333	5	3	14.333	0.246
Integrace	5	3	1	7	5	21.000	0.360

(V3)						0	
Implementace (V4)	0.2	0.2	0.143	1	5	6.543	0.112
Podpora (V5)	0.333	0.333	0.2	5	1	6.866	0.118
Součet	-	-	-	-	-	58.275	1.00

Zdroj: vlastní zpracování

Po vyplnění tabulky byla vypočtena úroveň důležitosti každého kritéria V součtem hodnot v řádcích tabulky a také celková úroveň důležitosti všech kritérií. Poté byly potřebné hodnoty vah kritérií V získány vydělením úrovní důležitosti jednotlivých kritérií jejich celkovou úrovní důležitosti, výsledky jsou uvedeny v tabulce 6.

Tabulka 6 Váhy relativní důležitosti hodnotících kritérií.

Váhové faktory (V)	Kritéria
Použitelnost (V1)	0.164
Integrace se stávajícími systémy (V2)	0.246
Cena (V3)	0.360
Doba realizace (V4)	0.112
Podpora a údržba (V5)	0.118
Součet	1.00

Zdroj: vlastní zpracování

V tomto případě se zobecněné kritérium vypočítá pomocí vzorce:

$$S = K1*V1 + K2*V2 + K3*V3 + K4*V4 + K5*V5.$$

Výsledky výpočtů jsou uvedeny v tabulce 7. K automatizaci výpočtů jsou použity excelové tabulky.

Tabulka 7 Výpočet zobecněného kritéria se zohledněním váhových koeficientů.

Alternativy (A) / Kritéria (K)	Použitelnost (K1) K*V	Integrace se stávajícími systémy (K2) K*V	Cena (K3) K*V	Doba realizace (K4) K*V	Podpora a údržba (K5) K*V	Součet (S)
Akord Pro (A1)	1.476	1.968	2.16	1.008	0.944	7.556
STAPRO Medea (A2)	1.148	1.722	1.8	0.672	0.826	6.168
WinMedicalc (A3)	1.476	1.968	2.16	0.56	0.944	7.108
UNIS (A4)	1.148	1.722	2.88	0.672	0.944	7.366
AMIS*H (A5)	1.312	2.214	2.16	0.784	0.826	7.296
IKIS® (A6)	1.312	2.46	2.16	0.672	0.944	7.548
CGM CLINICOM (A7)	1.312	2.46	2.52	0.672	0.944	7.908
Vlastní systém (A8)	1.312	2.214	3.6	0.784	1.062	8.972

Zdroj: vlastní zpracování

Na základě analýzy předložených údajů můžeme konstatovat, že nejvhodnější variantou CRM systému z navrhovaných je vývoj vlastního systému (A8). Tato varianta získala nejvyšší zobecněné kritérium s váhovými koeficienty rovnými 8.972.

Na základě získaných údajů má ukazatel Cena (V3) vysoký váhový faktor (0,360) a je rozhodující pro efektivní využívání CRM. „Vlastní systém“ vykázal nejlepší skóre (10) a celkové skóre po vynásobení váhovým faktorem je 3,6 , což svědčí o vysoké konkurenceschopnosti a rezervě v plánu modernizace díky vkládání volných finančních prostředků. To z něj činí nejatraktivnější možnost z hlediska rozpočtu. V hodnocení integrace se stávajícími systémy (K2) vykazuje „vlastní systém“ rovněž vynikající skóre 2,214, což zdůrazňuje jeho flexibilitu a schopnost efektivně pracovat v rámci stávající IT

infrastruktury zdravotnického zařízení. Podpora a údržba (K5) má skóre (9) a zobecněné vážené skóre (1.062) naznačuje, že vlastní systém má dostatečně umírněnou míru podpory.

Použitelnost má vážené skóre (1.312) a skóre (8) co je také velmi průměrné ukazatele, ale tento ukazatel bude možné rychle a relativně levně škálovat k lepším hodnotám

Doba realizace (doba implementace) (K4): Poslední kritérium má váhovou hodnotu(0,112), přestože tento kritérium má nižší váhovou hodnotu a hodnocení vlastního systému také není tak vysoké (7), lze tento kritérium považovat za dlouhodobou investici, která mimo jiné přináší bonusy i dalším kritériím

Tyto faktory dohromady dělají ze systému „Vlastní systém“ nejpreferovanější volbu pro zdravotnická zařízení, která hledají CRM systém zajišťující vysokou spokojenost uživatelů, flexibilní integraci, optimální náklady, rychlou implementaci a kvalitní podporu.

4.5 Směry vývoje CRM

Na základě analýzy obchodních procesů kliniky a stávajících řešení na trhu CRM systémů pro kliniku byly definovány následující vývojové úkoly:

1. Vytvořit elektronickou databázi pro společnost, která bude evidovat klienty a služby, které jim byly poskytnuty.
2. Vyvinout aplikaci, která bude sloužit ke spolupráci s databází kliniky pro poskytování služeb, s přihlédnutím k dříve analyzovaným CRM systémům.
3. Ve vyvinuté aplikaci realizovat samostatné moduly, které umožní zjednodušit obsluhu klientů kliniky a práci personálu.

Po automatizaci procesů kliniky „ABC“ se plánuje dosažení následujících výsledků:

- zkrácení doby potřebné ke zpracování lékařské dokumentace,
- automatizace pracovních postupů zaměstnanců kliniky,
- dlouhodobé ukládání údajů o zaměstnancích a klientech kliniky.

4.6 Charakteristika funkcionality vyvinutého CRM systému

Předpokládaný CRM systém bude disponovat následujícími klíčovými funkcionalitami: systém umožní evidovat základní údaje o pacientech, včetně jména, příjmení, data narození, pohlaví, kontaktního čísla, e-mailu a adresy bydliště. Tato funkcionalita poslouží jako základ pro další interakce se systémem. Nabídne možnost plánování a správy lékařských termínů pro pacienty s detaily o účelu návštěvy a

poznámkách, což zefektivní organizaci práce. Zahrne správu nabízených lékařských služeb s popisem a ceníkem, což lékařům umožní poskytovat pacientům přehled o dostupných službách a jejich nákladech. Umožní uchovávání a správu lékařských záznamů pacientů, včetně diagnózy, předpisů a lékařských poznámek, což je důležité pro sledování historie léčby. Poskytne integraci komunikačních nástrojů, jako jsou e-maily, chaty nebo videohovory, zlepší komunikaci mezi lékařem a pacientem. Umožní automatizaci běžných procesů, jako je zadávání termínů a správa zásob, čímž se sníží administrativní zátěž. Efektivní využívání systému přispěje k optimalizaci nákladů a zdrojů, což je významný faktor pro ekonomickou efektivitu zdravotnického zařízení.

V rámci vývoje CRM systému byla vytvořena databázová struktura skládající se ze čtyř hlavních tabulek: pacienti (Patients), schůzky (Appointments), služby (Services) a lékařské záznamy (Medical Records), které jsou navzájem propojeny pomocí vztahů „mnoho k jednomu“ a „jeden k mnoha“.

Tabulka pacientů obsahuje základní informace o pacientech, včetně jejich jména, pohlaví, kontaktních údajů a adresy. Každý pacient má vlastní ID.

Tabulka schůzek uchovává informace o schůzkách mezi lékařem a pacientem, včetně data schůzky a jejího účelu. Každá schůzka je spojena s konkrétním pacientem prostřednictvím cizího klíče.

Tabulka služeb obsahuje seznam služeb nabízených zdravotnickým střediskem s podrobnostmi o každé službě, jako je název služby, popis a cena. Služby jsou spojeny se schůzkami a pacienty, což umožňuje sledovat, které služby byly poskytnuty během konkrétní schůzky.

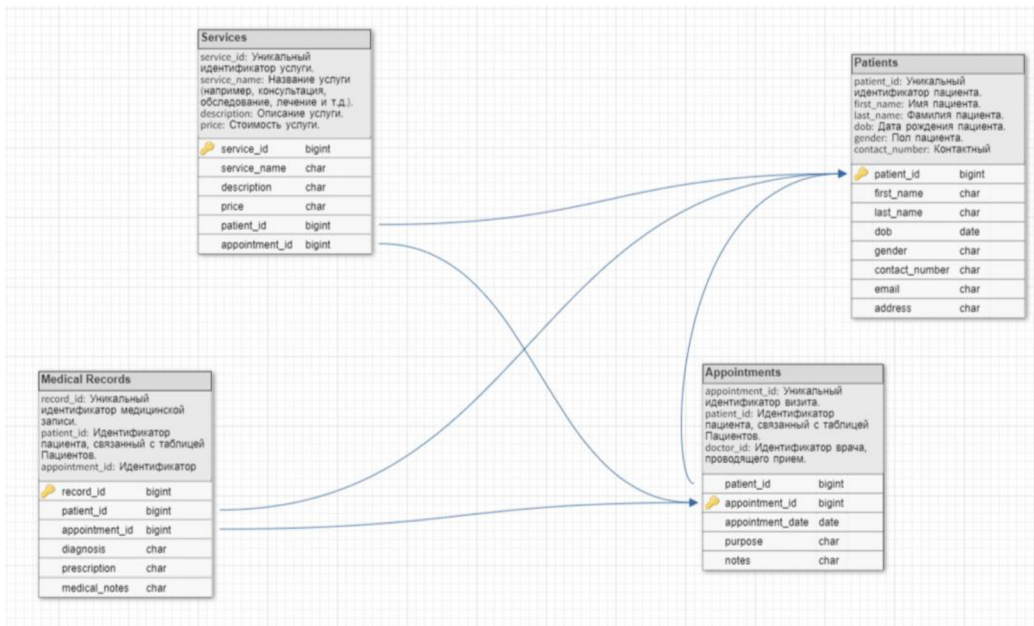
Tabulka lékařských záznamů obsahuje diagnostické a další odborné informace o pacientech, včetně diagnózy, předpisů a lékařských poznámek. Tyto záznamy jsou spojeny s pacienty a schůzkami, což umožňuje uchovávat komplexní historii zdravotní péče každého pacienta.

Vztahy mezi tabulkami jsou navrženy tak, aby odrážely reálné procesy v zdravotnickém středisku. Vztah „mnoho k jednomu“ mezi schůzkami a pacienty umožňuje, aby jeden pacient měl více schůzek, zatímco každá schůzka je přiřazena právě jednomu pacientovi. Vztah „jeden k mnoha“ mezi lékařskými záznamy a pacienty i schůzkami zajišťuje, že každý záznam je spojen s jedním pacientem a jednou schůzkou, ale jeden pacient nebo jedna schůzka může mít více záznamů. Tato struktura umožňuje efektivní správu a analýzu dat v rámci CRM systému.

Přiložené skripty pro vytváření tabulek a omezení pole demonstrují, jak lze v praxi databázovou strukturu implementovat, a zároveň definují pravidla pro integritu a konsistenci dat. Například, použití cizích klíčů pro propojení tabulek zajišťuje, že data mezi tabulkami zůstanou konzistentní a relevantní vztahy jsou dodrženy.

Navržená databázová struktura CRM systému poskytuje pevný základ pro správu informací o pacientech, schůzkách, službách a lékařských záznamech, což přispívá k lepší organizaci a efektivitě zdravotnického zařízení.

Obrázek 4. Databázová struktura CRM



Zdroj: vlastní zpracování

4.7 Create Script

Tento kód ze souboru admin.py (příloha 1) zohledňuje modely pro tabulky pacienti, schůzky, služby a lékařské záznamy a také zahrnuje administrativní funkce jako vyhledávání, filtrování a úpravy vložených záznamů pro přívětivé uživatelské rozhraní administrace.

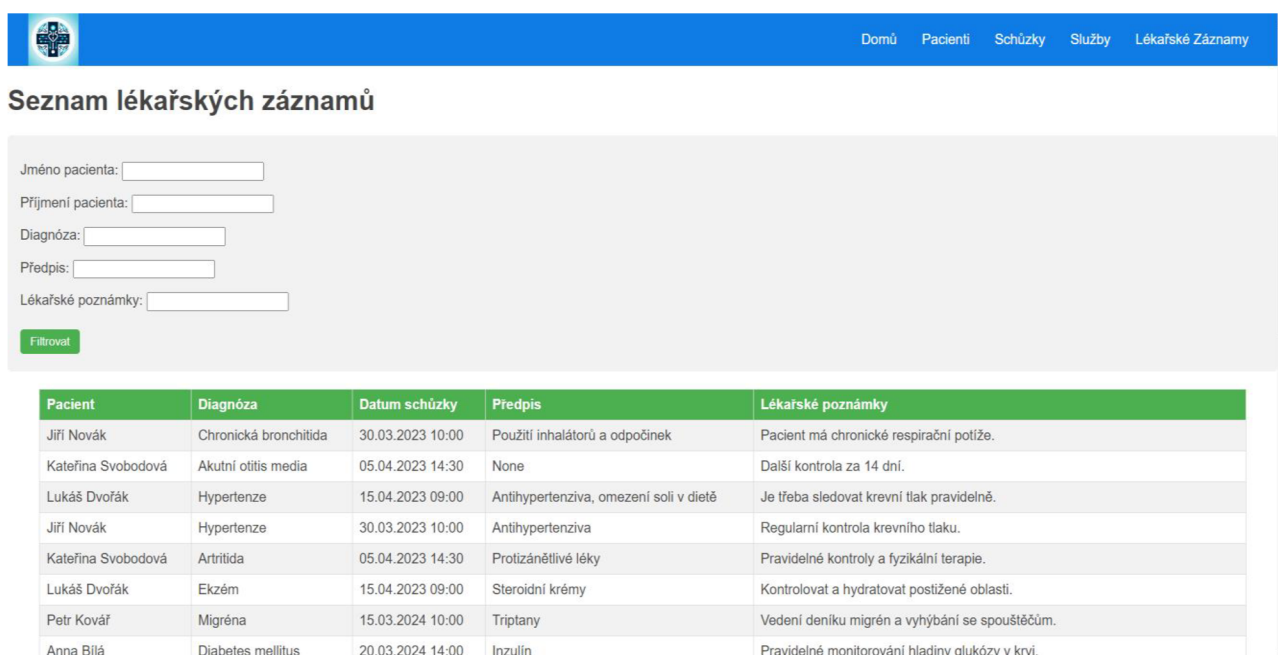
Zahrnuje:

Přizpůsobená administrační rozhraní pro každý z modelů (pacient, schůzka, služba, lékařský záznam).

Možnosti úprav vložených záznamů (SchuzkaInline, SluzbaInline, LekarskyZaznamInline) pro zjednodušení správy souvisejících záznamů přímo ze stránek administrace pacientů nebo schůzek.

Možnosti vyhledávání a filtrování pro snazší navigaci a správu dat v rámci administračního webu Django.

Obrázek 5. Příklad možnosti vyhledávání a filtrování lékařských záznamů



Seznam lékařských záznamů

Jméno pacienta:

Příjmení pacienta:

Diagnóza:

Předpis:

Lékařské poznámky:

Pacient	Diagnóza	Datum schůzky	Předpis	Lékařské poznámky
Jiří Novák	Chronická bronchitida	30.03.2023 10:00	Použití inhalátorů a odpočinek	Pacient má chronické respirační potíže.
Kateřina Svobodová	Akutní otitis media	05.04.2023 14:30	None	Další kontrola za 14 dní.
Lukáš Dvořák	Hypertenze	15.04.2023 09:00	Antihypertenziva, omezení soli v dietě	Je třeba sledovat krevní tlak pravidelně.
Jiří Novák	Hypertenze	30.03.2023 10:00	Antihypertenziva	Regulární kontrola krevního tlaku.
Kateřina Svobodová	Artritida	05.04.2023 14:30	Protizánětlivé léky	Pravidelné kontroly a fyzikální terapie.
Lukáš Dvořák	Ekzém	15.04.2023 09:00	Steroidní krémy	Kontrolovat a hydratovat postižené oblasti.
Petr Kovář	Migréna	15.03.2024 10:00	Triptany	Vedení deníku migrén a vyhýbání se spouštěčům.
Anna Bílá	Diabetes mellitus	20.03.2024 14:00	Inzulín	Pravidelné monitorování hladiny glukózy v krvi.

Zdroj: vlastní zpracování

V těchto třídách bylo přidáno několik funkcí:

`fields`: Specifikují, která pole se mají zobrazit v inline formulářích pro `SchuzkaInline` a `SluzbaInline`.

`list_filter`: Umožňuje filtrovat seznam objektů v administračním rozhraní na základě zadaných kritérií (např. podle pohlaví u `PatientAdmin` nebo podle data schůzky u `SchuzkaAdmin`).

`ordering`: Určuje defaultní řazení objektů v administračním seznamu.

`date_hierarchy`: Umožňuje navigaci v čase v seznamu objektů (použito u `SchuzkaAdmin` pro navigaci schůzek podle data).

Tato rozšíření poskytují užitečné nástroje pro efektivní správu a přehled o datech v rámci CRM systému přímo z Django administračního rozhraní.

4.8 Soubor forms.py

V medicínské CRM aplikaci na platformě Django jsou definovány formuláře, které umožní uživatelům vytvářet a upravovat záznamy o pacientech, schůzkách, službách a lékařských záznamech prostřednictvím webového rozhraní (příloha 2). Tyto formuláře by měly odpovídat strukturám modelů a admin třídám.

Obrázek 6. Formuláře

Formuláře obsahují následující prvky:



Zdroj: vlastní zpracování

PatientForm: Pro vytváření a úpravu záznamů pacientů. Obsahuje všechna pole definovaná v modelu pacient a používá speciální widget pro datum narození, aby se uživatelům zobrazil výběr data.

SchuzkaForm: Slouží k plánování a úpravě termínů schůzek. Zahrnuje pole pro výběr pacienta, datum a čas schůzky, její účel a poznámky. Datum a čas schůzky jsou prezentovány pomocí widgetu pro datetime-local, což uživatelům umožňuje snadný výběr.

SluzbaForm: Umožňuje správu služeb poskytovaných zdravotnickým zařízením. Obsahuje pole pro název služby, její popis, cenu a schůzku, ke které je služba přiřazena. Cena má speciální widget pro číselný vstup s krokem 0.01.

LekarskyZaznamForm: Slouží k vytváření a úpravě lékařských záznamů s informacemi o diagnóze, předpisech a poznámkách.

4.9 Models.py

V rámci medicínského CRM systému budou modely pacient a schůzka sloužit jako základ pro správu informací o pacientech a jejich schůzkách s lékaři. Tyto modely umožňují evidovat a spravovat klíčové údaje o pacientech a plánovat jejich schůzky, což je nezbytné pro efektivní fungování zdravotnického zařízení (příloha 3). Níže je podrobnější charakteristika funkcionalit těchto modelů:

Obrázek 7. Seznam služeb

Seznam služeb

Název služby:
Popis:
Minimální cena:
Maximální cena:

Název služby	Popis	Cena
Konzultace	První konzultace s lékařem	500.00
Očkování	Očkování proti chřipce	300.00
Rutinní kontrola	Roční preventivní lékařská prohlídka	700.00
Konzultace	První konzultace s lékařem, pro diskuzi o zdravotním stavu a plánování léčby.	500.00
Očkování proti chřipce	Sezónní očkování proti chřipce pro prevenci.	300.00
Kontrolní prohlídka	Pravidelná roční kontrola zdravotního stavu.	700.00
Ultrazvuk srdce	Diagnostické vyšetření srdce ultrazvukem.	1200.00
Rentgen plic	Vyšetření plic rentgenem pro detekci potenciálních problémů.	800.00
Dentální kontrola	Kontrola zubů a ústní dutiny u zubaře.	600.00
Derma kontrola	Vyšetření kůže, léčba kožních problémů.	500.00
Fyzioterapie	Série fyzioterapeutických sezení pro rehabilitaci.	1500.00

Zdroj: vlastní zpracování

Model pacient

Tento model slouží k evidenci základních údajů o pacientech, které jsou nezbytné pro identifikaci pacienta a jeho komunikaci se zdravotnickým zařízením. Obsahuje následující pole:

Jméno a příjmení: Umožňuje identifikaci pacienta. Tato data jsou základními údaji potřebnými pro jakoukoliv interakci mezi pacientem a zdravotnickým zařízením.

Datum narození: Důležitý údaj pro ověření věku pacienta a pro poskytování věkově specifické péče.

Pohlaví pacienta: Informace o pohlaví může být relevantní pro určité typy léčebných postupů a pro správnou identifikaci pacienta.

Kontaktní číslo a e-mail: Základní kontaktní údaje, které umožňují zdravotnickému zařízení komunikovat s pacientem, ať už pro potvrzení schůzky, nebo pro poskytování důležitých informací.

Adresa: Adresa bydliště pacienta, která může být důležitá pro administrativní účely nebo v případě potřeby domácí péče.

Model schůzka

Tento model je navržen pro plánování a správu schůzek pacientů se zdravotnickým personálem. Obsahuje následující atributy:

Pacient: Cizí klíč, který odkazuje na model pacient. Tento vztah umožňuje spojení schůzky s konkrétním pacientem.

Datum schůzky: Přesný čas a datum schůzky. Tato informace je klíčová pro plánování a organizaci práce lékařů a zdravotnického personálu.

Účel schůzky: Stručný popis účelu schůzky, který pomáhá lékařům připravit se na konkrétního pacienta.

Poznámky: Volitelné pole pro jakékoli další informace nebo specifické požadavky týkající se schůzky, které mohou být důležité pro lékaře.

Model služba

Tento model je navržen pro správu lékařských služeb spojených s konkrétními schůzkami. Je důležitý pro sledování nabízených služeb a jejich cen, což umožňuje lékařům a pacientům mít přehled o finančních aspektech léčby.

ForeignKey Schůzka: Toto pole vytváří vztah mezi službou a konkrétní schůzkou. Díky tomuto je možné přesně určit, které služby byly poskytnuty v rámci každé schůzky. Tento vztah také usnadňuje agregaci informací o službách pro zprávy a analýzy.

Název služby: Unikátní identifikátor služby, který usnadňuje její rozpoznání a správu. Toto pole by mělo být dostatečně popisné, aby lékaři a pacienti mohli snadno pochopit povahu služby.

Popis: Detailní popis služby, který může obsahovat informace o postupech, očekávaných výsledcích, a jakékoliv další relevantní informace. Toto pole pomáhá pacientům lépe pochopit, co služba zahrnuje.

Cena: Finanční náklady spojené se službou. Toto pole je zásadní pro účtování a finanční plánování jak pro zdravotnické středisko, tak pro pacienty.

Model lékařský záznam

Model pro uchovávání lékařských záznamů o pacientech, který je důležitý pro sledování historie léčby, diagnóz a předpisů.

ForeignKey Patient a Schůzka: Tyto vztahy umožňují přiřadit lékařský záznam konkrétnímu pacientovi a konkrétní schůzce. Toto usnadňuje sledování léčebných postupů a historie pacienta.

Diagnóza: Textové pole, které obsahuje informace o diagnóze pacienta. Je zásadní pro lékařskou dokumentaci a umožňuje lékařům a dalším zdravotnickým pracovníkům rychle získat přehled o zdravotním stavu pacienta.

Předpis: Volitelné pole pro zaznamenání jakýchkoli léků nebo terapií předepsaných pacientovi. Toto pole pomáhá ke sledování léčebného plánu a zajištění kontinuity péče.

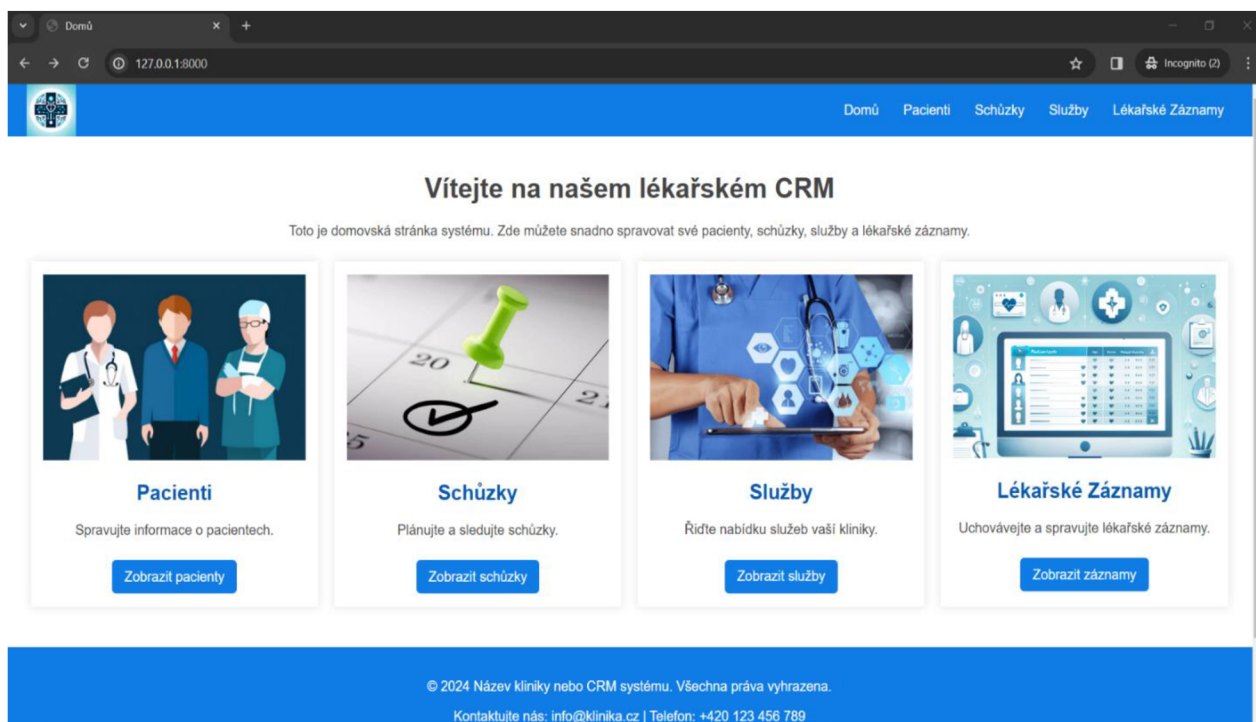
Lékařské poznámky: Další pole pro jakékoli poznámky nebo doplňující informace, které lékař považuje za důležité. Toto může zahrnovat podrobnosti o reakci pacienta na léčbu, zvláštní pokyny, nebo další pozorování.

4.10 Urls.py

Soubor `urls.py` je konfigurační soubor (příloha 4) pro Django aplikaci, slouží k definování URL adres pro webovou aplikaci. Obsahuje definice cest (`path`), které jsou propojeny s pohledy (`views`), jež zpracovávají požadavky uživatelů a vrací odpovídající odpovědi.

Hlavní stránka systému (`/`) slouží jako vstupní bod pro uživatele, kde mohou získat přehled o systému nebo navigovat k dalším funkcím.

Obrázek 8. Hlavní stránka systému



Zdroj: vlastní zpracování

Seznam pacientů ('/pacienti/') a detail pacienta ('/pacient/<int:pk>/') umožňují uživatelům prohlížet seznam všech pacientů evidovaných v systému a získat detailní informace o konkrétním pacientovi.

Funkce pro přidání ('/pacient/add/'), úpravu ('/pacient/edit/<int:pk>/') a odstranění ('/pacient/delete/<int:pk>/') pacientů poskytují možnost spravovat záznamy pacientů přímo z webového rozhraní.

Obdobně, cesty pro správu schůzek ('/schuzky/', '/schuzka/add/', atd.), služeb ('/sluzby/', '/sluzba/add/', atd.) a lékařských záznamů ('/zaznamy/', '/zaznam/add/', atd.) umožňují efektivní správu.

Každá z těchto cest je propojena s konkrétním pohledem, který implementuje logiku pro zpracování požadavků a generování odpovědí, včetně formulářů pro vytváření a úpravu záznamů a zobrazení detailů.

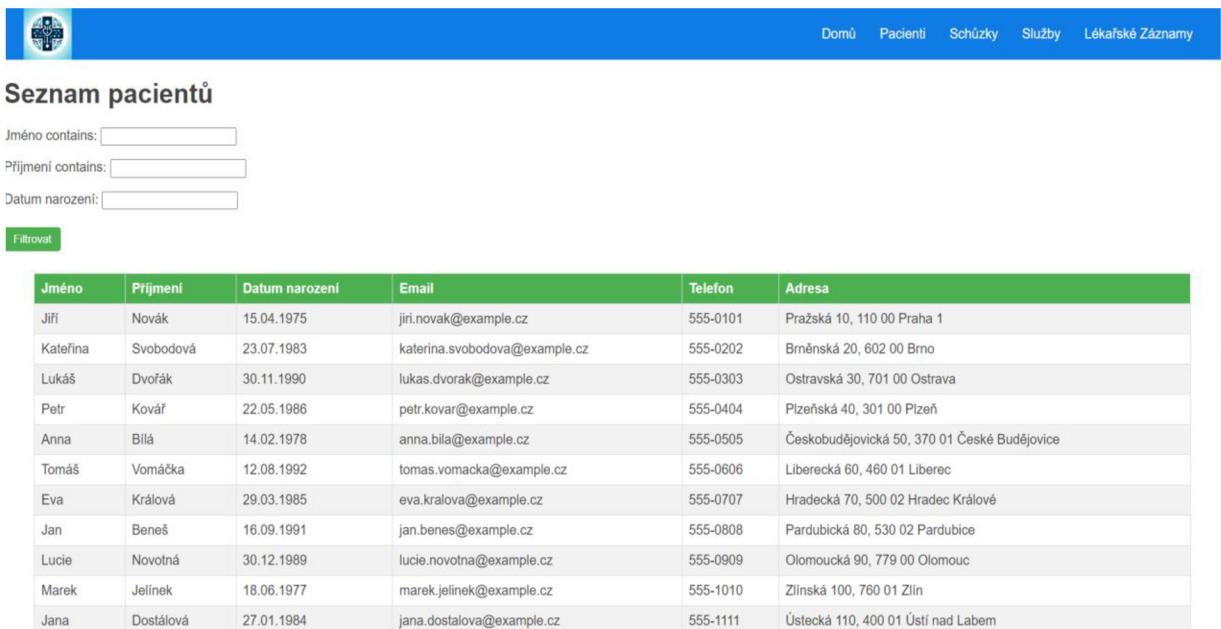
Tato konfigurace URL adres tedy tvoří základ pro navigaci a interakci uživatelů s CRM systémem, což umožňuje efektivní správu pacientů, schůzek, služeb a lékařských záznamů ve zdravotnickém zařízení. Výsledkem je zvýšení efektivity a kvality poskytovaných služeb pacientům.

4.11 Views.py

Část views.py obsahuje několik tříd a funkcí založených na Django views, které zpracovávají požadavky z URL konfigurace a interagují s databází pro získání, vytvoření, aktualizaci nebo odstranění dat. Zde je příklad jen několika tříd:

Tato implementace **PacientDetailView** umožňuje nejen zobrazit detailní informace o pacientovi, ale také poskytuje přehled o všech schůzkách, službách a lékařských záznamech spojených s daným pacientem. Funkce **get_context_data** schopna do kontextu šablony přidávat dodatečné seznamy dat, které pak se da v šabloně patient_detail.html využít pro zobrazení souvisejících informací.

Obrázek 9. PatientListView



Jméno	Příjmení	Datum narození	Email	Telefon	Adresa
Jiří	Novák	15.04.1975	jiri.novak@example.cz	555-0101	Pražská 10, 110 00 Praha 1
Kateřina	Svobodová	23.07.1983	katerina.svobodova@example.cz	555-0202	Brněnská 20, 602 00 Brno
Lukáš	Dvořák	30.11.1990	lukas.dvorak@example.cz	555-0303	Ostravská 30, 701 00 Ostrava
Petr	Kovář	22.05.1986	petr.kovar@example.cz	555-0404	Pižeňská 40, 301 00 Pízeň
Anna	Bílá	14.02.1978	anna.bila@example.cz	555-0505	Českobudějovická 50, 370 01 České Budějovice
Tomáš	Vomáčka	12.08.1992	tomas.vomacka@example.cz	555-0606	Liberecká 60, 460 01 Liberec
Eva	Králová	29.03.1985	eva.kralova@example.cz	555-0707	Hradecká 70, 500 02 Hradec Králové
Jan	Beneš	16.09.1991	jan.benes@example.cz	555-0808	Pardubická 80, 530 02 Pardubice
Lucie	Novotná	30.12.1989	lucie.novotna@example.cz	555-0909	Olomoucká 90, 779 00 Olomouc
Marek	Jelínek	18.06.1977	marek.jelinek@example.cz	555-1010	Zlínská 100, 760 01 Zlín
Jana	Dostálová	27.01.1984	jana.dostalova@example.cz	555-1111	Ústecká 110, 400 01 Ústí nad Labem

Zdroj: vlastní zpracování

Třída **PacientListView** je již dobře nastavená pro zobrazení seznamu pacientů v CRM systému. Navíc je možná možnost zahrnout možnost filtrování pacientů podle různých kritérií, jako jsou jméno, příjmení, datum narození, nebo jakékoli další atributy, které by mohly být užitečné pro uživatele systému.

Pro zavedení filtrů využito Django **FilterSet** třídu, která je součástí externí knihovny `django-filter`. Tato knihovna umožňuje definovat filtry, které lze snadno použít v rámci pohledů pro filtrování querysetů.

V tomto příkladu `PacientFilter` definuje několik filtrů, které mohou být použity pro vyhledávání pacientů podle jména, příjmení, nebo data narození. `FilterView` je použit místo `ListView` pro integraci filtrů do pohledu. Uživatelům tak bude umožněno filtrovat seznam pacientů podle zadaných kritérií, což zlepší uživatelskou přívětivost a efektivitu systému při vyhledávání konkrétních záznamů pacientů.

Třída **PacientCreateView** funkcionalita:

Obrázek 10. PacientCreateView

Přidat pacienta

Jméno:

Příjmení:

Datum narození:

mm/dd/yyyy

Pohlaví:

Muž

Kontaktní číslo:

Email:

Adresa:

Uložit

Zdroj: vlastní zpracování

Validace formuláře: Bylo by možné přidat další logika pro validaci formuláře, aby bylo zajištěno, že data vstupující do vaší databáze jsou správná a úplná. To může zahrnovat například ověření, že e-mailová adresa je ve správném formátu, nebo že kontaktní číslo obsahuje správný počet číslic.

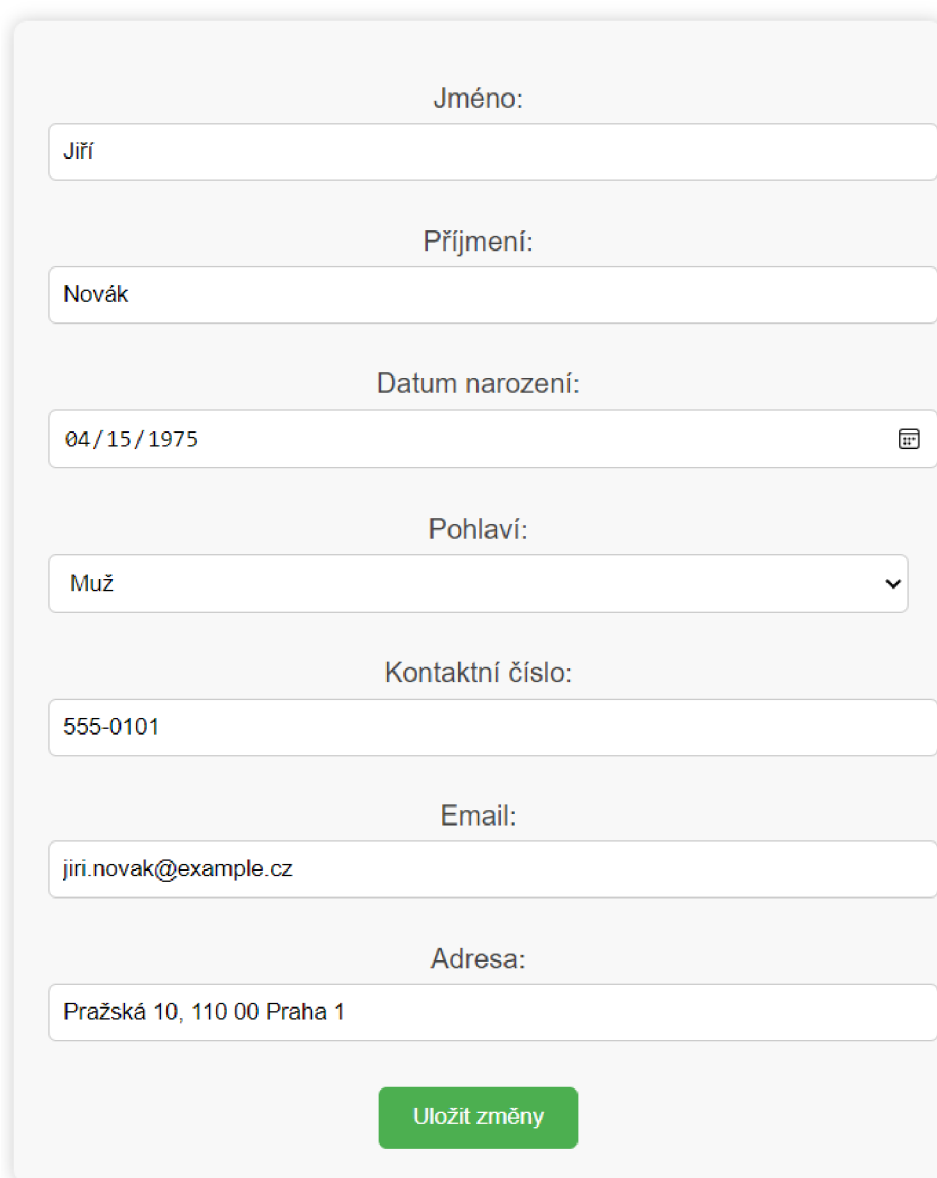
Vlastní zpětné volání po úspěšném uložení: Mohlo by se implementovano metodu `form_valid`, která umožňuje provést dodatečné akce po úspěšném uložení formuláře,

například odeslání e-mailu s potvrzením registrace pacienta nebo přesměrování na vlastní úspěšnou stránku s podrobnostmi o nově přidáném pacientovi.

Dynamické nastavení `success_url`: Místo pevně nastaveného `success_url` mohlo by se dynamicky určit URL pro přesměrování po úspěšném uložení na základě kontextu nebo dat formuláře. To může být užitečné, pokud chcete po přidání pacienta zobrazit jeho detailní stránku.

Třída **PacientUpdateView** umožňuje zpracování situací, které mohou nastat při úpravě informací o pacientovi. Obrázek 11. PatientUpdateView

Upravit pacienta



Jméno:

Jiří

Příjmení:

Novák

Datum narození:

04/15/1975

Pohlaví:

Muž

Kontaktní číslo:

555-0101

Email:

jiri.novak@example.cz

Adresa:

Pražská 10, 110 00 Praha 1

Uložit změny

Zdroj: vlastní zpracování

Možnosti `PatientUpdateView`:

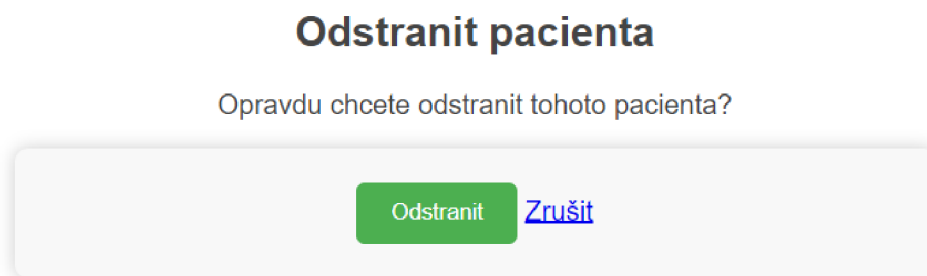
- přesměrování na dynamickou URL: Místo pevně nastaveného `success_url` lze přesměrovat uživatele na stránku, která je více relevantní v kontextu úpravy, například zpět na detailní stránku pacienta. To lze provést přepsáním metody `get_success_url`.

- ověření oprávnění uživatele: V případě potřeby omezit úpravu informací o pacientovi pouze na určité uživatele nebo skupiny lze využít mixiny `UserPassesTestMixin` pro ověření, zda má aktuálně přihlášený uživatel oprávnění k provedení úpravy.

- předvyplnění formuláře: Pokud existují informace, které je třeba předvyplnit ve formuláři pro úpravu (například aktuální datum nebo informace o lékaři), lze to provést v metodě `get_initial`.

Třída **`PatientDeleteView`** umožňuje potvrzení odstranění pacienta z databáze

Obrázek 12. `PatientDeleteView`



© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.

Kontaktujte nás: info@klinika.cz | Telefon: +420 123 456 789

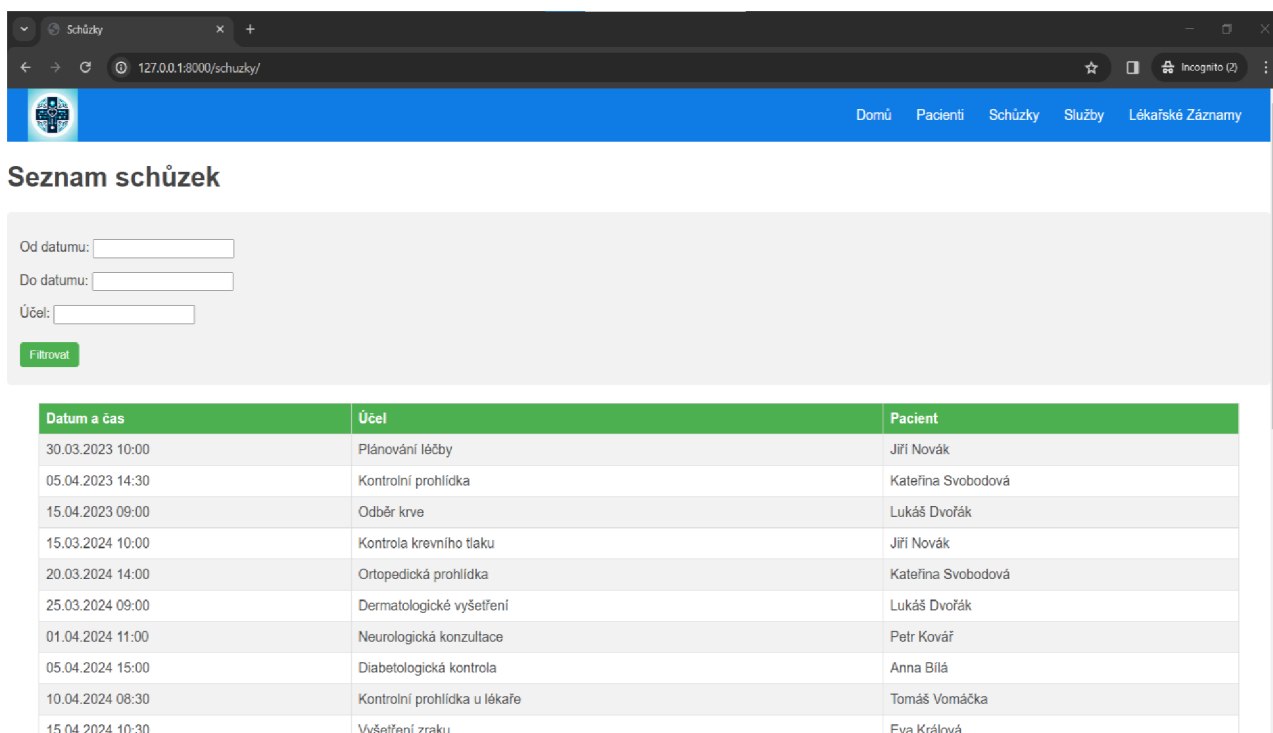
Zdroj: vlastní zpracování

Ověření oprávnění uživatele: Stejně jako u předchozího příkladu je možné pomocí `UserPassesTestMixin` omezit možnost odstranění pouze na určité uživatele nebo skupiny.

Vlastní zpráva po odstranění: Pro lepší informování uživatele o výsledku operace lze použít Django zprávy (`django.contrib.messages`) pro zobrazení upozornění po úspěšném odstranění. Přesměrování na dynamickou URL: Místo pevně nastaveného `success_url` lze implementovat logiku pro dynamické přesměrování, například zpět na stránku, odkud byla odstranění akce zahájena.

Třída **SchuzkaListView** zajišťuje zobrazení seznamu všech schůzek v CRM systému. Tento seznam je přehledně prezentován na stránce `schuzka_list.html`, kde je každá schůzka reprezentována základními informacemi, jako jsou datum schůzky, její účel a jméno pacienta. Seznam umožňuje uživatelům rychle najít potřebné schůzky a získat o nich přehled.

Obrázek 13. SchuzkaListView



Datum a čas	Účel	Pacient
30.03.2023 10:00	Plánování léčby	Jiří Novák
05.04.2023 14:30	Kontrolní prohlídka	Kateřina Svobodová
15.04.2023 09:00	Odběr krve	Lukáš Dvořák
15.03.2024 10:00	Kontrola krevního tlaku	Jiří Novák
20.03.2024 14:00	Ortopedická prohlídka	Kateřina Svobodová
25.03.2024 09:00	Dermatologické vyšetření	Lukáš Dvořák
01.04.2024 11:00	Neurologická konzultace	Petr Kovář
05.04.2024 15:00	Diabetologická kontrola	Anna Bílá
10.04.2024 08:30	Kontrolní prohlídka u lékaře	Tomáš Vomačka
15.04.2024 10:30	Vyšetření zraku	Eva Králová

Zdroj: vlastní zpracování

Třída **SchuzkaDetailView** poskytuje detailní pohled na konkrétní schůzku. Na stránce `schuzka_detail.html` jsou zobrazeny všechny relevantní informace o schůzce,

včetně účelu schůzky, poznámek, a informací o pacientovi. Tento pohled umožňuje lékařům a dalším zaměstnancům získat komplexní přehled o konkrétní schůzce.

Obrázek 14. SchuzkaDetailView

Detail Schůzky

Datum schůzky: 15. dubna 2024, 14:00

Účel schůzky: Roční kontrolní prohlídka

Poznámky: Pacient si stěžuje na občasné bolesti hlavy. Doporučena hydratace a pravidelný spánek.

Pacient: Jan Novák

Kontaktní číslo pacienta: +420 123 456 789

Email pacienta: jan.novak@example.cz

Adresa pacienta: Ulice Příkladová 123, Praha

Lékař: MUDr. Anna Vzorová

Poznámky lékaře: Dále sledovat symptomy a případně upravit léčbu po dalším testování.

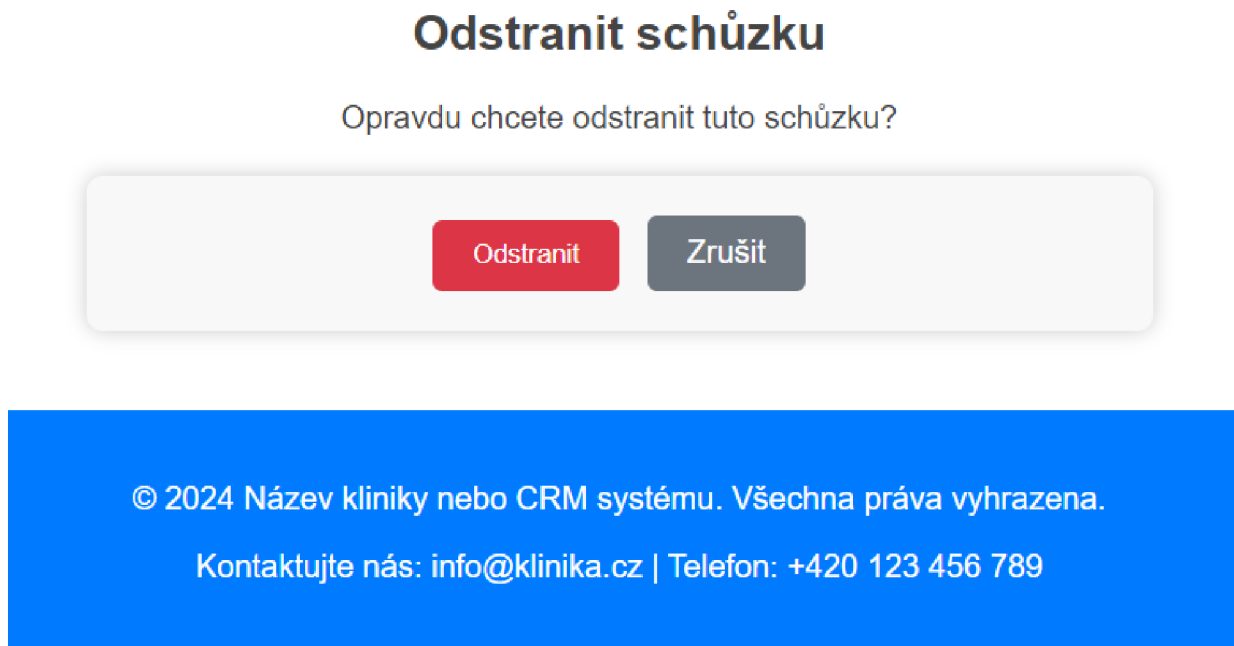
© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.
Kontaktujte nás: info@klinika.cz | Telefon: +420 123 456 789

Zdroj: vlastní zpracování

SchuzkaDeleteView umožňuje odstranění schůzky. Uživatel je před odstraněním požádán o potvrzení svého úmyslu na stránce `schuzka_confirm_delete.html`. Po potvrzení

je schůzka odstraněna a uživatel je přesměrován na seznam schůzek, kde již daná schůzka nebude figurovat.

Obrázek 15. SchuzkaDeleteView



Zdroj: vlastní zpracování

SchuzkaCreateView umožňuje vytvoření nové schůzky prostřednictvím formuláře na stránce form.html. Formulář zahrnuje pole pro zadání data, času schůzky, účelu a

pacienta. Po úspěšném uložení schůzky je uživatel přesměrován zpět na seznam schůzek, což umožňuje okamžitou kontrolu vložených dat.

Obrázek 16. SchuzkaCreateView

Vytvořit novou schůzku

Datum a čas:

Účel schůzky:

Pacient:

Poznámky:

© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.

Zdroj: vlastní zpracování


SchuzkaUpdateView poskytuje formulář pro úpravu existující schůzky. Uživatel může upravit datum, čas, účel a pacienta schůzky. Tato třída zajišťuje, že všechny změny

jsou reflektovány v systému po uložení. Po úspěšné úpravě je uživatel opět přesměrován na seznam schůzek.

Obrázek 17. SchuzkaUpdateView

Aktualizovat Schůzku

Datum schůzky:

Účel schůzky:

Poznámky:

Pacient:

Aktualizovat

© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.

Kontaktujte nás: info@klinika.cz | Telefon: +420 123 456 789

Zdroj: vlastní zpracování

Celkově tyto třídy poskytují uživatelům CRM systému efektivní nástroje pro správu schůzek, od jejich vytváření a úprav až po odstranění. Zajišťují plynulou a uživatelsky přívětivou práci se schůzkami, což je klíčové pro efektivní organizaci práce v zdravotnickém zařízení.

V třídách pro lékařský záznam je pozornost věnována především částí CRM, která zahrnuje zpracování detailních lékařských záznamů pacientů. Následuje podrobný popis toho, co by mohla každá třída zahrnovat:

LekarskyZaznamListView

Obrázek 18. LekarskyZaznamListView

Pacient	Diagnóza	Datum schůzky	Předpis	Lékařské poznámky
Jiří Novák	Chronická bronchitida	30.03.2023 10:00	Použití inhalátorů a odpočinek	Pacient má chronické respirační potíže.
Kateřina Svobodová	Akutní otitis media	05.04.2023 14:30	None	Další kontrola za 14 dní.
Lukáš Dvořák	Hypertenze	15.04.2023 09:00	Antihypertenziva, omezení soli v dietě	Je třeba sledovat krevní tlak pravidelně.
Jiří Novák	Hypertenze	30.03.2023 10:00	Antihypertenziva	Regulární kontrola krevního tlaku.
Kateřina Svobodová	Artritida	05.04.2023 14:30	Protizánětlivé léky	Pravidelné kontroly a fyzikální terapie.
Lukáš Dvořák	Ekzém	15.04.2023 09:00	Steroidní krémy	Kontrolovat a hydratovat postižené oblasti.
Petr Kovář	Migréna	15.03.2024 10:00	Triptany	Vedení deníku migrén a vyhýbání se spouštěčům.
Anna Bílá	Diabetes mellitus	20.03.2024 14:00	Inzulín	Pravidelné monitorování hladiny glukózy v krvi.

Zdroj: vlastní zpracování

Tento pohled prezentuje seznam všech lékařských záznamů v systému. Mohl by být rozšířen o filtry a vyhledávací funkce, které umožní uživatelům rychle najít záznamy podle kritérií, jako je jméno pacienta, datum diagnózy nebo specifické podmínky. Implementace stránkování také zlepšit uživatelskou zkušenost při práci s velkým počtem záznamů.

LekarskyZaznamDetailView

Detailní pohled na konkrétní lékařský záznam by měl poskytnout komplexní informace, včetně všech podrobností o diagnóze pacienta, předepsané léčbě a jakýchkoli poznámkách k následné péči. Mohl by také zobrazovat související záznamy, jako jsou minulé schůzky nebo testy související s diagnózou, aby poskytl úplný přehled o lékařské historii.

Obrázek 19. LekarskyZaznamDetailView

Detail Lékařského Záznamu

Pacient: Jan Novák

Věk: 34 let

Pohlaví: Muž

Kontaktní číslo: +420 123 456 789

Adresa: Praha 8, Černínova 1234/5, 182 00

Datum schůzky: 15. dubna 2024

Čas schůzky: 10:30

Diagnóza: Akutní bronchitida

Detaily diagnózy: Pacient trpí kašlem, horečkou a dušností. Výsledky rentgenového snímku hrudníku naznačují zánět průdušek.

Předpis: Antibiotika (Amoxicilin 500mg třikrát denně po dobu 7 dní), odpočinek, dostatečný příjem tekutin.

Doporučený režim: Vyhnout se kouření a znečištěnému ovzduší. Při problémech s dýcháním okamžitě kontaktovat lékaře.

Lékařské poznámky: Pacient byl odborně vyšetřen. Další kontrola za 2 týdny. Pokud symptomy přetrvávají nebo dojde k zhoršení stavu, bude potřeba další vyšetření, včetně možné hospitalizace.

Lékař: MUDr. Eva Hrušková

Kontakt na lékaře: +420 987 654 321

© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.

Zdroj: vlastní zpracování

LekarskyZaznamCreateView

Při vytváření nového lékařského záznamu, kromě základních polí formuláře, je vhodné zvážit začlenění dynamických prvků, jako jsou rozbalovací seznamy pro diagnózy naplněné na základě nejběžnějších podmínek léčených ve vašem zařízení. Lze také zahrnout možnost připojit soubory, jako jsou výsledky testů nebo digitální obrázky.

Obrázek 20. LekarskyZaznamCreateView

Vytvoření Nového Lékařského Záznamu

Pacient:

Jan Novák

Datum Schůzky:

mm/dd/yyyy

Diagnóza:

Předpis:

Lékařské Poznámky:

Vytvořit Záznam

© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.

Zdroj: vlastní zpracování

LekarskyZaznamUpdateView

Pro aktualizaci lékařských záznamů je třeba zajistit, aby formulář byl vyplněn stávajícími daty. Je také zásadní logovat změny pro účely auditu, kde se zaznamená, kdo a kdy provedl aktualizaci. To může být důležité z právních důvodů a pro udržení integrity lékařského záznamu.

Obrázek 21. LekarskyZaznamUpdateView

Aktualizovat Lékařský Záznam

Pacient:

Datum schůzky:

Diagnóza:

Předpis:

Lékařské poznámky:

Aktualizovat

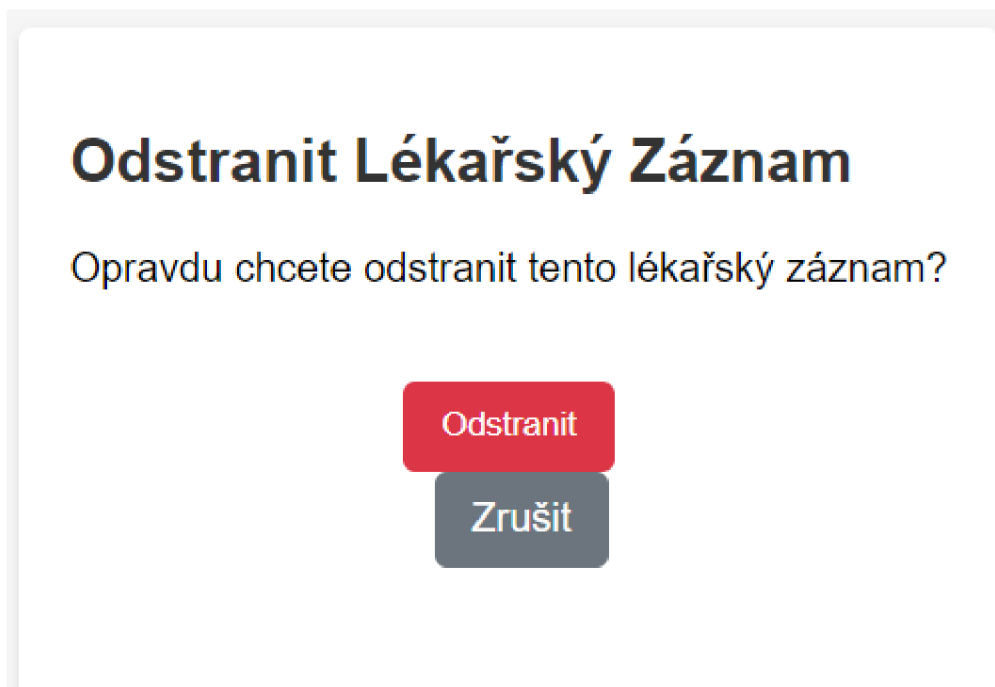
4 Název kliniky nebo CRM systému. Všechna práva vyhrazena. Kontaktujte nás: info@klinika.cz | Telefon: +420 123 456 78

Zdroj: vlastní zpracování

LekarskyZaznamDeleteView

Mazání lékařských záznamů by mělo být řešeno s nejvyšší opatrností. Možná nebude nutné odstranit záznamy z databáze, lze je označit jako neaktivní nebo archivované. Tento přístup udržuje data pro historické účely a vyhovuje mnoha zdravotnickým předpisům, které vyžadují uchovávání lékařských záznamů po určitou dobu.

Obrázek 22. LekarskyZaznamDeleteView



Zdroj: vlastní zpracování

Kontrola přístupu: Je vhodné zajistit, aby k lékařským záznamům měli přístup pouze oprávnění zaměstnanci, což může představovat řízení přístupu na základě rolí.

Auditní stopa: Je dobré uchovávat protokol o všech akcích prováděných v lékařských záznamech, včetně toho, kdo k záznamu přistupoval nebo ho upravil a kdy, je to důležité pro dodržování předpisů a pro bezpečnost.

Integrace: Je nutné zvážit, jak by tyto záznamy mohly být integrovány s dalšími částmi CRM, jako jsou schůzky, fakturace a komunikace s pacienty. Například aktualizace lékařského záznamu by automaticky mohla upozornit pacienta prostřednictvím e-mailu nebo SMS, pokud je potřeba další schůzka.

Bylo provedeno rozsáhlé využití systému šablon frameworku Django pro dynamické generování webového obsahu. Jedním z hlavních prvků této práce bylo efektivní využití konceptu základní šablony v Django, což je HTML soubor sloužící jako univerzální vzor pro ostatní šablony v aplikaci.

Základní šablona obsahuje všechny společné prvky webové stránky, jako jsou hlavička, navigace a odkazy na CSS/JS soubory. Tímto přístupem bylo zajištěno konzistentní rozložení a stylizace napříč všemi stránkami aplikace, čímž se výrazně usnadnila údržba a aktualizace projektu.

Pro vkládání specifického obsahu do jednotlivých stránek byly vytvořeny tzv. dětské šablony, které dědí základní rozložení a styl ze základní rodičovské šablony pomocí Django šablonovacího tagu `{% extends 'base.html' %}`. Díky tomuto mechanismu dědičnosti bylo možné přizpůsobit každou stránku s jedinečným obsahem bez nutnosti opakovaně psát kód pro společné prvky.

V základní šabloně byly definovány důležité komponenty:

Hlavička s navigačním menu poskytující odkazy na hlavní sekce aplikace.

Hlavní obsahový blok, kde byl pomocí tagů `{% block content %}``{% endblock %}` umožněn vklad specifického obsahu pro každou stránku.

Patička obsahující základní informace o projektu a kontakty.

Odkazy na styly a skripty pro zajištění konzistentního vizuálního stylu a funkcionalit.

Tento přístup umožnil nejen zajistit vizuální a strukturální konzistenci webové aplikace, ale také zefektivnit proces vývoje tím, že bylo eliminováno zbytečné opakování kódu. Implementace základní šablony a systému dědičnosti šablon v Django frameworku tak představovala základ pro efektivní a flexibilní vývoj webové aplikace v rámci diplomové práce.

4.12 Mikroslužby a CSS

V rámci této diplomové práce byl představen koncept, který má sice některé charakteristiky mikroslužeb, ale neplní je všechny dle klasické definice. I když mikroslužby typicky zahrnují komunikaci mezi nezávisle fungujícími částmi aplikace pomocí HTTP nebo API, v mé realizaci bylo vytvořeno samostatné Django aplikace, která slouží specifickému účelu - registraci a autorizaci uživatelů, avšak bez použití těchto lehkých protokolů pro komunikaci.

Namísto toho, abych sledoval striktní mikroslužební architekturu, jsem se rozhodl vyvinout samostatnou aplikaci, která je sice nezávislá a může být snadno přesunuta mezi projekty, ale nekomunikuje přímo přes API s hlavním CRM systémem. Tento přístup byl zvolen z důvodu zjednodušení vývoje a zvýšení efektivity práce na projektu.

Implementace mikroslužby jako samostatné aplikace pro autorizaci nebo registraci uživatelů

Obrázek 23. Autorizace nebo registrace uživatelů

Vítejte na naší stránce

Přihlášení Registrace

© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.

Zdroj: vlastní zpracování

Obrázek 24. Přihlášení

Obnovit heslo' (Forgot your password? [Reset password](#))."/>

Přihlášení

Přihlášení

Přihlášení

Uživatelské jméno:

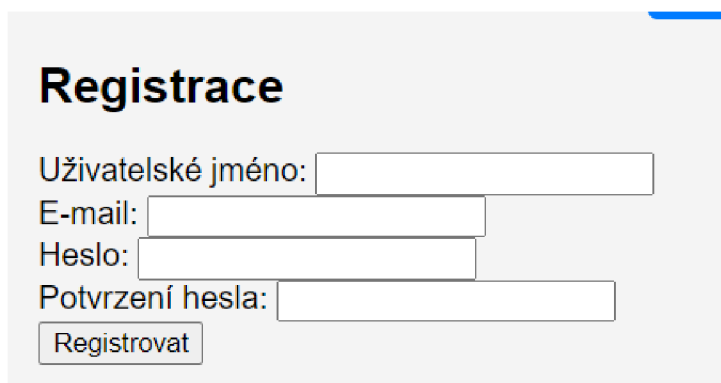
Heslo:

Přihlásit se

Zapomněli jste heslo? [Obnovit heslo](#)

Zdroj: vlastní zpracování

Obrázek 25. Registrace



Registrace

Uživatelské jméno:

E-mail:

Heslo:

Potvrzení hesla:

Zdroj: vlastní zpracování

Takto navržené řešení přináší určité výhody, jako je snadnější testování a ladění, jelikož každá aplikace může být vyvíjena a testována nezávisle. Navíc, toto řešení umožňuje rychlou integraci do různých projektů bez nutnosti přizpůsobování API, což by bylo ideální v případě, že by byla potřeba aplikaci rychle přizpůsobit a nasadit v jiném kontextu.

Ačkoliv toto řešení nemůže být považováno za mikroslužbu v pravém smyslu slova, poskytuje cenné lekce o flexibilitě a modularitě ve vývoji software. To znamená, že i bez striktního dodržování mikroslužební architektury je možné vytvářet efektivní a dobře strukturované systémy, které jsou přizpůsobitelné a škálovatelné.

Důraz na vytvoření čistého a efektivního CSS kódu, který podporuje strukturovanost, konzistenci a přístupnost celé webové aplikace. Díky promyšlenému návrhu a vhodné implementaci bylo dosaženo řady pozitivních výsledků, které zlepšily uživatelskou zkušenost a vizuální atraktivitu projektu.

Důležitým rozhodnutím bylo využití společných stylů pro elementy jako navigační menu, tabulky a patičky, což zajišťovalo vizuální konzistenci napříč různými částmi aplikace. Použití flexboxu pro navigační menu nejen že umožnilo flexibilní a responzivní rozložení, ale také zjednodušilo adaptaci rozhraní na různé velikosti obrazovek. Zároveň jednoduchost a intuitivní design vizuálních prvků napomáhaly uživatelské přístupnosti, což je v moderním webdesignu nezbytné.

Přechody a hover efekty, aplikované na odkazy v navigaci, přidaly na dynamice uživatelskému rozhraní a zvýšily interaktivitu aplikace. Tyto prvky nejen, že oživily design, ale také zlepšily celkový dojem z uživatelské interakce s aplikací.

Nicméně, přestože bylo dosaženo mnoha pozitiv, v projektu se objevily určité problémy. Omezení v komplexnosti a adaptabilitě současného CSS kódu může ztížit implementaci složitějších designových požadavků nebo dynamických uživatelských interakcí, které vyžadují pokročilé techniky. Ačkoli bylo použití flexboxu pro navigační menu krokem správným směrem, celková responzivita designu by mohla být lepší, což je důležité pro zajištění optimálního zobrazení na různých zařízeních.

Z hlediska údržby a škálovatelnosti by současná struktura CSS mohla být v případě rozšíření projektu nedostatečná. Výzvou je zajistit, aby CSS kód zůstal spravovatelný a snadno rozšiřitelný i při přidávání nových funkcí nebo vizuálních prvků.

Naštěstí však existuje řada možností pro budoucí rozšíření a zlepšení. Přidání responzivních stylových prvků a mediálních dotazů by mohlo zlepšit adaptabilitu na různé velikosti displejů. Implementace CSS preprocesorů, jako je SASS nebo LESS, by mohla usnadnit údržbu kódu a zlepšit jeho škálovatelnost. Rozšíření vizuálních efektů a animací by mohlo přinést ještě působivější uživatelské rozhraní a zavedení CSS frameworků, jako je Bootstrap nebo Foundation, by mohlo poskytnout robustní a testované řešení pro zajištění responzivity a usnadnit vývoj.

5 Výsledky

V rámci procesu nasazení webové aplikace bude provedena instalace systému pro správu verzí Git na server. Tento krok je zásadní pro efektivní správu a aktualizaci kódu aplikace, umožňující plynulou synchronizaci změn mezi vývojovým prostředím a produkčním prostředím. Instalace Gitu poskytne základ pro verzování kódu a jeho bezpečné ukládání, což je klíčové pro správu projektu a jeho další rozvoj.

Po úspěšné instalaci Gitu na server se přistoupí k klonování repozitáře s projektem přímo na server. Tento krok zajišťuje, že na serveru bude k dispozici nejnovější verze kódu aplikace. Klonování projektu umožní snadný přístup k veškerému kódu a zdrojům na serveru, což je nezbytné pro spuštění a provoz webové aplikace v produkčním prostředí. Klonování repozitáře na server také usnadňuje další aktualizace a údržbu aplikace, neboť všechny změny v kódu mohou být snadno synchronizovány prostřednictvím příkazů Git.

Tyto počáteční kroky vytvářejí základ pro bezpečné a efektivní nasazení a správu webové aplikace, přičemž kladený důraz na verzování a správu kódu umožňuje plynulý vývoj a nasazení aplikace.

Po klonování projektu na server bude vytvořeno virtuální prostředí, což je nezbytný krok k izolaci závislostí projektu od jiných aplikací, které běží na stejném serveru. Toto virtuální prostředí poskytne izolovaný prostor pro aplikaci, ve kterém mohou být instalovány a spravovány specifické knihovny a závislosti potřebné pro projekt bez ovlivnění ostatních aplikací na serveru. Vytvoření virtuálního prostředí tak přispěje k stabilitě a bezpečnosti celého serveru tím, že zabraňuje konfliktům mezi závislostmi různých aplikací.

Dalsi procesy:

Instalace závislostí - je nutné nainstalovat všechny další závislosti potřebné pro správnou funkci webu. Tyto závislosti jsou obvykle definovány v souboru requirements.txt.

Vytvoření databáze - systém bude využívat databázi, kterou je třeba vytvořit a následně propojit s Django aplikací.

Přenos dat - pokud je to potřeba, lze přenést všechny tabulky a data z vývojového prostředí do produkčního.

Následně bude aplikace nasazena s využitím Gunicornu, což je WSGI server sloužící k spuštění Django aplikace. Gunicorn funguje jako intermediář mezi webovou aplikací a webovým serverem Nginx, zajišťující efektivní a bezpečnou komunikaci mezi

nimi. Použití Gunicornu jako WSGI serveru umožňuje lepší správu pracovních procesů a žádostí, což zvyšuje schopnost aplikace zvládat větší počet současných požadavků. Tímto způsobem je aplikace připravena na provoz v produkčním prostředí, kde Nginx přijímá příchozí HTTP požadavky a přeposílá je na Gunicorn, který následně zpracovává logiku aplikace a vrátí odpověď zpět uživatelům.

Tento proces nejen zajišťuje optimální výkon a dostupnost webové aplikace, ale také posiluje její bezpečnost tím, že odděluje veřejně přístupný webový server od samotného běhu aplikace.

Po úspěšném nasazení aplikace s využitím Gunicornu bude provedena konfigurace Nginx jako proxy serveru. Nginx zde přebírá roli brány, která přijímá veškeré požadavky z internetu a efektivně je přeposílá na Gunicorn. Tím se zajišťuje, že veškerá komunikace s aplikací je centralizovaně spravována a optimalizována. Kromě zvýšení výkonu a schopnosti aplikace reagovat na vyšší počet požadavků Nginx také poskytuje další vrstvu zabezpečení tím, že skrývá interní strukturu síťové infrastruktury a detaily aplikace před veřejným internetem. Nginx navíc umožňuje implementaci dodatečných bezpečnostních pravidel, jako je omezení přístupu, šifrování SSL/TLS a ochrana proti DDoS útokům.

V rámci údržby a aktualizace aplikace bude pravidelně využíván příkaz `git pull`, který umožňuje stáhnout nejnovější změny kódu z repozitáře na server. Tento postup zajišťuje, že aplikace na produkčním serveru vždy reflektuje aktuální stav vývojového repozitáře. Před prováděním aktualizace je doporučeno aplikaci dočasně zastavit, aby se předešlo potenciálním konfliktům nebo nestabilitě během aktualizacího procesu. Po úspěšném dokončení aktualizace je aplikace opět spuštěna, čímž je zaručena její nejnovější verze v provozu bez přerušení služby.

Tento proces aktualizace zajišťuje, že aplikace je neustále v souladu s posledními vývojovými změnami a zabezpečením, což je důležité pro dlouhodobou udržitelnost a spolehlivost webového projektu.

Po konfiguraci Nginx jako proxy serveru a aktualizaci aplikace se přikročí k posílení bezpečnosti a implementaci mechanismů pro load balancing. Tyto kroky jsou zásadní pro ochranu aplikace před potenciálními hrozbami a zajištění její schopnosti efektivně rozdělovat příchozí požadavky mezi více instancí. V rámci zabezpečení bude nastaveno šifrování spojení pomocí SSL/TLS certifikátů, které zajistí bezpečnou komunikaci mezi klientem a serverem. Dále bude implementován systém pro load

balancing, který umožní distribuci zátěže mezi více instancemi aplikace, čímž se zvýší její dostupnost a odolnost vůči vysokému provozu.

Pro dlouhodobou stabilitu a výkon aplikace je klíčové pravidelné monitorování a údržba. Monitorování zahrnuje sledování výkonu aplikace, spotřeby zdrojů a detekci možných bezpečnostních hrozeb. Na základě získaných dat se provádí údržba, která zahrnuje aktualizace závislostí, zabezpečení a optimalizaci konfigurace pro zajištění optimálního výkonu. Důležitým aspektem je také pravidelné testování zabezpečení, aby se předešlo potenciálním zranitelnostem a udržela vysoká úroveň ochrany dat a soukromí uživatelů.

Tato fáze zabezpečení, load balancingu, monitorování a údržby, představuje nepřetržitý proces, který vyžaduje aktivní zapojení a správu ze strany vývojářů a správců systému. Cílem je nejen zajistit hladký a bezpečný provoz aplikace, ale také připravit ji na budoucí rozšíření a růst.

6 Závěr

V této diplomové práci byl podrobně rozebrán proces od počáteční analýzy, přes výběr a přizpůsobení vhodného CRM systému, až po jeho pilotní implementaci. Je možné zdůraznit, že i když navržený CRM systém nemusí okamžitě překonat ostatní systémy na trhu svými charakteristikami, ale má potenciál vyhrát na delší trati. Jeho rychlá implementace, snadná škálovatelnost a schopnost poskytnout všechny potřebné funkce představují zásadní výhody. Hlavním přínosem je však jeho cenová dostupnost ve srovnání s ostatními alternativami na trhu.

Tento systém byl navržen s myšlenkou na dlouhodobou udržitelnost a růst, což znamená, že i když jeho počáteční výkon může být srovnatelný s konkurenty, jeho pružnost a nízké náklady na vlastnictví jej činí ideálním řešením pro kliniky hledající cenově efektivní a budoucností orientované CRM řešení. Vzhledem k těmto aspektům má systém potenciál poskytnout vyšší návratnost investic a stát se preferovanou volbou pro zdravotnická zařízení, která chtějí optimalizovat své operace a zároveň kontrolovat náklady.

Vývoj vlastního CRM systému byl proveden s ohledem na specifické požadavky a procesy, což umožnilo dosáhnout vysoké úrovně personalizace a zároveň funkcionalitu přesně šitou na míru potřebám uživatelů.

Celkem diplomová práce ukazuje, že pečlivě navržený a implementovaný CRM systém může mít významný pozitivní dopad na zvýšení efektivity, kvality služeb a spokojenosti pacientů v medicínském zařízení. Přínosy zavedeného systému jsou důležitým krokem k modernizaci a inovaci v oblasti zdravotní péče, což umožňuje lépe reagovat na dynamické požadavky a výzvy současného zdravotnického prostředí. Z hlediska udržitelnosti a rozvoje je rovněž důležité, aby systém CRM podporoval integraci s jinými zdravotnickými systémy a technologiemi, čímž se zvýší jeho hodnota a využitelnost. To zahrnuje možnost sdílení a analýzy dat mezi různými zdravotnickými zařízeními, což přispívá k lepší koordinaci péče a zlepšení výsledků pro pacienty.

V závěru lze tedy říci, že realizace projektu CRM systému pro medicínské zařízení byla úspěšná a otevírá cestu k dalším inovacím a zlepšením v oblasti zdravotní péče. Důležité je udržet dynamický přístup k rozvoji systému, aby bylo možné efektivně reagovat na nové výzvy a příležitosti v rychle se vyvíjícím zdravotnickém sektoru.

Seznam použitých zdrojů

AGEL. *Skupina AGEL je nejúspěšnějším poskytovatelem zdravotní péče ve střední Evropě* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.agel.cz/index.html>

ÅNGSTRÖM, R. et al. Getting AI Implementation Right: Insights from a Global Survey. *Calif. Manag. Rev.* 2023, 66(1): 1-18

ANSHARI, M. ALMUNAWAR, M. Evaluating CRM implementation in healthcare organization. *Software Engineering*. 2012, arXiv:1204.3689

BAASHAR, Y., ALHUSSIAN, H. et al. Customer relationship management systems (CRMS) in the healthcare environment: A systematic literature review. *Computer standards & interfaces*. 2020, 71, 103442.

BENZ, G., PADDISON, N. Developing patient-based marketing strategies. *Healthc. Execut.* 2004, 19(5): 40-42

BUGHIN, J. et al. Notes from the AI frontier: Modeling the impact of AI on the world economy. *McKinsey Global Institute*, 2018: 1-61

BUTTLE, F., MAKLAN, S. *Customer Relationship Management: Concepts and Technologies*. 3rd ed. London: Routledge, 2015. 426 p. ISBN 978-1138236813

CARD, S., MACKINLAY, J. SHNEIDERMAN, B. *Readings in information visualization: using vision to think*. San Francisco, Calif.: Morgan Kaufmann Publishers, 1999. ISBN 978-1-55860-533-6

CGM. *Komplexní péče pro Vaši ordinaci*. [online] 2024 [cit. 2024-02-14] Dostupné z:

https://www.cgm.com/cze_cz?utm_source=seznam&utm_medium=cpc&utm_campaign=S_A_Brand_CGM&utm_content=CGM+compugroup+medical&utm_term=compugroup+medical+cr&sznaiid=10997570363405639488&sznclid=RC0geXV0cHd9d3x3dX19dn13d3V2fXx1ODB5dXN0fXd8dHRzfGp9cnA4MCF5dXN0fXN2cn1xfGp3fXM4J3lxcQd1Bn0GfQJ8dQACcXEHBXEAcHR3AHJ9cQF8Bn0BdQ

CHOUDHURY, N. World Wide Web and Its Journey from Web 1.0 to Web 4.0. *International Journal of Computer Science and Information Technologies*, 2014, 5(6): 8096-8100.

DATALAN. *Profil spoločnosti* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.datalan.sk/o-nas/>

- DWIVEDI, Y., WANG, Y. Artificial intelligence for B2B marketing: challenges and opportunities. *Ind. Mark. Manag.* 2022, 06: 105
- DYCHE, J. *The CRM Handbook: A Business Guide to Customer Relationship Management*. Boston: Addison-Wesley Professional, 2002. 307 p. ISBN 9780201730623
- FOTR, J., ŠVECOVÁ, L., a kol. *Manažérske rozhodování. Postupy, metody a nástroje*. Praha: Ekopress, 2016. 375 s. ISBN 978-80-87865-33-0
- GRANT, R., et al. Customer Relationship Management. *Management Journal*, 1998, 17: 3-6
- HUNG, S., HUNG, W., TSAI, C., JIANG, C. Critical factors of hospital adoption on CRM system: Organizational and information system perspectives. *Decis. Supp. Syst.* 2010, 48(4): 592-603
- HUSSEIN, A., ASHARAI, H., AZIZAH, A. The Core Components and Types of CRM. *Pakistan Journal of Humanities and Social Sciences*, 2019, 7(1): 121-145
- IRIANA, R., BUTTLE, F. Strategic, Operational, and Analytical Customer Relationship Management: Attributes and Measures. *Journal of Relationship Marketing*, 2006, 2006, 5: 23-42
- KALČEVOVÁ, J. *Studijní materiály - Kteriální matice a hodnocení variant* [online] 2008. [cit. 2024-02-01]. Dostupné z: <http://jana.kalcev.cz/vyuka/kestazeni/EKO422-KriterialniMatice.pdf>
- KEEGAN, B., CANHOTO, A., YEN, D. Power negotiation on the tango dancefloor: the adoption of AI in B2B marketing. *Ind. Mark. Manag.*, 2022, 100: 36-48
- LAU, F., KUZIEMSKY, C., PRICE, M., GARDNER, J. A review on systematic reviews of health information system studies. *J. Am. Med. Inf. Assoc.* 2010, 17(6): 637-645.
- LEDRO, C., NOSELLA, A., DALLA POZZA I. Integration of AI in CRM: Challenges and guidelines. *Journal of Open Innovation: Technology, Market, and Complexity*. 2023, 9(4)
- LEE, S. AI: A component based methodology for Web application development. *Journal of Systems and Software*, 2004, 71(1): 177-187.
- LUN, Z., JINLIN, L., YINGYING, W. Customer relationship management system framework design of Beijing Rural Commercial Bank. *Proceedings of the IEEE International Conference on Service Operations and Logistics, and Informatics*, 2008, 101: 97-101

MATEROVÁ, H. VRUBLOVÁ, Y. *Informatika a legislativa ve zdravotnictví*. Ostrava: Ostravská univerzita v Ostravě, 2013. 91 s. ISBN 978-80-7464-434-6.

MCNAUGHTON, B., RAY, P., LEWIS, L. Designing an evaluation framework for IT service management. *Information & Management*, 2010, 47(4): 219-225.

MEDAX. *MEDAX Systems s.r.o.* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.medax.cz/spolecnost.php>

MEDICALC. *Řešení pro moderní zdravotnictví* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.medicalc.cz/cs>

MEDICONAS. *Komplexní péče o vaše zdraví v medicon* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.mediconas.cz/cs>

MEDINFO. *Hospital Information Systems (HIS) in the Czech Republic* [online] 2024 [cit. 2024-02-14] Dostupné z: <http://www.medinfo.cz/index.php/en/areas/information-systems/113-his>

MISHRA, N., MUKHERJEE, S. Effect of artificial intelligence on customer relationship management of Amazon in Bangalore. *Int. J. Manag.*, 2019, 10 (4): 168-172

NAGAR, S. *Introduction to Python for engineers and scientists*. New York: Springer Science+Business Media, 2017. 352 p. ISBN 978-1-4842-3203-3

PERNA, A., BARALDI, E. *CRM Systems in industrial companies: Intra-and inter-organizational effects*. London: Springer. 2014. 262 p. ISBN 978-1-349-463176

PROSOFT. *About Prosoft* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.prosoft.sk/>

RUBIO, D. *Beginning Django: web application development and deployment with Python*. New York: Springer Science+Business Media, 2017. 385 p. ISBN 9781484227862

SKUPINA ICZ. *Přední dodavatel IT řešení* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.iczgroup.com/>

STAPRO. *Informace v ceně života* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.stapro.cz/spolecnost/>

STEINER. *Naše vyspělé informační systémy pomáhají zdravotníkům po celém světě* [online] 2024 [cit. 2024-02-14] Dostupné z: <https://www.steiner.cz/>

YINA, W. Proceedings of the 2010 Second International Conference on Multimedia and Information Technology (MMIT). *Application of customer relationship management in health care*. 2010, 1: 52-55

ZABLAH, A., BELLENGER, D., JOHNSTON, W. An evaluation of divergent perspectives on customer relationship management: towards a common understanding of an emerging phenomenon. *Industrial Marketing Management*, 2004, 33(6): 475-489

7 Seznam obrázků, tabulek, grafů a zkratk

7.1 Seznam obrázků

Obrázek 1. Integrace umělé inteligence do systémů CRM.....	29
Obrázek 2. Proces vizualizace	35
Obrázek 3. MVT	37
Obrázek 4. Databázová struktura CRM	41
Obrázek 5. Příklad možnosti vyhledávání a filtrování lékařských záznamů.....	59
Obrázek 6. Formuláře	60
Obrázek 7. Seznam služeb	61
Obrázek 8. Hlavní stránka systému	64
Obrázek 9. PacientListView	65
Obrázek 10. PacientCreateView	67
Obrázek 11. PacientUpdateView	68
Obrázek 12. PacientDeleteView	69
Obrázek 13. SchuzkaListView	70
Obrázek 14. SchuzkaDetailView	71
Obrázek 15. SchuzkaDeleteView	72
Obrázek 16. SchuzkaCreateView	73
Obrázek 17. SchuzkaUpdateView	74
Obrázek 18. LekarskyZaznamListView	75
Obrázek 19. LekarskyZaznamDetailView.....	76
Obrázek 20. LekarskyZaznamCreateView	77
Obrázek 21. LekarskyZaznamUpdateView	78
Obrázek 22. LekarskyZaznamDeleteView	79
Obrázek 23. Autorizace nebo registrace uživatelů	81
Obrázek 24. Přihlášení	81
Obrázek 25. Registrace	82

7.2 Seznam tabulek

Tabulka 1 Rozdíly mezi jednotlivými generacemi webových technologií.....	31
Tabulka 2 Hodnotící matice.....	44
Tabulka 3 Nemocniční informační systémy (HIS) v České republice	48
Tabulka 4 Výsledky hodnocení alternativ podle kritérií	53
Tabulka 5 Výsledky párového porovnání kritérií V	55
Tabulka 6 Váhy relativní důležitosti hodnotících kritérií	55
Tabulka 7 Výpočet zobecněného kritéria se zohledněním váhových koeficientů	56

7.3 Seznam použitých zkratk

API - nástroj a rozhraní pro programování aplikací

CRM (customer relationship management) - řízení vztahů se zákazníky

DL - hluboké učení

EHR - elektronické zdravotní záznamy

HIS – Hospital Information System

HIT - zdravotnické informační technologie

LIS - laboratorní informační systémy

ML - strojové učení

MVT - Model-View-Template

NIS - Nemocniční informační systémy

ORM - objektově-relační mapování

SQ - kvalita služeb

UI - umělá inteligence

Přílohy

Odkazovaný seznam příloh

Příloha 1 Create Script

```
CREATE TABLE "public"."Patients" (  
    "patient_id" SERIAL NOT NULL,  
    "first_name" VARCHAR(255) NOT NULL,  
    "last_name" VARCHAR(255) NOT NULL,  
    "dob" DATE NOT NULL,  
    "gender" CHAR(1) CHECK ("gender" IN ('M', 'F', 'O')) NOT NULL,  
    "contact_number" VARCHAR(20) NOT NULL,  
    "email" VARCHAR(255) NOT NULL,  
    "address" TEXT NOT NULL,  
    CONSTRAINT "Patients_pk" PRIMARY KEY ("patient_id")  
),  
CREATE TABLE "public"."Appointments" (  
    "appointment_id" SERIAL NOT NULL,  
    "patient_id" INT NOT NULL,  
    "appointment_date" DATE NOT NULL,  
    "purpose" TEXT NOT NULL,  
    "notes" TEXT,  
    CONSTRAINT "Appointments_pk" PRIMARY KEY ("appointment_id"),  
    CONSTRAINT "Appointments_fk0" FOREIGN KEY ("patient_id")  
REFERENCES "public"."Patients"("patient_id") ON DELETE CASCADE  
),  
CREATE TABLE "public"."Services" (  
    "service_id" BIGINT NOT NULL,  
    "service_name" VARCHAR(255) NOT NULL,  
    "description" TEXT NOT NULL,  
    "price" DECIMAL(10,2) NOT NULL,  
    "appointment_id" BIGINT NOT NULL,  
    "patient_id" BIGINT NOT NULL,  
    CONSTRAINT "Services_pk" PRIMARY KEY ("service_id"),  
    CONSTRAINT "Services_fk0" FOREIGN KEY ("appointment_id")  
REFERENCES "public"."Appointments"("appointment_id") ON DELETE CASCADE,
```

```

        CONSTRAINT "Services_fk1" FOREIGN KEY ("patient_id") REFERENCES
"public"."Patients"("patient_id") ON DELETE CASCADE
    ),
CREATE TABLE "public"."Medical_Records" (
    "record_id" SERIAL NOT NULL,
    "patient_id" INT NOT NULL,
    "appointment_id" INT NOT NULL,
    "diagnosis" TEXT NOT NULL,
    "prescription" TEXT,
    "medical_notes" TEXT,
    CONSTRAINT "Medical_Records_pk" PRIMARY KEY ("record_id"),
    CONSTRAINT "Medical_Records_fk0" FOREIGN KEY ("patient_id")
REFERENCES "public"."Patients"("patient_id") ON DELETE CASCADE,
    CONSTRAINT "Medical_Records_fk1" FOREIGN KEY ("appointment_id")
REFERENCES "public"."Appointments"("appointment_id") ON DELETE CASCADE
    ),

```

```

from django.contrib import admin

```

```

from .models import Pacient, Schuzka, Sluzba, LekarskyZaznam

```

```

# Inline třídy pro editaci souvisejících objektů přímo z administrační stránky

```

```

class SchuzkaInline(admin.TabularInline):

```

```

    model = Schuzka

```

```

    extra = 1 # Počet formulářů pro nové záznamy

```

```

    fields = ('datum_schuzky', 'ucel', 'notes') # Pole k zobrazení v inline

```

```

class SluzbaInline(admin.TabularInline):

```

```

    model = Sluzba

```

```

    extra = 1

```

```

    fields = ('nazev_sluzby', 'popis', 'cena') # Pole k zobrazení v inline

```

```

class LekarskyZaznamInline(admin.StackedInline):

```



```

model = LekarskyZaznam
extra = 1
fields = ('diagnosa', 'predpis', 'medical_notes') # Pole k zobrazení v inline

# Admin třída pro model Pacient s integrací inlines
class PacientAdmin(admin.ModelAdmin):
    list_display = ('jmeno', 'prijmeni', 'datum_narozeni', 'pohlavi') # Zobrazené pole v seznamu
    inlines = [SchuzkaInline, LekarskyZaznamInline]

# Admin třída pro model Schuzka s integrací inlines
class SchuzkaAdmin(admin.ModelAdmin):
    list_display = ('pacient', 'datum_schuzky', 'ucel') # Zobrazené pole v seznamu
    inlines = [SluzbaInline, LekarskyZaznamInline]

# Admin třída pro model Sluzba
class SluzbaAdmin(admin.ModelAdmin):
    list_display = ('schuzka', 'nazev_sluzby', 'cena') # Zobrazené pole v seznamu

# Admin třída pro model LekarskyZaznam
class LekarskyZaznamAdmin(admin.ModelAdmin):
    list_display = ('pacient', 'schuzka', 'diagnosa') # Zobrazené pole v seznamu

admin.site.register(Pacient, PacientAdmin)
admin.site.register(Schuzka, SchuzkaAdmin)
admin.site.register(Sluzba, SluzbaAdmin)
admin.site.register(LekarskyZaznam, LekarskyZaznamAdmin)

```

Příloha 2 Soubor forms.py

```
from django import forms

from .models import Pacient, Schuzka, Sluzba, LekarskyZaznam

class PacientForm(forms.ModelForm):

    class Meta:

        model = Pacient

        fields = ['jmeno', 'prijmeni', 'datum_narozeni', 'pohlavi', 'kontaktni_cislo', 'email',
'adresa']

        widgets = {

            'datum_narozeni': forms.DateInput(attrs={'type': 'date'}),

        }

class SchuzkaForm(forms.ModelForm):

    class Meta:

        model = Schuzka

        fields = ['pacient', 'datum_schuzky', 'ucel', 'notes']

        widgets = {

            'datum_schuzky': forms.DateTimeInput(attrs={'type': 'datetime-local'}),

        }

class SluzbaForm(forms.ModelForm):

    class Meta:

        model = Sluzba

        fields = ['nazev_sluzby', 'popis', 'cena', 'schuzka']
```

```
widgets = {  
    'cena': forms.NumberInput(attrs={'step': '0.01'}),  
}
```

```
class LekarskyZaznamForm(forms.ModelForm):
```

```
    class Meta:
```

```
        model = LekarskyZaznam
```

```
        fields = ['pacient', 'schuzka', 'diagnosa', 'predpis', 'medical_notes']
```

Příloha 3 Models.py

```
from django.db import models
```

```
class Pacient(models.Model):
```

```
    jmeno = models.CharField(max_length=255, verbose_name="Jméno")
```

```
    prijmeni = models.CharField(max_length=255, verbose_name="Příjmení")
```

```
    datum_narozeni = models.DateField(verbose_name="Datum narození")
```

```
    pohlavi = models.CharField(max_length=1, choices=(('M', 'Muž'), ('Ž', 'Žena'), ('O',  
'Ostatní')), verbose_name="Pohlaví")
```

```
    kontaktni_cislo = models.CharField(max_length=20, verbose_name="Kontaktní číslo")
```

```
    email = models.EmailField(verbose_name="Email")
```

```
    adresa = models.TextField(verbose_name="Adresa")
```

```
    class Meta:
```

```
        verbose_name = "Pacient"
```

```
        verbose_name_plural = "Pacienti"
```

```

def __str__(self):
    return f"{self.jmeno} {self.prijmeni}"

class Schuzka(models.Model):
    pacient = models.ForeignKey(Pacient, on_delete=models.CASCADE,
verbose_name="Pacient")

    datum_schuzky = models.DateTimeField(verbose_name="Datum schůzky")
    ucel = models.TextField(verbose_name="Účel schůzky")
    notes = models.TextField(blank=True, null=True, verbose_name="Poznámky")

    class Meta:
        verbose_name = "Schůzka"
        verbose_name_plural = "Schůzky"

    def __str__(self):
        return f"Schůzka pro {self.pacient} dne {self.datum_schuzky.strftime('%Y-%m-%d
%H:%M')}"

class Sluzba(models.Model):
    schuzka = models.ForeignKey(Schuzka, on_delete=models.CASCADE,
verbose_name="Schůzka")

    nazev_sluzby = models.CharField(max_length=255, verbose_name="Název služby")
    popis = models.TextField(verbose_name="Popis")
    cena = models.DecimalField(max_digits=10, decimal_places=2, verbose_name="Cena")

```

```

class Meta:
    verbose_name = "Služba"
    verbose_name_plural = "Služby"

def __str__(self):
    return self.nazev_sluzby

class LekarskyZaznam(models.Model):
    pacient = models.ForeignKey(Pacient, on_delete=models.CASCADE,
    verbose_name="Pacient")
    schuzka = models.ForeignKey(Schuzka, on_delete=models.CASCADE,
    verbose_name="Schůzka")
    diagnosa = models.TextField(verbose_name="Diagnóza")
    predpis = models.TextField(blank=True, null=True, verbose_name="Předpis")
    medical_notes = models.TextField(blank=True, null=True, verbose_name="Lékařské
    poznámky")

class Meta:
    verbose_name = "Lékařský záznam"
    verbose_name_plural = "Lékařské záznamy"

def __str__(self):
    return f"Lékařský záznam pro {self.pacient} ze dne
    {self.schuzka.datum_schuzky.strftime('%Y-%m-%d %H:%M')}}"

```

Příloha 4 Urls.py

```
"""
URL configuration for Medical_CRM project.

The `urlpatterns` list routes URLs to views. For more information please see:
    https://docs.djangoproject.com/en/5.0/topics/http/urls/
"""
from django.contrib import admin
from django.urls import path
from CRM import views
from CRM.views import *

urlpatterns = [
    path('admin/', admin.site.urls),

    # Hlavní stránka systému
    path("", views.BaseView.as_view(), name='home'),
    path('privacy-policy/', PrivacyPolicyView.as_view(), name='privacy_policy'),
    path('terms-of-service/', TermsOfServiceView.as_view(), name='terms_of_service'),

    # Seznam pacientů
    path('pacienti/', views.PatientListView.as_view(), name='pacienti_list'),
    # Detail pacienta
    path('patient/<int:pk>/', views.PatientDetailView.as_view(), name='patient_detail'),
    # Přidání nového pacienta
    path('patient/add/', views.PatientCreateView.as_view(), name='patient_add'),
    # Úprava existujícího pacienta
    path('patient/edit/<int:pk>/', views.PatientUpdateView.as_view(), name='patient_edit'),
    # Odstranění pacienta
    path('patient/delete/<int:pk>/', views.PatientDeleteView.as_view(),
name='patient_delete'),
    # Seznam schůzek
    path('schuzky/', views.SchuzkaListView.as_view(), name='schuzky_list'),
    # Detail schůzky
    path('schuzka/<int:pk>/', views.SchuzkaDetailView.as_view(), name='schuzka_detail'),
    # Přidání nové schůzky
    path('schuzka/add/', views.SchuzkaCreateView.as_view(), name='schuzka_add'),
    # Úprava existující schůzky
    path('schuzka/edit/<int:pk>/', views.SchuzkaUpdateView.as_view(),
name='schuzka_edit'),
    # Odstranění schůzky
    path('schuzka/delete/<int:pk>/', views.SchuzkaDeleteView.as_view(),
name='schuzka_delete'),
    # Seznam služeb
    path('sluzby/', views.SluzbaListView.as_view(), name='sluzby_list'),
    # Detail služby
    path('sluzba/<int:pk>/', views.SluzbaDetailView.as_view(), name='sluzba_detail'),
```

```

# Přidání nové služby
path('sluzba/add/', views.SluzbaCreateView.as_view(), name='sluzba_add'),
# Úprava existující služby
path('sluzba/edit/<int:pk>', views.SluzbaUpdateView.as_view(), name='sluzba_edit'),
# Odstranění služby
path('sluzba/delete/<int:pk>', views.SluzbaDeleteView.as_view(),
name='sluzba_delete'),
# Seznam lékařských záznamů
path('zaznamy/', views.LekarskyZaznamListView.as_view(), name='zaznamy_list'),
# Detail lékařského záznamu
path('zaznam/<int:pk>', views.LekarskyZaznamDetailView.as_view(),
name='zaznam_detail'),
# Přidání nového lékařského záznamu
path('zaznam/add/', views.LekarskyZaznamCreateView.as_view(),
name='zaznam_add'),
# Úprava existujícího lékařského záznamu
path('zaznam/edit/<int:pk>', views.LekarskyZaznamUpdateView.as_view(),
name='zaznam_edit'),
# Odstranění lékařského záznamu
path('zaznam/delete/<int:pk>', views.LekarskyZaznamDeleteView.as_view(),
name='zaznam_delete'),
]

```

Příloha 5 Views.py

```

class BaseView(TemplateView):
    """
    Základní zobrazení obsahující společnou logiku pro další zobrazení.
    """
    PROJECT_ROOT = =
    os.path.abspath(os.path.join(os.path.dirname(os.path.abspath(__file__)), '../..'))
    print(PROJECT_ROOT)
    template_name = 'home.html'
    extra_context = {'title': 'Můj lékařský CRM'}
    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        # Přidání extra kontextu do každého zobrazení
        context.update(self.extra_context)
        return context

```

```

class PrivacyPolicyView(TemplateView):
    template_name = 'privacy_policy.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['title'] = 'Politika ochrany osobních údajů'
        return context

```

```

class TermsOfServiceView(TemplateView):
    template_name = 'terms_of_service.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context['title'] = 'Podmínky používání'
        return context

```

```

class PacientDetailView(DetailView):
    """
    Detailní zobrazení informací o pacientovi.
    """
    model = Pacient
    template_name = 'crm/pacient_detail.html'

    def get_context_data(self, **kwargs):
        """
        Přidává do kontextu šablony informace o souvisejících schůzkách, službách a
lékařských záznamech.
        """
        context = super().get_context_data(**kwargs)
        # Získáváme ID pacienta z URL
        pacient_id = self.kwargs.get('pk')

```



```

# Přidáváme do kontextu seznam schůzek pro daného pacienta
context['schuzky_list'] = Schuzka.objects.filter(pacient_id=pacient_id)
# Přidáváme do kontextu seznam služeb spojených se schůzkami pacienta
context['sluzby_list'] = Sluzba.objects.filter(schuzka__pacient_id=pacient_id)
# Přidáváme do kontextu seznam lékařských záznamů pro daného pacienta
context['lekarske_zaznamy_list'] =
LekarskyZaznam.objects.filter(pacient_id=pacient_id)
return context

```

```

class PacientFilter(FilterSet):
    jmeno = CharFilter(field_name='jmeno', lookup_expr='icontains', label="Jméno")
    prijmeni = CharFilter(field_name='prijmeni', lookup_expr='icontains',
label="Příjmení")
    datum_narozeni_od = DateFilter(field_name='datum_narozeni', lookup_expr='gte',
label="Datum narození od")
    datum_narozeni_do = DateFilter(field_name='datum_narozeni', lookup_expr='lte',
label="Datum narození do")

```

```

class Meta:
    model = Pacient
    fields = []

```

```

from django_filters.views import FilterView
from .filters import PacientFilter

```

```

class PacientListView(FilterView, ListView):
    model = Pacient
    filterset_class = PacientFilter
    template_name = 'crm/pacient_list.html'

```

```

context_object_name = 'pacienti'

def get_queryset(self):
    self.filter = PatientFilter(self.request.GET, queryset=super().get_queryset())
    return self.filter.qs

def get_context_data(self, **kwargs):
    context = super().get_context_data(**kwargs)
    context['filter'] = self.filter
    return context

class PatientCreateView(CreateView):
    model = Patient
    form_class = PatientForm
    template_name = 'crm/form.html'

    def form_valid(self, form):
        response = super().form_valid(form)
        # Odeslání potvrzovacího e-mailu
        send_mail(
            'Registrace nového pacienta',
            f'Byl jste úspěšně zaregistrován v systému. Vaše ID pacienta je {self.object.id}.',
            'from@example.com',
            [self.object.email],
            fail_silently=False,
        )
        return response

    def get_success_url(self):
        # Dynamické nastavení URL pro přesměrování na detail pacienta
        return reverse('patient_detail', kwargs={'pk': self.object.pk})

```

Příloha 6

rodičovska základní šablona base_html:

```
{% load static %}
<!DOCTYPE html>
<html lang="cs">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}Název stránky{% endblock %}</title>

<style>
  html, body {
    font-family: 'Arial';
    margin: 0;
    padding: 0;
    height: 100%;
    width: 100%;
    color: #444;
  }

  a {
    color: #0059b2;
    text-decoration: none;
  }
  a:hover {
    color: #CC0000;
    text-decoration: underline;
  }

  .header {
    background-color: #0F7CE6;
    height: 60px;
```

```
display: flex;
align-items: center;
justify-content: space-between;
padding: 0 20px;
}
```

```
.logo {
    background: url('/static/images/logo.png') no-repeat center;
background-size: contain;
width: 70px;
height: 60px;
margin-right: 20px;
}
```

```
#menu {
    list-style: none;
display: flex;
margin: 0;
padding: 0;
}
```

```
#menu li a {
    display: block;
padding: 10px 15px;
color: white;
text-decoration: none;
}
```

```
#menu li a:hover {
    background-color: #555;
}
```

```
.content {
  padding: 20px;
}

footer {
  background-color: #0F7CE6;
  color: white;
  padding: 20px 0;
  text-align: center;
}

footer .container {
  max-width: 1200px;
  margin: auto;
  padding: 0 20px;
}

footer a {
  color: #ffffff;
  text-decoration: none;
}

footer a:hover {
  text-decoration: underline;
}

.social-links, .footer-links {
  list-style: none;
  padding: 0;
  text-align: center;
}
```

```
.social-links li, .footer-links li {
    display: inline;
    margin-right: 15px;
}
</style>
```

```
{% block styles %}{% endblock %}
```

```
</head>
```

```
<body>
```

```
<header class="header">
```

```
<div class="logo"></div>
```

```
<nav>
```

```
<ul id="menu">
```

```
<li><a href="{% url 'home' %}">Domů</a></li>
```

```
<li><a href="{% url 'pacienti_list' %}">Pacienti</a></li>
```

```
<li><a href="{% url 'schuzky_list' %}">Schůzky</a></li>
```

```
<li><a href="{% url 'sluzby_list' %}">Služby</a></li>
```

```
<li><a href="{% url 'zaznamy_list' %}">Lékařské Záznamy</a></li>
```

```
</ul>
```

```
</nav>
```

```
</header>
```

```
<main>
```

```
{% block content %}
```

```
<!-- Hlavní obsah stránky -->
```

```
{% endblock %}
```

```
</main>
```

```
<footer>
```

```
<div class="container">
```

```
<p>© 2024 Název kliniky nebo CRM systému. Všechna práva vyhrazena.</p>
```

```
<p>Kontaktujte nás: <a href="mailto:info@klinika.cz">info@klinika.cz</a> |
```

```
Telefon: +420 123 456 789</p>
```

```

<ul class="social-links">
  <li><a href="https://www.facebook.com/klinika">Facebook</a></li>
  <li><a href="https://www.twitter.com/klinika">Twitter</a></li>
  <li><a href="https://www.instagram.com/klinika">Instagram</a></li>
  <!-- Další sociální sítě podle potřeby -->
</ul>
<ul class="footer-links">
  <li><a href="{% url 'privacy_policy' %}">Politika ochrany osobních
údajů</a></li>
  <li><a href="{% url 'terms_of_service' %}">Podmínky používání</a></li>
  <!-- Další odkazy -->
</ul>
</div>
</footer>
{% block scripts %}
<!-- skripty -->
<script src="{% static 'js/script.js' %}"></script>
{% endblock %}
</body>
</html>

```

Příloha 7

příklad rozšířené šablony

```
{% extends 'base_html.html' %}
```

```
{% block content %}
```

```
<style>
```

```

.pacienti-list {
  margin: 20px auto;
  width: 95%;
}

```

```
}

.pacienti-list table {
  width: 100%;
  border-collapse: collapse;
}

.pacienti-list th, .pacienti-list td {
  border: 1px solid #ddd;
  padding: 8px;
  text-align: left;
}

.pacienti-list th {
  background-color: #4CAF50;
  color: white;
}

.pacienti-list tr:nth-child(odd) {
  background-color: #f2f2f2;
}

.pacienti-list tr:hover {
  background-color: #ddd;
}

.btn {
  background-color: #4CAF50;
  color: white;
  padding: 6px 12px;
  margin: 5px 0;
  border: none;
```



```
cursor: pointer;
border-radius: 4px;
text-decoration: none;
}
```

```
.btn:hover {
    background-color: #45a049;
}
```

```
</style>
```

```
<h1>Seznam pacientů</h1>
```

```
<div class="filter-form">
```

```
  <form method="get">
```

```
    {{ filter.form.as_p }}
```

```
    <button type="submit" class="btn">Filtrovat</button>
```

```
  </form>
```

```
</div>
```

```
<div class="pacienti-list">
```

```
  {% if pacienti %}
```

```
    <table>
```

```
  <thead>
```

```
    <tr>
```

```
      <th>Jméno</th>
```

```
      <th>Příjmení</th>
```

```
      <th>Datum narození</th>
```

```
      <th>Email</th>
```

```
      <th>Telefon</th>
```

```
      <th>Adresa</th>
```

```
    </tr>
```

```

</thead>
<tbody>
  {% for pacient in pacienti %}
    <tr>
      <td>{{ pacient.jmeno }}</td>
      <td>{{ pacient.prijmeni }}</td>
      <td>{{ pacient.datum_narozeni|date:"d.m.Y" }}</td>
      <td>{{ pacient.email }}</td>
      <td>{{ pacient.kontaktني_cislo }}</td>
      <td>{{ pacient.adresa }}</td>
    </tr>
  {% endfor %}

</tbody> </table>
{% else %}
  <p>Žádní pacienti nebyli nalezeni.</p>
{% endif %}
</div>
{% endblock %}

```