

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЇВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

Розробка CRM-системи для стоматологічної клініки

Текстова частина до курсової роботи
за спеціальністю «Комп'ютерні науки» - 122

Керівник курсової роботи

асистент

Калітовський Б.В.

(Підпис)

“ ____ ” _____ 2022 року

Виконав студент КН-3

Котков О.А.

“ ____ ” _____ 2022 року

Київ – 2022

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА
АКАДЕМІЯ»

Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Зав. кафедри мультимедійних систем

_____ Жежерун О.П.

“__” _____ 2021 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу студенту Коткову Олександру Андрійовичу

факультету інформатики 3 курсу бакалаврської програми

ТЕМА: Розробка CRM-системи для стоматологічної клініки

Зміст ТЧ до курсової роботи:

Індивідуальне завдання

Вступ

Теоретична частина

Проектування додатку

Висновки

Дата видачі „__” _____ 2022 р. Керівник _____

(підпис)

Завдання отримав _____ (підпис)

№ п/п	Назва етапу курсової роботи	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	25.10.2021	
2.	Аналіз актуальності, дослідження аналогів.	10.11.2021.	
3.	Вивчення технологій для розробки системи.	21.11.2021.	
4.	Створення технічного завдання.	08.01.2022.	
5.	Написання практичної частини.	30.01.2022.	
6.	Написання теоретичної частини.	25.05.2022.	
7.	Внесення змін до роботи відповідно до зауважень наукового керівника.	29.05.2022.	
9.	Захист курсової роботи.	14.06.2022	

Календарний план виконання курсової роботи

Тема: Розробка CRM-системи для стоматологічної клініки

Календарний план виконання курсової роботи:

Студент Котков О.А.

Керівник Калітовський Б.В.

“ ____ ” _____ 2022 р.

Зміст

Вступ	5
РОЗДІЛ 1: Опис предметної області	7
1.1. Типи CRM-систем	7
1.1.1. Операційна CRM-система	7
1.1.2. Аналітична CRM-система	7
1.1.3. Колабораційна CRM-система.....	8
1.2. Аналіз існуючих програмних рішень	8
1.2.1. Pipedrive.....	8
1.2.2. Agile CRM	10
1.2.3. Less Annoying CRM.....	12
1.2.4. Порівняльний аналіз CRM-систем.....	14
1.3. Технічні вимоги.....	16
1.4. Висновок до розділу 1.....	17
РОЗДІЛ 2: Проектування та розробка	18
2.1. Технологічний стек	18
2.1.1. Spring	18
2.1.2. Hibernate	18
2.1.3. Project Lombok	19
2.1.4. Hibernate Validator	19
2.1.5. Thymeleaf.....	19
2.2. ER-модель бази даних.....	19
2.3. Карта сайту CRM-системи.	20
2.4. Висновок до розділу 2.....	21
РОЗДІЛ 3: Програмна реалізація CRM-системи	22
3.1. Загальна структура проекту	22
3.2. Процес авторизації.....	22
3.3. Взаємодія клієнта та сервера.....	24
3.4. Опис інтерфейсних рішень.....	26
3.5. Висновок до розділу 3.....	39
Висновки	40
Список використаної літератури	41

Вступ

Актуальність та практичне значення обраної теми

На сьогоднішній день бізнесу стає дедалі складніше знаходити нових клієнтів та втримувати існуючих. Конкуренція між компаніями призводить до поліпшення якості кінцевого продукту, але разом з тим збільшуються очікування користувачів. В зв'язку з цим, найбільше виграють компанії, які вкладаються в поліпшення якості сервісу і використовують персональний підхід до кожного клієнта. Менше з тим, зі значним зростанням клієнтської бази постає питання – яким чином забезпечити індивідуальний підхід для кожного? Цим займається CRM (від англ. Customer Relationship Management – управління взаємовідносин з клієнтами).

Згідно визначення, CRM – це «сукупність стратегій, що використовуються компанією для роботи з наявними клієнтами або для залучення нових клієнтів».[1] Ці стратегії перш за все спрямовані на організацію бази даних з інформацією про попередні взаємодії з клієнтами та її аналіз, що значно поліпшує якість обслуговування. У 1980-ті роки вже робились перші спроби налагодити комунікацію між компанією та клієнтом: Роберт і Кейт Кестнбаум аналізували клієнтські бази даних з метою прогнозу, яка категорія клієнтів позитивно відреагує на маркетингову кампанію. Вони стали піонерами напряму, який здобув назву «маркетинг баз даних», а також створили фундамент для розвитку CRM.[2] Проте поступово сам термін CRM стали частіше інтерпретувати не як стратегію ведення бізнесу, а як програмне забезпечення. Це пов'язано з появою CRM-систем, які з часом лише стають більш актуальними для бізнесу.

CRM-система – це керована система, яка передбачає збір і безпечно зберігання інформації, що стосується клієнтів. Вона також є централізованим рішенням

проблеми комунікації між різними відділами компанії та спільного доступу до даних. [3] При успішній імплементації подібної системи в структуру компанії вона зекономить час та поліпшить обізнаність працівників під час роботи з конкретним клієнтом, а тому дозволить виконувати більший обсяг роботи. Сучасні CRM-системи не намагаються зробити свій аналог для всього функціоналу, а замість цього підтримують інтеграції зі сторонніми додатками з можливістю імпорту та експорту даних.

Мета та завдання курсової роботи

Мета: Порівняти різні типи CRM-систем, проаналізувати їх можливості та застосування

Завдання: Дослідження доступних варіантів CRM-систем, створення власної реалізації CRM-системи.

Об'єкт дослідження

CRM-системи різних виробників у відкритому доступі.

Методи дослідження

Дослідження доступних варіантів CRM-систем буде здійснюватися за допомогою вивчення можливостей кожного рішення та проведення порівняльного аналізу.

Джерела дослідження

Було використано у навчальних цілях й опрацьовано багато інтернет-ресурсів, пов'язаних з темою роботи, а також випробовані доступні варіанти CRM-систем та вивчені тренувальні матеріали від їх виробників.

РОЗДІЛ 1: Опис предметної області

1.1. Типи CRM-систем

Одна з головних задач сучасних CRM-систем - надання детальної інформації про клієнтів. Такі системи розрізняють на три типи: операційна, аналітична та колаборативна. Таке розділення перш за все зумовлено типом збереженої інформації про клієнта, а також погляду на цю інформацію. В наступних параграфах описуються основні риси кожного типу.

1.1.1. Операційна CRM-система

Операційна CRM-система підтримує операції, які стосуються взаємодії з клієнтами, як-то кол-центри, веб-сайти, поштову розсилку та інше, при цьому зберігаючи інформацію про клієнтів, потенційних клієнтів та співробітників за допомогою спільного інтерфейсу. Операційна CRM також забезпечує автоматизацію послуг та надає можливість більш ефективно виконувати завдання.

1.1.2. Аналітична CRM-система

Аналітична CRM-система підтримує операції, пов'язані зі збором, безпечним зберіганням, аналізом та видачею статистичних даних про користувачів. Цей тип CRM може аналізувати дані з багатьох джерел, таких як веб-форми, розсилка пошти, опитування тощо. Використання аналітичної CRM допомагає проаналізувати попередню історію взаємодії, визначити уподобання клієнтської бази, спрогнозувати тенденції та планувати стратегії на майбутнє.

1.1.3. Колабораційна CRM-система

Колабораційна CRM-система забезпечує взаємодію між різними відділами компанії та керує загальним розподіленням інформації, дозволяючи працівнику мати актуальну інформацію про роботу з клієнтом. Це корисно при роботі з залученням декількох відділів, а завдяки оптимізації обміну даними навіть віддаленим працівникам легко працювати в команді.

1.2. Аналіз існуючих програмних рішень

При дослідженні існуючих програмних рішень було проаналізовано та випробовано три CRM-системи, а саме: Pipedrive, Agile CRM, Less Annoying CRM. Нижче наведені характеристики вказаних систем та проведений порівняльний аналіз між ними.

1.2.1. Pipedrive

Тип системи

Pipedrive поєднує в собі операційну та аналітичну CRM-систему.

Головна сторінка

Головна сторінка додатку - «Deals» - показує угоди, класифіковані за етапами (показано у вигляді різних стовпців).[4] В цьому ж вікні можна додавати нові угоди, контакти та організації. Кожна угода помічається як виграна або програна, в разі програшу є можливість описати причину для подальшого аналізу. Угоди, які мають найвищий пріоритет (відсортовані за датою), показуються вищими в стовпцях. Користувач може мати більше ніж одне таке вікно та перемикатись між ними, а також кастомізувати кожне з вікон за

своїми потребами – додавати або видаляти стовпці, міняти назву стовпців, продилюватись угоди у вигляді списку.

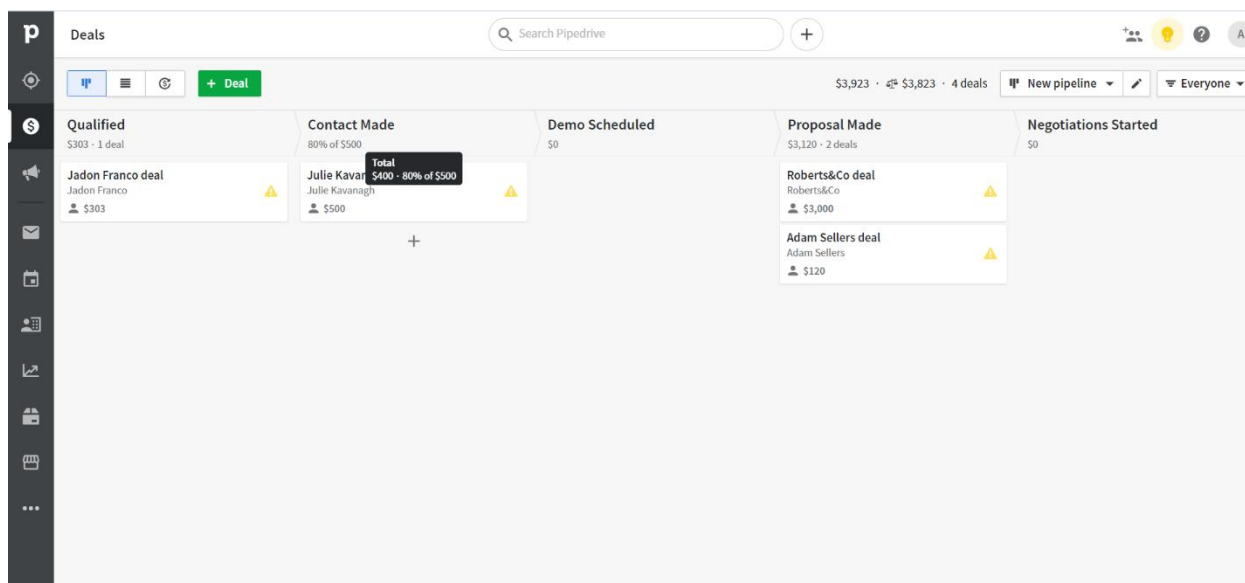


Рисунок 1.2.1.1 – головна сторінка CRM-системи Pipedrive

Основні вкладки

«Leads» – робота з потенційними клієнтами. Створюються так само як і контакти, є можливість імпорту даних, а також у стадії бета-розробки інтеграції з Facebook та WhatsApp для комунікації з ними.

«Mail» – відправка та отримання електронної пошти. Наявна функція смарт-адреси.

«Activities» – додавання подій. Можливо обрати тип та дату події, прив'язати до контакту, потенційного клієнта або організації, а також надіслати запрошення за електронною адресою і використати інтеграції з Google Meet/MS Teams/Zoom Meetings.

«Contacts» – створення контакту/організації. Є розділ «Contacts timeline», де наведена дата створення кожного з контактів/організацій.

Збір статистики

Вкладка «Insights» містить статистичні дані, що фільтруються за датою та користувачами. У базовому плані доступно 11 доповідей стосовно потенційних клієнтів, угод та прогнозу прибутку. У плані Enterprise та вище доступно 150 доповідей.

Додаткові можливості

- Зручний імпорт та експорт даних : доступний імпорт з 30 провідних CRM-систем, а також імпорт та експорт у форматах CSV, Excel.
- Шаблони автоматизації робочого процесу.
- Інтеграція з RingCentral (VoIP-система) та DocuSign(система електронного документообігу, що надає можливість підписувати документи не відходячи від комп'ютера).

1.2.2. Agile CRM

Тип системи

Agile CRM поєднує в собі операційну, аналітичну та колаборативну CRM-систему.

Головна сторінка

Головна сторінка додатку для відділу продажу показує таблиці - список контактів, календар з запланованими подіями та угодами, список завдань та останніх дій.[5] Усі з цих таблиць можуть фільтруватись за датою, а також можливо додати інші таблиці, як-то список подій, діаграма угод, діаграма програних угод та інше. Також є окремі вікна для відділу маркетингу та сервісу, налаштовані під потреби цих відділів. Є можливість створювати свої власні вікна для кожного з відділів та кастомізувати їх.

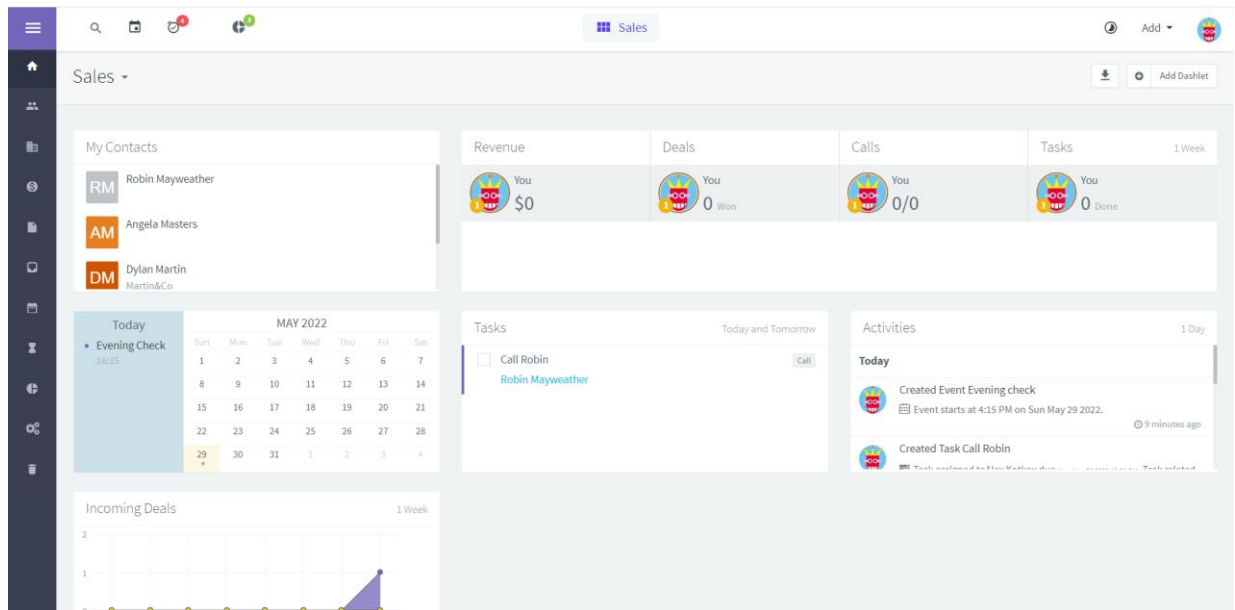


Рисунок 1.2.2.1 – головна сторінка відділу продажів CRM-системи Agile CRM

Основні вкладки

«Inbox» – відправка та отримання електронної пошти. Наявна функція смарт-адреси, а також можливо створювати шаблони електронних листів.

«Activities» – показ останніх дій користувача з можливістю фільтрування за датою.

«Contacts» – створення контакту, а також місце де можна побачити всі контакти з можливістю фільтрування за країною, датою створення, датою редагування, тегами та іншим.

«Companies» – створення організації, а також місце де можна побачити всі організації з можливістю фільтрування за країною, датою створення, датою редагування, тегами та іншим.

«Deals» – створення угод, а також місце де можна побачити всі угоди, класифіковані за етапами (показано у вигляді різних стовпців). Кожна угода помічається як виграна або програна, в разі програшу є можливість обрати причину програшу або написати свою. Угоди можна фільтрувати за датою створення, датою закриття, датою виграшу, ціною і тегами.

«Documents» - створення документів або електронних документів.

Відділ маркетингу має вкладки з функціями, що дозволяють створювати шаблони електронних листів та автоматизувати їх відправку, записувати відео, створювати онлайн-форми для заповнення та інше.

Відділ сервісу має вкладки з функціями, що дозволяють вести чат з клієнтами та збирати відгуки від них, мати доступ до навчальних та тренувальних матеріалів, аналізувати статистичні дані та інше.

Збір статистики

Кожен з відділів має вкладку «Reports», яка містить статистичні дані стосовно усіх ключових аспектів роботи відділу, що фільтруються за датою та користувачами. У базовому плані доступні всі доповіді.

Додаткові можливості

- Велика кількість інтеграцій з відомими брендами у сферах VoIP-телефонії, сервісу, соціального маркетингу та платіжних систем, а також можливість інтегрування кастомного віджету.
- Шаблони автоматизації робочого процесу (у стадії бета-розробки).

1.2.3. Less Annoying CRM

Тип системи

Less Annoying CRM є операційною CRM-системою.

Головна сторінка

Головна сторінка додатку показує список запланованих подій та завдань на сьогодні, останні дії користувача, а також таблицю доповідей, що кастомізується. [6] Також є можливість додати новий контакт, організацію, подію або завдання.

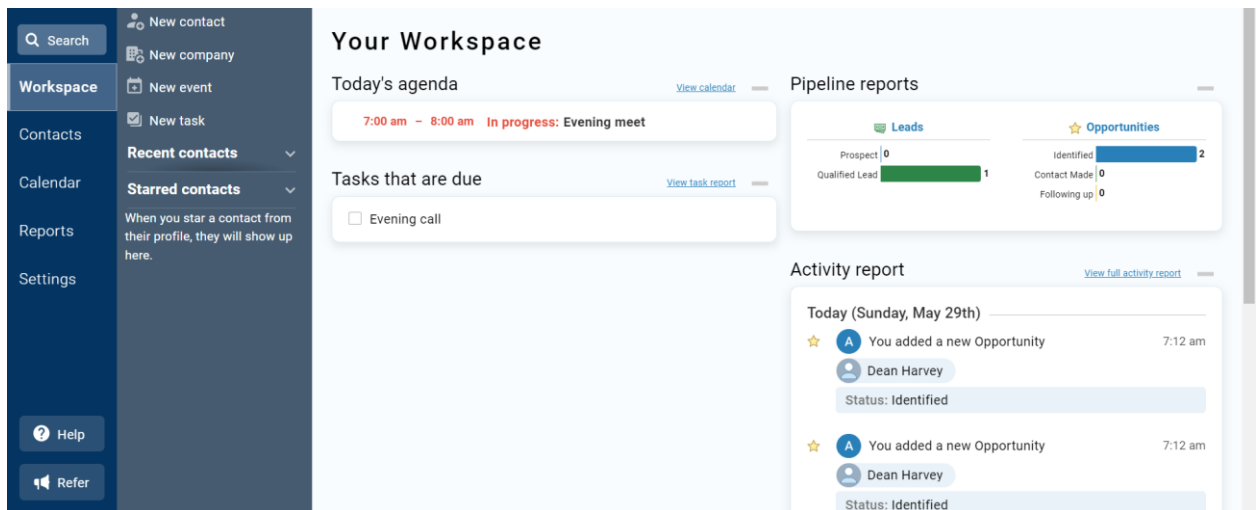


Рисунок 1.2.3.1 – головна сторінка CRM-системи Less Annoying CRM

Основні вкладки

«Contacts» – створення контакту/організації, а також місце де можна побачити всі контакти/організації з можливістю фільтрування за іменем, датою створення, датою оновлення, адресою, віком.

«Calendar» – розширена версія календаря з можливістю додавання події або завдання.

«Mail» – відправка та отримання електронної пошти. Наявна функція смарт-адреси.

Збір статистики

Less Annoying CRM не є аналітичною CRM-системою, а тому не дає можливості проводити глибокий статистичний аналіз. В якості альтернативи система пропонує кастомізовану таблицю доповідей, яку можна побачити на головній сторінці.

Додаткові можливості

- Кастомізація полів при додаванні контактів/організацій.
- Розділення користувачів на команди: так, різні відділи компанії можуть знаходитись в різних командах.
- Зручний експорт даних в форматах .xlsx, .zip, .csv.

- Підтримка інтеграцій з Outlook, Gmail, Yahoo для надсилання та отримання електронної пошти. Також є можливість інтеграції з VoIP-системами Skype та RingCentral.

1.2.4. Порівняльний аналіз CRM-систем

Назва	Pipedrive	Agile CRM	Less Annoying CRM
Максимальна кількість користувачів на аккаунті	Необмежено	Необмежено	Необмежено
Кооперація між відділами	Функціонал для відділа продажу, окремі елементи функціоналу для відділа маркетингу	Функціонал для відділа продажу, маркетингу та сервісу	Функціонал тільки для відділа продажу
Організація зустрічей	Оповіщення всіх учасників, інтеграції з Google Meet, MS Teams, Zoom Meetings	Оповіщення всіх учасників	Оповіщення всіх учасників
Збір відгуків від клієнтів	Так	Так	Ні
Аналітика та доклади	Так	Так	Ні

Адаптивність під мобільні пристрої	Ні	Так	Так
E-mail	Власне рішення, підтримка смарт-адреси + інтеграція з Gmail	Власне рішення, підтримка смарт-адреси та шаблонів електронних листів	Інтеграції з Outlook, Gmail, Yahoo
Календар	Власне рішення + синхронізація з Google Calendar	Власне рішення + синхронізація з Google Calendar або Office 365 Calendar	Власне рішення + синхронізація з Google Calendar
VoIP-система	Інтеграція з RingCentral	Інтеграція з RingCentral	Інтеграція з RingCentral, Skype та можливість налаштувати інтеграцію з довільною VoIP-системою
Додаткові інтеграції	DocuSign, WhatsApp, Facebook та інше (всього 357 інтеграцій)	Twitter, Facebook, LinkedIn, ZenDesk, PayPal та інше	-

Таблиця 1.2.4.1 – порівняння CRM-систем Pipedrive, Agile CRM, Less Annoying CRM

1.3. Технічні вимоги

- Кросбраузерність
- Адаптивність
- Використання фірмових кольорів компанії (#0487E2, #FFFFFF)

CRM-система матиме такі ролі:

- Доктор
- Секретар
- Адміністратор (супер-юзер)

Доктор має вирішувати такі задачі:

- Додавати нові події, контакти.
- Редагувати події, контакти.
- Переглядати список усіх контактів на аккаунті.
- Переглядати список усіх користувачів на аккаунті.
- Переглядати список подій, в яких він бере участь.
- Бачити події в яких він бере участь в календарі на головній сторінці.
- Бачити найближчі записи в календарі на головній сторінці.
- Бачити історію записів контакту.

Секретар має вирішувати такі задачі:

- Додавати нові записи, події, контакти.
- Переглядати список усіх контактів на аккаунті.
- Переглядати список усіх користувачів на аккаунті.
- Переглядати список усіх записів на аккаунті.
- Переглядати список подій, в яких він бере участь.
- Редагувати контакти, події та записи.
- Бачити події в яких він бере участь в календарі на головній сторінці.

Адміністратор має вирішувати такі задачі:

- Додавати нові записи, події, контакти.
- Переглядати список усіх контактів на аккаунті.
- Переглядати список усіх користувачів на аккаунті.
- Переглядати список усіх подій на аккаунті.
- Переглядати список усіх записів на аккаунті.
- Редагувати створені ним контакти, події та записи.
- Бачити події в яких він бере участь в календарі на головній сторінці.
- Видаляти користувачів.
- Реєструвати нових користувачів.

1.4. Висновок до розділу 1

У розділі були визначені основні типи CRM-систем, досліджені існуючі програмні рішення та проведене порівняння між ними, а також наведений опис користувачів CRM-системи та визначені їх технічні вимоги.

РОЗДІЛ 2: Проектування та розробка

Задача даної курсової роботи: спроектувати та розробити CRM-систему для стоматологічної клініки, яка являтиме собою веб-застосунок.

2.1. Технологічний стек

2.1.1. Spring

Spring – це фреймворк базований на мові Java, призначений для створення веб-застосунків.

Даний фреймворк є кросплатформенним, тож його буде можливо розгорнути на всіх основних операційних системах. Spring забезпечує зручну ін'єкцію залежностей, що дозволяє легке під'єднання до бази даних і загально скорочує час розробки.[7] Також цей фреймворк дозволяє легко імплементувати принципи REST за допомогою анотації @RestController.

2.1.2. Hibernate

Під час розробки CRM-системи в якості бази даних використовувалось open-source рішення H2. Основними рисами цієї бази даних є швидкість та безпека. Для зручної роботи з H2 використовується ORM-фреймворк Hibernate. Він дозволяє писати однакові запити незалежно від бази даних, підтримує автоматичне створення таблиць та підтримує концепт «Lazy-Loading», що підвищує швидкість роботи програми. Також Hibernate підтримує об'єктно-орієнтовану мову запитів HQL (Hibernate Query Language), яка є більш потужним аналогом SQL.[8]

2.1.3. Project Lombok

Для скорочення часу розробки використовувалась бібліотека на мові програмування Java Project Lombok. Її ціль - зменшення кількості шаблонного коду за допомогою автоматичного створення байт-коду Java у відповідності з анотаціями проекту.[9]

2.1.4. Hibernate Validator

Для валідації даних поступаючих на вхід від користувача була використана імплементація Java Bean Validation – Hibernate Validator. Модель валідації Java Bean Validator підтримується обмеженнями у вигляді анотацій на полях, методах або класах, визначених користувачем.

2.1.5. Thymeleaf

В якості допоміжного інструмента при роботі з HTML було обрано open-source шаблонізатор Thymeleaf.[10] Його використання допомогло мені зв'язати бекенд і фронтенд мого застосунку, а також надало прості та зручні рішення для виводу інформації з додатку на екран.

2.2. ER-модель бази даних

Нижче наведено модель «сутність-зв'язок» для бази даних розробляемого веб-застосунку.

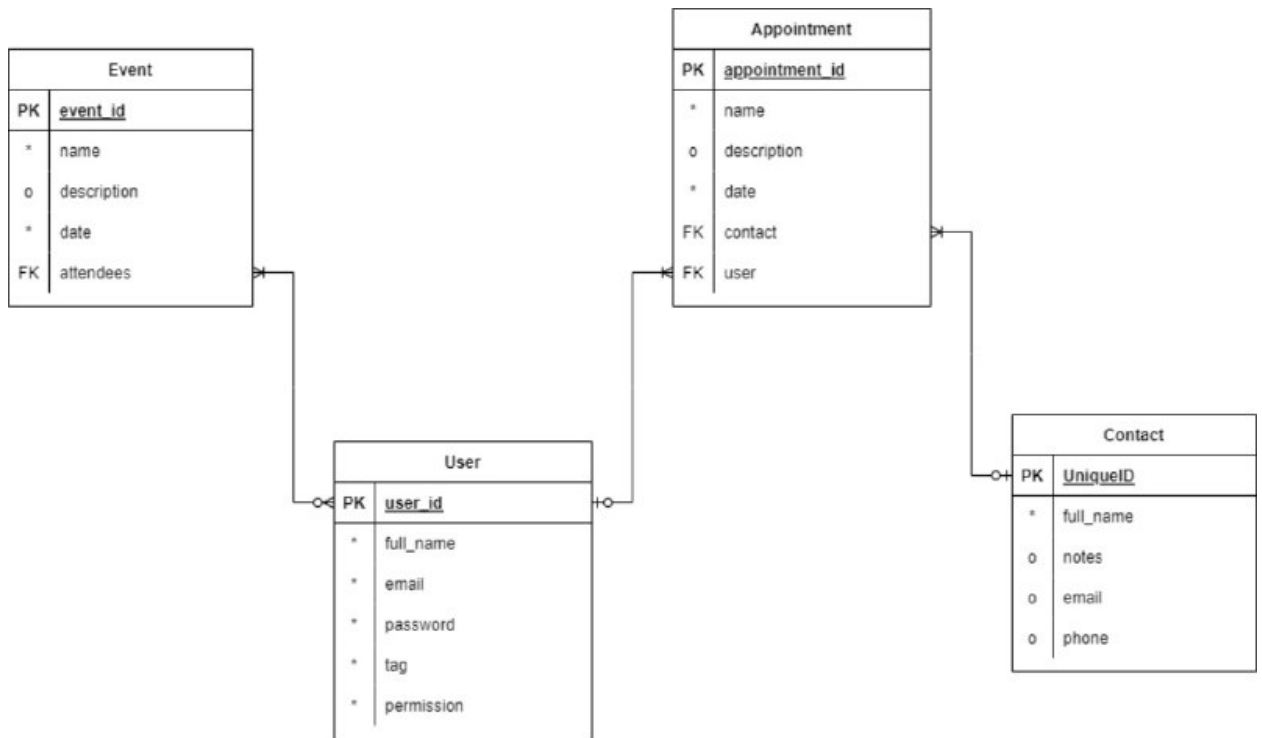


Рисунок 2.2.1 - ER-модель бази даних для CRM-системи

2.3. Карта сайту CRM-системи.

Карта сайту (англ. – sitemap) – це повний перелік усіх розділів/сторінок сайту в ієрархічному порядку. Використовується як елемент навігації, щоб показати зв'язки між сторінками сайту. [11]

Нижче наведено карту сайту розробляемого веб-застосунку.

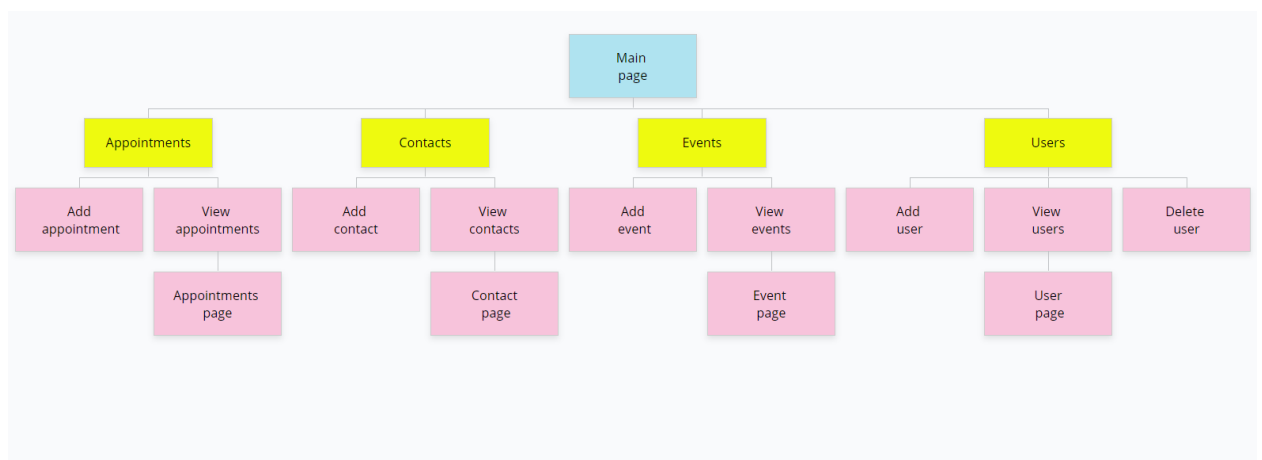


Рисунок 2.3.1 - sitemap розробляємої CRM-системи

2.4. Висновок до розділу 2

У розділі були визначені та розглянуті основні інструменти, що використовувались при розробці, а також надані модель «сутність-зв'язок» бази даних і карта сайту веб-застосунку.

РОЗДІЛ 3: Програмна реалізація CRM-системи

3.1. Загальна структура проекту

При розробці CRM-системи було використано паттерн Controller-Service-Repository. Суть паттерну полягає в тому, щоб розбити роботу серверу на три шари. Controller відповідає за прийом та передачу даних до Service, загальну логіку застосунку. Service містить в собі усю бізнес-логіку та звертається до Repository, коли мова йде про збереження, оновлення або видалення даних. Repository працює лише з базою даних. Такий підхід забезпечує логічне розподілення обов'язків та підвищує читабельність коду. Виходячи з вищесказаного, проект був поділений на модулі: controller, service, repository. Додаткові модулі: entity для зберігання основних сутностей, config для зберігання конфігурації, яка застосовується при авторизації користувача. Файли .html розташовані в директорії resources/templates проекту, а реюзабельні компоненти (наприклад, навігаційна панель) – в піддиректорії resources/templates/fragments. Файли .js розташовані в директорії resources/static, файли .css – в піддиректорії resources/static/css.

3.2. Процес авторизації

Авторизація користувачів CRM-системи налаштована з використанням фреймворку Spring Security, який є стандартним інструментом для забезпечення безпеки застосунків Spring, а також надає необхідний функціонал для легкого розбиття користувачів на ролі. Для кодування паролів був використаний механізм BCryptPasswordEncoder – одне з наявних готових рішень у Spring Security з сильною функцією хешування та можливість обрати кількість ітерацій від 4 до 31 (варто зауважити що час на хешування пароля

росте експоненційно з кількістю ітерацій). За замовчуванням кількість ітерацій дорівнює 10. [12]

Для тестових цілей в базу даних були мануально внесені адміністратор, секретарі та доктори. Їх авторизація відбувалась за допомогою методу `inMemoryAuthentication()` екземпляра класу `AuthenticationManagerBuilder`:

```
@Override
protected void configure(final AuthenticationManagerBuilder auth) throws Exception {

    auth.inMemoryAuthentication().withUser( username: "admin@hismail.com").password(encoder().encode( rawPassword: "admin123")).authorities(Permission.ADMIN.name());
    auth.inMemoryAuthentication().withUser( username: "sec1@hismail.com").password(encoder().encode( rawPassword: "sec123")).authorities(Permission.SECRETARY.name());
    auth.inMemoryAuthentication().withUser( username: "sec2@hismail.com").password(encoder().encode( rawPassword: "sec123")).authorities(Permission.SECRETARY.name());

    auth.inMemoryAuthentication().withUser( username: "frank@hismail.com").password(encoder().encode( rawPassword: "frank123")).authorities(Permission.DOCTOR.name());
    auth.inMemoryAuthentication().withUser( username: "debra@hermail.com").password(encoder().encode( rawPassword: "debra123")).authorities(Permission.DOCTOR.name());
    auth.inMemoryAuthentication().withUser( username: "albert@hismail.com").password(encoder().encode( rawPassword: "albert123")).authorities(Permission.DOCTOR.name());
    auth.inMemoryAuthentication().withUser( username: "raven@hismail.com").password(encoder().encode( rawPassword: "raven123")).authorities(Permission.DOCTOR.name());
    auth.inMemoryAuthentication().withUser( username: "raven2@hismail.com").password(encoder().encode( rawPassword: "raven123")).authorities(Permission.DOCTOR.name());
    auth.inMemoryAuthentication().withUser( username: "raven3@hismail.com").password(encoder().encode( rawPassword: "raven123")).authorities(Permission.DOCTOR.name());

    auth.authenticationProvider(authProvider());
}
```

Рисунок 3.2.1 - Аутентифікація тестових користувачів CRM-системи

Для успішної авторизації Spring Security також потребує реалізації інтерфейсу `UserDetailsService`, а саме методів `loadUserByUsername()` для знаходження користувача в базі даних і `mapAuthorities()` для присвоєння йому ролі:

```
@RequiredArgsConstructor
public class MyUserDetailsService implements UserDetailsService {

    @Autowired
    private final UsersRepository usersRepository;

    @Override
    public UserDetails loadUserByUsername(final String email) throws UsernameNotFoundException {
        final UserEntity user = usersRepository.findByEmail(email)
            .orElseThrow(() -> new UsernameNotFoundException("No user with email: " + email));
        System.out.println(user);
        return User.builder()
            .username(email)
            .password(user.getPassword())
            .authorities(mapAuthorities(Collections.singletonList(user.getPermission())))
            .build();
    }

    private static List<GrantedAuthority> mapAuthorities(final List<PermissionEntity> permissions) {
        return permissions.stream()
            .map(PermissionEntity::getPermission)
            .map(Enum::name)
            .map(SimpleGrantedAuthority::new)
            .collect(Collectors.toUnmodifiableList());
    }
}
```

Рисунок 3.2.2- Реалізація інтерфейсу UserDetailsService

3.3. Взаємодія клієнта та сервера

При розробці веб-застосунку CRM-системи використовувалась клієнт-серверна архітектура. Нижче наведений приклад, як відбувається взаємодія між клієнтом та сервером у застосунку.

- Користувач передає інформацію про контакт, який треба додати, через HTML-форму на сервер.

```
<form id="addContactForm" method="post">

  <fieldset>
    <legend>Add contact:</legend>
    <input type="text" placeholder="Full name..." name="fullName" id="fullName" required>
    <textarea placeholder="Notes..." name="notes" id="notes"></textarea>
    <input type="email" placeholder="E-mail..." name="email" id="email">
    <input type="number" placeholder="Phone number..." name="phone" id="phone">
    <input type="submit" value="Add Contact +">
  </fieldset>
</form>
```

Рисунок 3.3.1- HTML-форма для додавання контакту

- Після натискання користувачем кнопки «Add Contact» інформація передається на сервер, за допомогою AJAX-запиту викликається метод контролера.


```

(function(){
    $('#addContactForm').submit(function(e){
        e.preventDefault();

        $.ajax( url: {
            type: 'POST',
            url: '/contacts/add',
            dataType: 'json',
            data: JSON.stringify( value: {
                fullName: $(this).find( selector: '[name=fullName]').val(),
                notes: $(this).find( selector: '[name=notes]').val(),
                email: $(this).find( selector: '[name=email]').val(),
                phone: $(this).find( selector: '[name=phone]').val(),
            },
            beforeSend: function(xhr) {
                xhr.setRequestHeader('Content-Type', 'application/json')
            },
            success: function (response) {
                window.location.href = "../contacts/"+response.id;
                console.log(response);
            },
            error: function () {
                console.log("Error adding a contact!");
            }
        });
    });
})();

```

Рисунок 3.3.2 - AJAX-запит, передача інформації на вхід контролеру

- Контролер приймає запит та дані з AJAX-запиту, передає їх сервісу.

```

@PostMapping("/add")
@ResponseBody
public ResponseEntity<ContactEntity> addContact(@RequestBody ContactEntity contactEntity) {
    contactsService.createContact(contactEntity);
    return ResponseEntity.status(HttpStatus.OK).body(contactEntity);
}

```

Рисунок 3.3.3 – Робота ContactsController

- Сервіс приймає дані з контролеру. Оскільки дія передбачає збереження контакту в базу даних, сервіс викликає метод репозиторія.

```
@Transactional
public ContactEntity createContact(ContactEntity contactEntity) {
    return contactsRepository.saveAndFlush(contactEntity);
}
```

Рисунок 3.3.4 - Робота ContactsService

- Репозиторій отримує дані, додає користувача до бази даних. Контролер надсилає відповідь, AJAX-запит виводить її у лог консолі та перекидає користувача на сторінку новоствореного контакту.

```
public interface ContactsRepository extends JpaRepository<ContactEntity, Integer> {

    List<ContactEntity> findAll();
    ContactEntity saveAndFlush(ContactEntity contactEntity);

    Optional<ContactEntity> findById(long id);
}
```

Рисунок 3.3.5 - Робота ContactsRepository

3.4. Опис інтерфейсних рішень

Головна сторінка:

- Містить основну навігацію, інформацію про активність користувача за останній тиждень, календар з показом запланованих зустрічей, кнопку “Add Appointment”, “Add Contact”, “Add Event”.
- Основна навігація розташована на панелі навігації зверху сторінки, панель показується на усіх сторінках додатку.

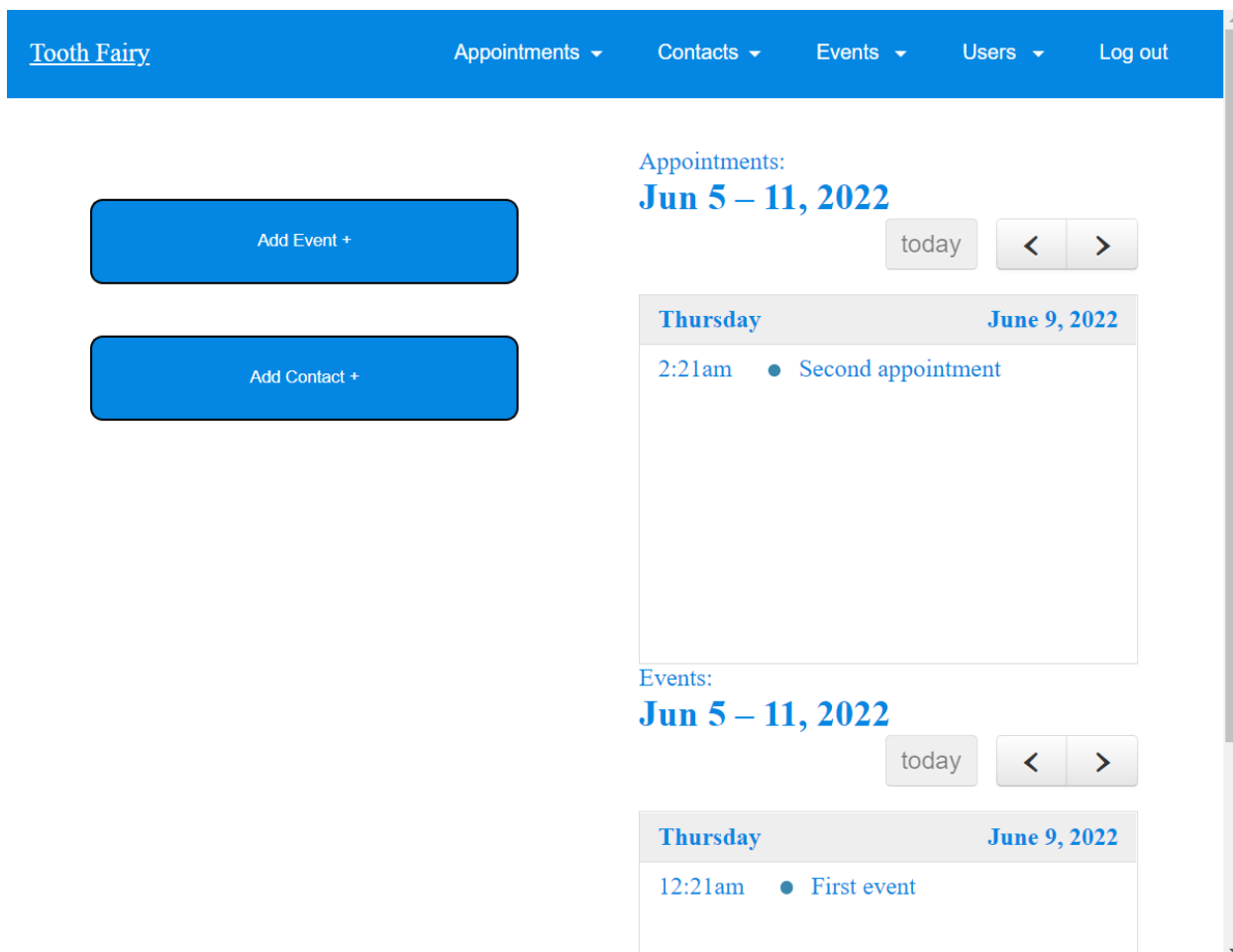


Рисунок 3.4.1 - головна сторінка

Навігаційна панель:

- Містить пункти :
 - “Appointments” з підпунктами “Add Appointment”, “View Appointments”
 - “Contacts” з підпунктами “Add Contact”, “View Contacts”
 - “Events” з підпунктами “Add Event”, “View Events”
 - “Users” з підпунктом “View Users” (“Add Users”, “Delete Users” – для адміністратора)
 - “Log out”

При натисканні на логотип користувач переходить на головну сторінку.

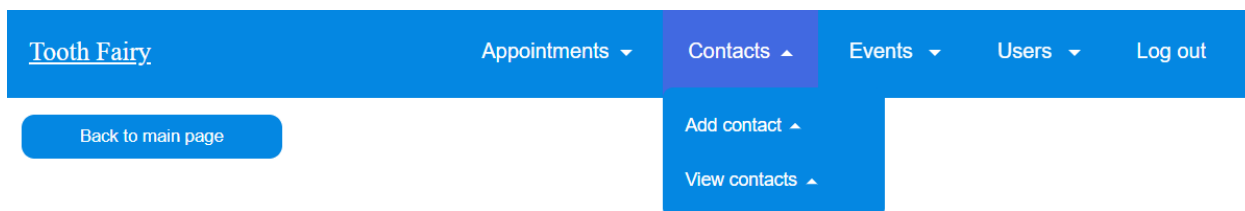


Рисунок 3.4.2 – Навігаційна панель

“Appointments”

“Add Appointment”

- Містить наступні поля для заповнення:

- Name*
- Description
- Date*
- Contact* (обирається з випадаючого списку)
- User* (обирається з випадаючого списку)

Знизу сторінки розташована кнопка «Add Appointment», при натисканні новий запис створюється, а користувач переходить на сторінку запису.

Рисунок 3.4.3 - Сторінка "Add Appointment"

“View Appointments”

- Містить список з угод, в яких користувач бере участь. Містить всю інформацію про запис.
- При натисканні на одну з угод користувач переходить на сторінку запису.

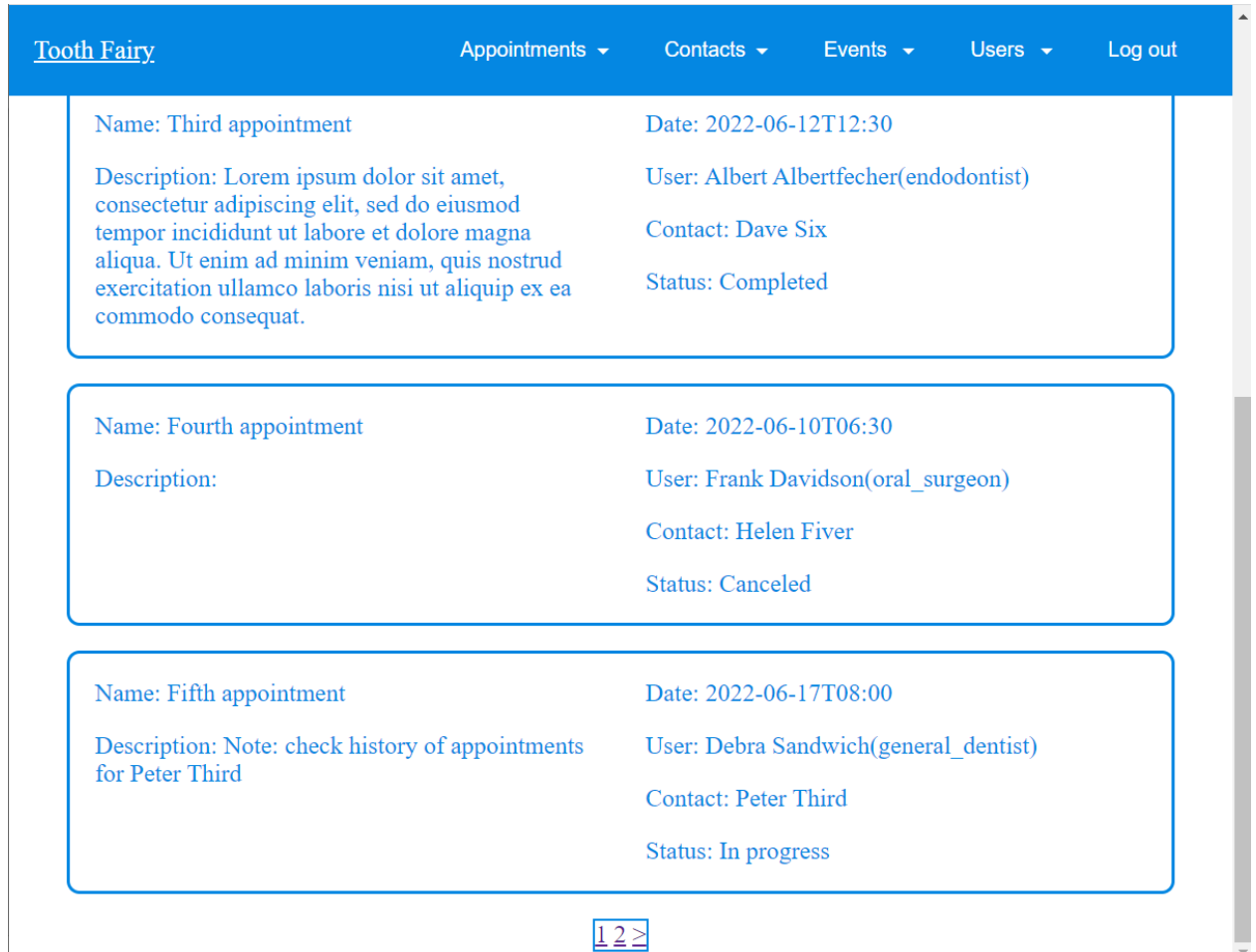


Рисунок 3.4.4 - сторінка "View Appointments"

Сторінка запису

- Містить в собі усю інформацію про запис, всі поля є змінюваними.
- Окрім цього, є можливість обрати з випадаючого списку статус угоди : “Completed”-“Canceled”-“In Progress”.

[Tooth Fairy](#)
[Appointments ▾](#)
[Contacts ▾](#)
[Events ▾](#)
[Users ▾](#)
[Log out](#)

Back to main page

First Appointment

Name

Date

Description

User

Contact

Status

Submit

Рисунок 3.4.5 - Сторінка запису

“Contacts”

“Add Contact”

- Містить наступні поля для заповнення:
 - Full name*
 - E-mail
 - Phone
 - Notes (сюди можна записувати нотатки стосовно контакта).

Знизу сторінки розташована кнопка «Add Contact», при натисканні новий контакт створюється, а користувач переходить на сторінку контакту.

[Tooth Fairy](#)[Appointments ▾](#)[Contacts ▾](#)[Events ▾](#)[Users ▾](#)[Log out](#)

Back to main page

Add contact:

Dave Fellow

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

dave@hismail.com

1222345646

Add Contact +

Рисунок 3.4.6 - Сторінка "Add Contact"

“View Contacts”

- Містить список з усіх контактів що були створені в системі.
- При натисканні на один з контактів користувач переходить на сторінку контакту.

Tooth Fairy Appointments ▾ Contacts ▾ Events ▾ Users ▾ Log out

[back to main page](#)

Full name: Dave Six	Email:
Notes: Helped us to test the pagination feature	Phone number:

Full name: Dave Fellow	Email: dave@hismail.com
Notes: Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.	Phone number: 1222345646

Full name: Valerie Schafer	Email: valerie@hermail.com
Notes:	Phone number:

Full name: Helen DeLaBelle	Email:
Notes: Ask her contact number when she arrives	Phone number:

< 1 2

Рисунок 3.4.7 - Сторінка "View Contacts"

Сторінка контакту

- Містить в собі усю інформацію про контакт, всі поля є змінюваними.
- До інформації про контакт також додається історія його попередніх візитів.

[Tooth Fairy](#)
[Appointments ▾](#)
[Contacts ▾](#)
[Events ▾](#)
[Users ▾](#)
[Log out](#)

Back to main page

Vanessa Seconder

Full name

E-mail

Notes

Phone number

Appointments

Name	Doctor	Date	Status
First appointment	Frank Davidson	2022-06-09T00:21	In progress
Second appointment	Debra Sandwich	2022-06-09T02:21	In progress

Submit

Рисунок 3.4.8 - Сторінка контакту (історія записів видна лише ролі "Доктор")

5. “Events”

“Add Event”

- Містить наступні поля для заповнення:

- Name*
- Description
- Date*
- Attendees* (обирається 0 або більше користувачів з випадального списку, це відобразиться у них на головній сторінці)

Знизу сторінки розташована кнопка «Add Event», при натисканні нова подія створюється, а користувач переходить на сторінку події.

[Tooth Fairy](#)

Appointments ▾Contacts ▾Events ▾Users ▾Log out

Back to main page

Add event:

Evening event

Place: staff room
Bring with you: good mood
Till: 21:30

10.06.2022 19:00

Albert Albertfecher (endodontist)
Raven Dawson (orthodontist)
Raven Dawson twin (orthodontist)
Raven Dawson double twin (orthodontist)
Super_admin (null)

Add Event +

Рисунок 3.4.9 - Сторінка "Add Event"

“View Events”

- Містить список з подій, які були створені користувачем.
- При натисканні на одну з подій користувач переходить на сторінку події.

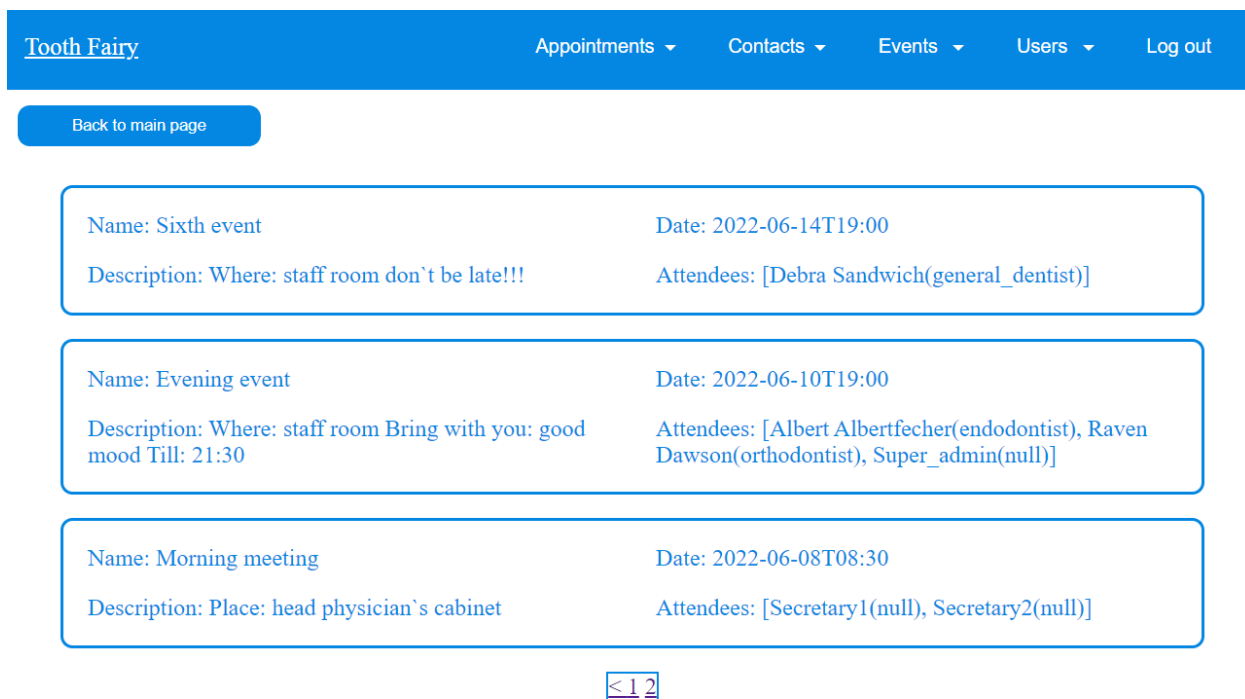


Рисунок 3.4.10 - сторінка "View Events"

Сторінка події

- Містить в собі усю інформацію про подію, поля є змінюваними (сторінка показується лише тому, хто створив подію, інші бачать це на головній сторінці в скороченому вигляді).

[Tooth Fairy](#)
[Appointments](#)
[Contacts](#)
[Events](#)
[Users](#)
[Log out](#)

[Back to main page](#)

Evening event

Name

Date

Description

Attendees

Рисунок 3.4.11 - сторінка події

6. “Users”

“Add User”

- Містить наступні поля для заповнення:

- Full name*
- E-mail*
- Status (роль)*
- Tag (тільки для доктора)

Знизу сторінки розташована кнопка «Add User», при натисканні новий користувач створюється, а адміністратор переходить на сторінку користувача.

[Tooth Fairy](#)
[Appointments](#)
[Contacts](#)
[Events](#)
[Users](#)
[Log out](#)

[Back to main page](#)

Add user:

newuser@hismail.com

Alex Kevland

Doctor

Oral Surgeon

Add User +

Рисунок 3.4.12 - Сторінка "Add User"

“View Users”

- Містить список з користувачів, які є в системі.
- При натисканні на одного з них користувач переходить на сторінку користувача.

[Tooth Fairy](#)
[Appointments](#)
[Contacts](#)
[Events](#)
[Users](#)
[Log out](#)

[Back to main page](#)

Full name: Raven Dawson double twin	Status: DOCTOR
E-mail: raven3@hismail.com	Tag: orthodontist
Full name: Super_admin	Status: ADMIN
E-mail: admin@hismail.com	Tag: null
Full name: Secretary1	Status: SECRETARY
E-mail: sec1@hismail.com	Tag: null

[<1](#)
[2](#)

Рисунок 3.4.13 - Сторінка "View Users"

Сторінка користувача

- Містить в собі усю інформацію про користувача окрім пароля, уся інформація не може бути змінена

Інформація про користувача:

- Full name
- E-mail
- Status
- Tag («ортодонт», «ендодонт», «хірург» та інші)

The screenshot shows the 'Tooth Fairy' user profile page. At the top is a blue navigation bar with the 'Tooth Fairy' logo on the left and links for 'Appointments', 'Contacts', 'Events', 'Users', and 'Log out' on the right. Below the navigation bar is a blue button labeled 'Back to main page'. The main content area has a light blue header with the name 'Alex Kevland'. Below this, there are four input fields arranged in a 2x2 grid. The first row contains 'Full name' with the value 'Alex Kevland' and 'Status' with the value 'DOCTOR'. The second row contains 'E-mail' with the value 'newuser@hismail.com' and 'Tag' with the value 'oral_surgeon'.

Рисунок 3.4.14 - Сторінка користувача

“Delete User”

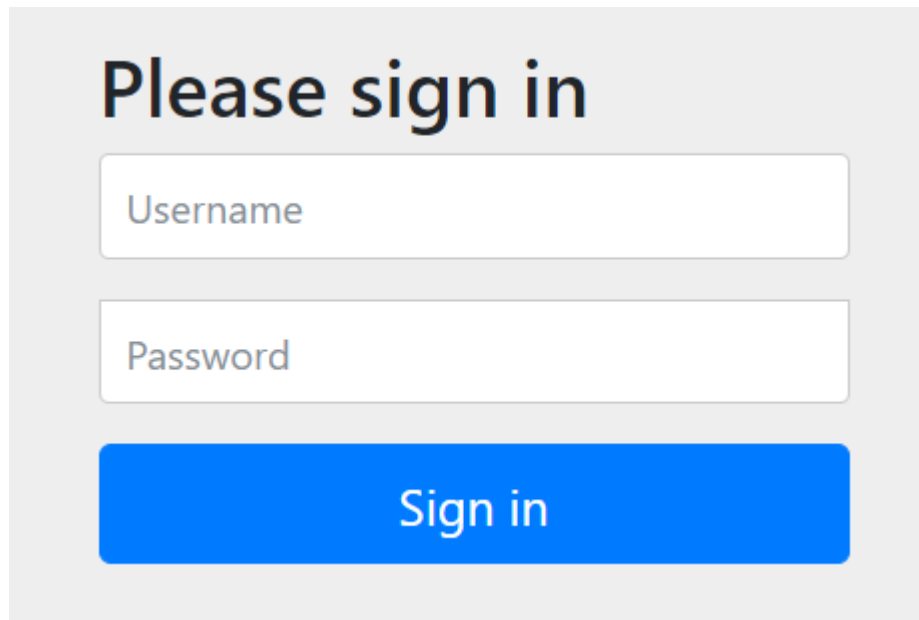
Містить в собі випадаючий список користувачів (окрім адміністратора). При натисканні на кнопку «Delete User» обраний користувач видаляється з системи, а адміністратор переходить на сторінку користувача.

The screenshot shows the 'Delete User' page. It features the same blue navigation bar and 'Back to main page' button as the previous page. The main content area has a light blue header with the text 'Delete user:'. Below this is a dropdown menu showing 'Frank Davidson (oral_surgeon)' with a downward arrow. Below the dropdown is a blue button labeled 'Delete User'.

Рисунок 3.4.15 - Сторінка "Delete User"

Інше

- Авторизація (для зареєстрованих користувачів):
 - Е-mail*
 - Пароль*



Please sign in

Username

Password

Sign in

Рисунок 3.4.16 - Сторінка авторизації

3.5. Висновок до розділу 3

В розділі був наведений опис загальної структури проекту. Продемонстровані елементи коду, що реалізують авторизацію користувача та взаємодію між клієнтом та сервером, а також надано опис інтерфейсних рішень.

Висновки

Під час роботи були визначені усі переваги використання систем управління взаємовідношень з клієнтами, їх основні типи, випробовані та порівняні між собою готові програмні рішення. Поставлене технічне завдання було повністю виконане.

При реалізації системи був використаний фреймворк Spring на мові програмування Java. Усі обрані технології були обрані вдало та в комбінації з паттерном Controller – Service - Repository дали гнучкий та легко розширюваний веб-застосунок. Можливі варіанти розвитку системи: відправка та отримання електронних листів, створення автоматичної системи запису пацієнтів, імпорт та експорт подій з інших календарів.

Список використаної літератури

1. Collins English Dictionary [Електронний ресурс] – Режим доступу до ресурсу: <https://www.collinsdictionary.com/dictionary/english/crm>
2. History Of CRM | Winners FDD [Електронний ресурс] – Режим доступу до ресурсу: <https://winnersfdd.com/2020/12/18/history-of-crm/>
3. Study.com: What are CRM Systems? - Definition, Types & Examples [Електронний ресурс] – Режим доступу до ресурсу: <https://study.com/academy/lesson/what-are-crm-systems-definition-types-examples.html>
4. Sales CRM & Pipeline Management Software | Pipedrive [Електронний ресурс] – Режим доступу до ресурсу: <https://www.pipedrive.com>
5. CRM Software | Customer Relationship Management System | Agile CRM [Електронний ресурс] – Режим доступу до ресурсу: <https://www.agilecrm.com/>
6. Less Annoying CRM | Simple Contact Management for Small Businesses [Електронний ресурс] – Режим доступу до ресурсу: <https://www.lessannoyingcrm.com/>
7. Spring Framework [Електронний ресурс] – Режим доступу до ресурсу: <https://spring.io/projects/spring-framework>
8. Hibernate – Query Language [Електронний ресурс] – Режим доступу до ресурсу: [https://www.tutorialspoint.com/hibernate/hibernate_query_language.htm#:~:text=Hibernate%20Query%20Language%20\(HQL\)%20is,turns%20perform%20action%20on%20database.](https://www.tutorialspoint.com/hibernate/hibernate_query_language.htm#:~:text=Hibernate%20Query%20Language%20(HQL)%20is,turns%20perform%20action%20on%20database.)
9. Lombok – Package lombok [Електронний ресурс] – Режим доступу до ресурсу: <https://projectlombok.org/api/lombok/package-summary.html>
10. Thymeleaf [Електронний ресурс] – Режим доступу до ресурсу: <https://www.thymeleaf.org/>
11. Site map - Wikipedia [Електронний ресурс] – Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Site_map
12. BCryptPasswordEncoder (spring-security-docs 5.7.1 API) [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.spring.io/spring-security/site/docs/current/api/org/springframework/security/crypto/bcrypt/BCryptPasswordEncoder.html>