

ФІС

(повна назва факультету)

Програмної інженерії

(повна назва кафедри)

---

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр

(назва освітнього ступеня)

на тему: Розробка CRM – системи на базі Lumen та Angular для Owl Photobook

---

---

---

Виконав(ла): студент(ка) VI курсу, групи СПм  
спеціальності 121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

(підпис)

(прізвище та ініціали)

Керівник

(підпис)

(прізвище та ініціали)

Нормоконтроль

(підпис)

(прізвище та ініціали)

Завідувач кафедри

(підпис)

(прізвище та ініціали)

Рецензент

(підпис)

(прізвище та ініціали)

## РЕФЕРАТ / ABSTRACT

Атестаційна робота магістра містить с., 17 рис., 2 табл., 16 джер.

CRM-СИСТЕМИ, REST API, СИСТЕМИ ДЛЯ БІЗНЕСУ, УПРАВЛІННЯ  
ВІДНОСИНАМИ З КЛІЄНТАМИ, МЕНЕДЖМЕНТ ЗАМОВЛЕНЬ,  
СТАТИСТИКА.

Метою роботи є аналіз існуючих рішень, збір усіх негативних та позитивних  
нюансів CRM-систем та розробка CRM для Owl Photobook.

У другому розділі розглянуте питання доцільності використання тих чи  
інших технологій, задля досягнення необхідних результатів. Проаналізовано різні  
технології, мови та фрейворки.

У третьому розділі описані результати виконання кваліфікаційної роботи,  
функціональність та можливості системи, перспективи її використання. Здійснено  
тестування, задля всебічної перевірки.

The master's attestation work contains pp., 17 figs., 2 tables, 16 vols.

CRM SYSTEMS, REST API, BUSINESS SYSTEMS, CUSTOMER  
RELATIONS MANAGEMENT, ORDER MANAGEMENT, STATISTICS.

The aim of the work is to analyze existing solutions, collect all the negative and  
positive nuances of CRM-systems and develop CRM for Owl Photobook.

The second section discusses the feasibility of using certain technologies to  
achieve the desired results. Various technologies, languages and frameworks are  
analyzed.

The third section describes the results of the qualification work, functionality and  
capabilities of the system, prospects for its use. Testing has been performed for a  
comprehensive check.

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

CRM (Управління відносинами з клієнтами) — поняття, що охоплює концепції, котрі використовуються компаніями для управління взаємовідносинами зі споживачами, включаючи збір, зберігання й аналіз інформації про споживачів, постачальників, партнерів та інформації про взаємовідносини з ними.

REST (Передача репрезентативного стану) — підхід до архітектури мережевих протоколів, які надають доступ до інформаційних ресурсів.

B2C - тип торгівлі товарами, а також виконання певних послуг, де покупцем є кінцевий споживач, фізична чи юридична особа.

B2B – (Бізнес для бізнесу) - маркетинговий термін в економіці, що означає обмін товарами, послугами або інформацією (чи їхній продаж) між компаніями і не включає в цей процес кінцевого фізичного споживача товару чи послуги.

URI - компактний рядок літер, який однозначно ідентифікує окремий абстрактний чи фізичний ресурс.

SEO - процес коригування HTML-коду, текстового наповнення (контенту), структури сайту, контроль зовнішніх чинників для відповідності вимогам алгоритму пошукових систем, з метою підняття позиції сайту в результатах пошуку в цих системах за певними запитами користувачів.

TDD - технологія розробки програмного забезпечення, яка використовує короткі ітерації розробки, що починаються з попереднього написання тестів.

IT – інформаційні технології.

## ЗМІСТ

ВСТУП.....	6
1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ.....	8
1.1 Теоретичні відомості про CRM.....	8
1.2 Використання CRM: переваги та недоліки.....	10
1.3 Огляд існуючих CRM.....	13
1.3.1 Salesforce.....	13
1.3.2 Close.....	14
1.3.3 Agile CRM.....	15
1.3.4 HubSpot CRM.....	16
1.3.5 Zoho CRM.....	18
2 КОНСТРУЮВАННЯ ПРОГРАМНОЇ СИСТЕМИ.....	20
2.1 Інструменти розробки.....	20
2.1.1 WebStorm.....	20
2.1.2 PhpStorm.....	20
2.1.3 mySQLWorkbench.....	21
2.1.4 PHP.....	21
2.1.5 JavaScript.....	22
2.1.6 TypeScript.....	23
2.1.7 CSS.....	23
2.1.8 HTML.....	24
2.2 Використані технології.....	24
2.2.1 Lumen.....	24
2.2.2 Angular.....	25
2.2.3 REST API.....	25
2.2.4 MVC.....	31
2.2.5 OAuth.....	33
2.2.6 JSON Web Token.....	36
3 РОЗРОБКА СИСТЕМИ.....	38

3.1 Функціонал програмного продукту.....	40
3.2 Тестування системи.....	43
4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	48
4.1 Охорона праці.....	48
4.2 Безпека в надзвичайних ситуаціях.....	51
ВИСНОВКИ.....	57
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	58
ДОДАТКИ.....	59

## ВСТУП

**Актуальність теми роботи.** Для власників бізнесу завжди було актуально «тримати руку на пульсі» свого бізнесу, прозоро бачити усю звітність, не вникаючи у бухгалтерські нюанси. CRM-системи допомагають бути ближче до своїх клієнтів, розуміти їхні потреби та давати швидкий відклик на їхні запити. Такі системи допомагають, не тільки зберегти кошти, а й розумно інвестувати їх у правильному напрямку, тобто інвестувати кошти, у ті рекламні компанії, які приносять найбільшу кількість потенційних клієнтів та направлені на правильну ЦА. Завдяки CRM менеджери завжди в контексті подій, які відбуваються, не пропустивши жодної дрібниці – від нового замовлення до нових листів на Email. Дані системи допомагають переглядати статистичні дані, такі як кількість відвідувачів сайту, кількість замовлень і тп. Тобто в правильних руках даний інструмент може підняти бізнес на зовсім новий рівень без постійних фінансових вливань.

**Мета дослідження** – аналіз існуючих рішень, збір усіх негативних та позитивних нюансів CRM-систем та розробка CRM та REST API для CRM, щоб задовільнити потреби та виконати поставлені, компанією Owl Photobook, завдання.

Перед початком дослідницької роботи були поставлені наступні **завдання**:

- Провести аналіз існуючих рішень – CRM-систем;
- Провести аналіз функціональних можливостей CRM-систем;
- Провести аналіз переваг та недоліків CRM-систем;
- Провести аналіз потреб та проблем компанії, задля вдалого їх вирішення, шляхом розробки програмного продукту;
- Визначити найбільш вдалі технологічні рішення, задля вирішення проблем;
- Розробити систему за технічним завданням.

**Об'єктом дослідження** є CRM-системи.

**Предмет дослідження** – розробка CRM-системи для Owl Photobook.

Дана розробка немає **наукової новизни**, адже схожі CRM-системи уже давно існують, проте особливістю даної розробки є те, що вона розроблена під певну нішу і задовольняє функціональні потреби певної категорії компаній.

Дана система не буде абстрактною чи практично не використаною розробкою, CRM-система та REST API до неї має **практичне застосування** в друкарні Owl Photobook, допомагає залучити потенційних клієнтів та легко менеджети усі процеси.

**Апробація результатів магістерської роботи** окремі результати роботи представленні на одній науковій конференції:

1. VIII науково-технічна конференція «Інформаційні моделі, системи та технології». На тему: «ІНСТРУМЕНТИ CRM-СИСТЕМИ».

# 1 АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ

## 1.1 Теоретичні відомості про CRM

Управління відносинами з клієнтами (CRM) - це технологія управління відносинами і взаємодією компанії із клієнтами та потенційними клієнтами. Ціль CRM-системи - покращення ділових зв'язків для розвитку бізнесу. CRM-система допомагає компаніям залишатися на зв'язку з клієнтами, впорядковувати процеси та підвищувати прибутковість.

Під CRM зазвичай мають на увазі CRM-систему, як інструмент, що допомагає в управлінні контактами, управлінні продажами, продуктивності агентів тощо. Інструменти CRM можна використовувати для управління взаємовідносинами з клієнтами протягом усього життєвого циклу клієнта, охоплюючи маркетинг, продажі, цифрову торгівлю та взаємодію з клієнтами.

Інструменти CRM дозволяють зберігати контактну інформацію про клієнтів та потенційних клієнтів, фіксувати проблеми із обслуговуванням та керувати маркетинговими кампаніями в одному центральному місці - і робити інформацію про кожну взаємодію з клієнтами доступною для тих, кому вона може знадобитися.

CRM може допомогти компаніям будь-якого розміру стимулювати зростання бізнесу, і це може бути особливо корисно для малого бізнесу, де командам часто потрібно шукати способи зробити більше, а не менше.

Функції, які виконує CRM-система і для чого її використовують: Створює фокус на клієнта; Допомагає миттєво знайти кожного клієнта, його контакт та розпочати спілкування; Покращує після продажне обслуговування та підтримку; Підвищує лояльність та утримання клієнтів; Збільшує продажі та прибутковість; Класифікує та визначає пріоритети потенційних клієнтів; Відстежує потенційних клієнтів; Автоматизовує процес продажу; Допомагає керувати завданнями; Створює централізований центр з усім в одному місці; Створює загальну видимість, яка покладає край дублюванню; Посилує співпрацю між командами та



між командами; Сприяє віддаленій роботі; Впорядковує процеси, інтегруючи інші системи, якими користується бізнес; Забезпечує аналіз та звітність; Точні, докладні звіти та прогнози; Аналіз діяльності клієнтів; Управління маркетинговою рентабельністю інвестицій; Завдяки прозорості системи та легкому доступу до даних простіше співпрацювати та підвищувати продуктивність. Кожен у організації може бачити всі взаємодії з клієнтом у системі, включаючи їхні зв'язки, що вони придбали та коли, скільки заплатили, а також багато, багато іншого. Якщо, звичайно, бізнес не хоче обмежити, хто що бачить, з рівнями дозволу користувача!

CRM - це найбільш швидкозростаюча категорія прикладного програмного забезпечення для підприємств, і до 2027 року, як очікується, світові витрати на CRM становитимуть 114,4 млрд. доларів США. Отримання актуальної, достовірної інформації про прогрес бізнесу може бути складним.

Система CRM може дати чіткий огляд клієнтів. Власник бізнесу може переглядати все в одному місці - просту, налаштовану інформаційну панель, яка може розповісти попередню історію клієнта з бізнесом, стан їхніх замовлень, будь-які непогашені проблеми обслуговування клієнтів тощо. Також можна навіть включити інформацію про їх публічну діяльність у соціальних мережах - їх симпатії та антипатії.

Хоча CRM-системи традиційно використовуються, як інструменти продажу та маркетингу, обслуговування та підтримка споживачів є зростаючим сегментом CRM та критичним елементом управління цілісними відносинами з клієнтами.

Рішення щодо управління взаємовідносинами з клієнтами (CRM) допомагає знайти нових клієнтів, виграти бізнесу, організувавши інформацію про клієнтів та потенційних клієнтів таким чином, щоб допомогти бізнесу побудувати міцніші стосунки з ними та швидше розвивати бізнес. CRM-системи починаються з того, що збирають веб-сайт, електронну пошту, телефон та дані соціальних мереж клієнта - і багато іншого - за різними джерелами та каналами. CRM також може автоматично втягувати іншу інформацію, таку як останні новини про діяльність компанії, може зберігати особисті дані, такі як особисті уподобання клієнта щодо

спілкування. Інструмент CRM упорядковує цю інформацію, щоб надати повну інформацію про приватних осіб та компанії в цілому, щоб мати можливість краще зрозуміти стосунки бізнесу з клієнтом.

З консолідованим уявленням про кожного потенційного клієнта система CRM використовується для управління повсякденною діяльністю та взаємодією із клієнтами. Комерційні команди можуть швидко запускати та масштабувати електронну комерцію - від онлайн-замовлень до призупинення - для своїх споживачів (B2C commerce) та бізнес-покупців (B2B commerce). А агенти з обслуговування клієнтів можуть реагувати на потреби клієнтів на будь-якому каналі – в домі або в офісі.

Платформа CRM також може підключатися до інших бізнес-додатків, які допомагають розвивати стосунки з клієнтами. Сьогодні рішення CRM є більш відкритими і можуть інтегруватися з іншими улюбленими бізнес-інструментами, такими як підписання документів, бухгалтерський облік, виставлення рахунків, та опитування, завдяки чому інформація обмінюється в обидві сторони, щоб дати справжній 360-градусний огляд клієнта.

## **1.2 Використання CRM: переваги та недоліки**

SWOT-аналіз - це коротка форма, яка використовується для опису конкретних сильних, слабких сторін, можливостей та загроз, які є стратегічними факторами для конкретної компанії. SWOT-аналіз може не тільки привести до виявлення відмітних компетенцій корпорації, а й до виявлення можливостей, якими фірма наразі не може скористатися через відсутність відповідних ресурсів. Використання SWOT-аналізу допомагає отримати максимум від майбутніх можливостей, використовуючи його сильні сторони та пом'якшуючи загрози, працюючи над його слабкістю. Це інструмент для перевірки та аналізу організації чи компанії. Це допомагає зосередитися на ключових питаннях. Можливості та

загрози є зовнішніми факторами. Сильні та слабкі сторони - це внутрішні фактори.

#### Сильні сторони:

1. Зосередження та пошук більш вигідних клієнтів: це головна сила хорошої стратегії управління персоналом, оскільки ці клієнти отримують прибуток у майбутньому.

2. Зберігання та підтримка всієї інформації про клієнтів шляхом збору їх даних: CRM збирає та зберігає всю цінну інформацію про клієнтів, щоб організації могли знати про своїх клієнтів для прогнозування їхньої майбутньої поведінки та отримання інших цінних знань для отримання більшого прибутку.

3. Підвищує ефективність продажів: Організація, яка має хороший CRM, знає, що потрібно клієнтам. Це підвищує лояльність клієнта.

4. Забезпечує задоволеність споживачів: Клієнти стають більш залежними від організації, тому вони не залишатимуть організацію.

5. Наявність знань про потенційних клієнтів та покупців: збираючи та зберігаючи дані про клієнтів, а потім використовуючи бізнес-аналітику для отримання інформації та отримання знань, за допомогою систем управління знаннями, CRM може вдосконалювати свої знання про клієнтів. [1]

#### Слабкі сторони:

1. Перевантаження непотрібною інформацією для організації: це може перервати CRM через ускладнення та конфлікти в прогнозуванні та прийнятті рішень, а також можуть бути проблеми з місцем для зберігання.

2. Не вдається досягти успіху з новими введеними даними: CRM не може бути успішним із меншими та новими даними клієнта. Щоб досягти успіху, слід набратися терпіння щодо збору більшої кількості даних, інакше це може призвести до провалу проекту, а організація втратить цінний час та гроші.

3. Відсутність культурної підготовки: впровадження CRM призведе до культурних змін в організації, особливо серед передового персоналу.

4. Недостатньо визначена діяльність: вона стосується внутрішньої діяльності в організаціях, як час процесу та витрати, продуктивність працівників, якість даних та координація завдань. [2]

### Можливості:

1. Задоволення клієнта: CRM може прогнозувати потреби клієнтів, аналізуючи їх дані та інформацію, а також допомагати організаціям затверджувати їх потреби та прогнозувати їхні майбутні потреби, щоб зробити своїх клієнтів щасливішими.

2. Збільшення бази продажів: Існуючі клієнти приносять майбутній прибуток, тому організації повинні пропонувати їм більше. Бази продажів будуть збільшені за рахунок включення інших послуг або заохочень.

3. Поліпшення відносин із клієнтами: Визначення клієнтів, що їм потрібно, зберігання корисних даних про них та контроль взаємовідносин персоналу та клієнтів.

4. Поліпшення взаємодії з клієнтами: Використовуючи нові комунікаційні технології в CRM, організації можуть покращити взаємодію між ними та клієнтами.

### Загрози:

1. Зменшення рівня особистої взаємодії: оскільки CRM використовує технологію для своїх комунікацій, отже, це може зменшити взаємодію очі в очі.

2. Над автоматизацією: CRM може стати занадто комп'ютеризованим і знову призведе до втрати особистої взаємодії. Клієнти віддають перевагу відносинам із своїми постачальниками.

3. Слабка комбінація із системами бек-офісу: якщо впроваджена технологія CRM не взаємодіє належним чином із системою бек-офісу, CRM не може досягти своїх цілей. Технологія окремо не може підтримувати швидкий зворотний зв'язок, постійну взаємодію, адаптацію та підтримку діяльності; компанії повинні розуміти, як заповнити ці прогалини.

4. Підвищення очікувань споживачів: Надаючи деякі незаплановані послуги, такі як дарування подарунків, акцій, бонусів, організації збільшують очікування своїх споживачів, тому в майбутньому вони можуть не мати змоги задовольнити своїх клієнтів, і це може спричинити втрату своїх лояльних клієнтів.

5. Змінення реакції споживача на CRM-діяльність: Іноді організація повністю покладається на CRM, але це можливо в певному сегменті через багато причин,

таких як інфляція, яка викликає різну реакцію на CRM-діяльність.

6. Зменшення доходу: Іноді подібна зворотна реакція клієнтів CRM, як згадувалося вище, призводить до втрати грошей та зменшення доходів організації.

### **1.3 Огляд існуючих CRM**

#### **1.3.1 Salesforce**

Salesforce.com, Inc. - американська компанія, що базується на хмарному програмному забезпеченні, головний офіс якої знаходиться в Сан-Франциско. Він надає послугу управління взаємовідносинами з клієнтами (CRM), а також продає додатковий набір корпоративних програм, орієнтованих на обслуговування клієнтів, автоматизацію маркетингу, аналітику та розробку додатків. У 2020 році журнал Fortune поставив Salesforce на шосте місце у своєму списку 100 найкращих компаній, на які потрібно працювати, на основі опитування співробітників. Логотип компанії зображено на рисунку 1.1.



Рисунок 1.1 Логотип компанії Salesforce

Переваги: Salesforce пропонує настроюваний CRM-додаток із глибокими, добре розробленими інструментами для процесів продажів, співпраці команд та багатьох інтеграцій. Це надійна система, що дозволяє менеджерам з продажу та командам продажів створювати дуже складні системи управління завданнями. Salesforce автоматизує багато нудних завдань та процесів прийняття рішень поза робочим днем. Головну сторінку сайту зображено на рисунку 1.2.

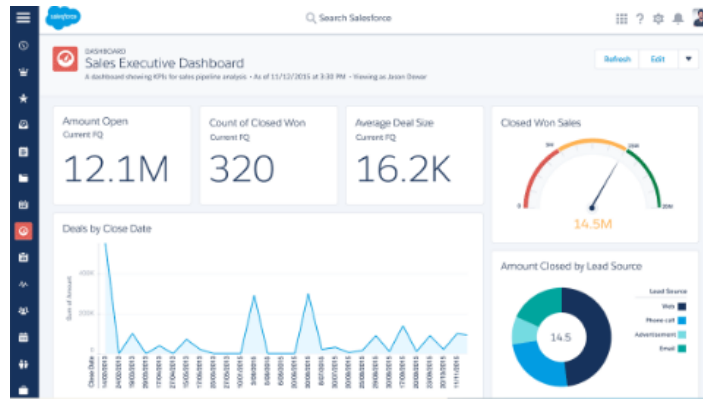


Рисунок 1.2 Скріншот головної сторінки сайту – Salesforce

Недоліки: Немає офлайн - функціональності. Довідкова служба CRM не завжди чудово реагує. Плинність персоналу ускладнює налагодження довгострокових, робочих відносин з конкретними членами служби підтримки. [3]

### 1.3.2 Close

Переваги: SMS, перегляд конвеєра, голос, звітування та маркетинг електронної пошти в одному місці. Close пропонує заздалегідь записану голосову пошту, чудову функцію економії часу для торгових представників, а функція передбачуваного набору дозволяє представникам автоматично набирати кілька номерів для підвищення продуктивності агента з продажу. На рисунку 1.3 зображено логотип компанії Close.



Рисунок 1.3 Логотип компанії Close

Close також пропонує безліч корисних інтеграцій з іншими платформами, включаючи Mailchimp, Zendesk, Facebook та Zapier. На рисунку 1.4 зображено головну сторінку сайту Close CRM.

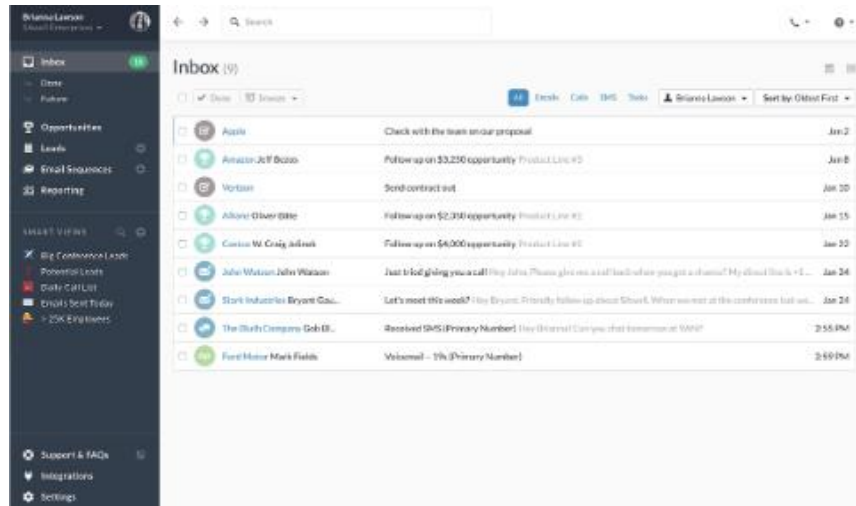


Рисунок 1.4 Скріншот головної сторінки сайту – Close

Недоліки: Імпорт масових даних CSV-файлів часто дає дублікати потенційних клієнтів, що вимагає очищення вручну. [4]

### 1.3.3 Agile CRM

Переваги: Повнофункціональний CRM із функціями автоматизації. З календарем зустрічей в Інтернеті, Agile CRM автоматизує всі майбутні голосові дзвінки та подальші дії. На рисунку 1.5 зображено логотип компанії Agile.



Рисунок 1.5 Логотип компанії Agile CRM

З можливістю перегляду контактів, управління проектами, звітуванням та можливостями часової шкали. Можливість додавання документів до угод, компаній, контактів та електронних листів у додатку. На рисунку 1.6 зображено головну сторінку сайту Agile CRM.

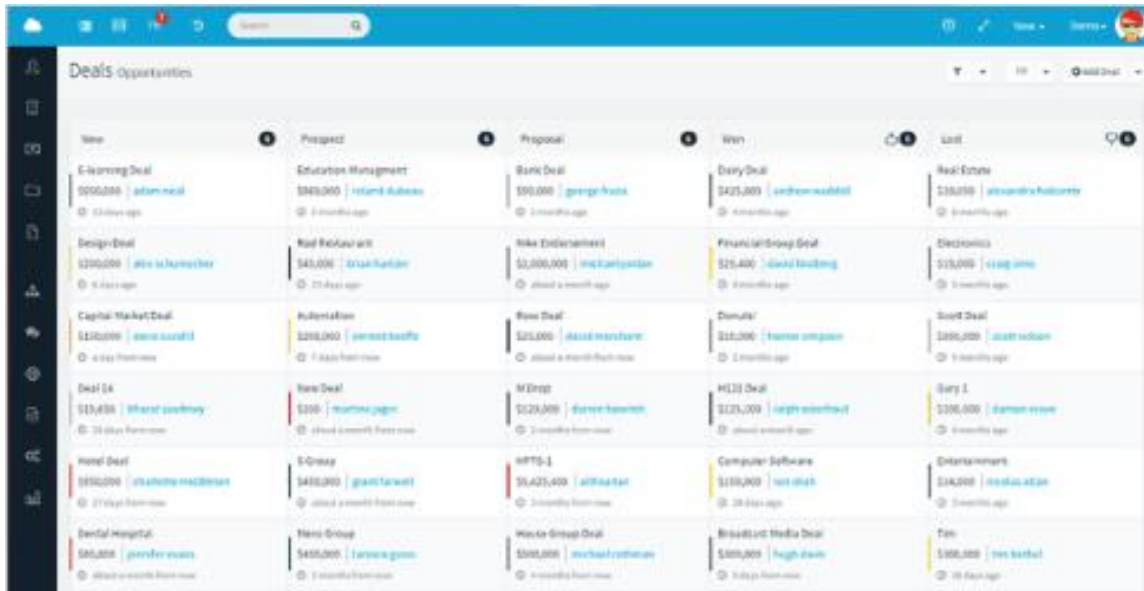


Рисунок 1.6 Скріншот головної сторінки сайту – Agile CRM

Недоліки: Ціна різко зростає між версіями, що може викликати неприємності та / або переривання угоди. Agile CRM не пропонує безліч варіантів налаштування. [5]

### 1.3.4 HubSpot CRM

HubSpot – американська компанія розробник та продавець програмних продуктів для маркетингу, продажів та обслуговування клієнтів. Її заснували Брайан Халліган та Дхармеш Шах у 2006 році. Її продукти та послуги спрямовані на надання інструментів для управління відносинами з клієнтами, маркетингу в



соціальних мережах, управління контентом, генерації потенційних клієнтів, веб-аналітики, оптимізації пошукових систем, чату та підтримки клієнтів.

Переваги: HubSpot - головне ім'я у світі CRM, пропонує безкоштовну платформу з основними функціями. Він обробляє основи процесу продажів, керуючи потенційними клієнтами та здійснює міжканальне відстеження взаємодії з клієнтами. Логотип компанії зображено на рисунку 1.7.



Рисунок 1.7 Логотип компанії HubSpot CRM

Інтеграція Zapier дозволяє обмінюватися інформацією між такими програмами, як Slack та Google Sheets. На рисунку 1.8 зображено головну сторінку сайту HubSpot.

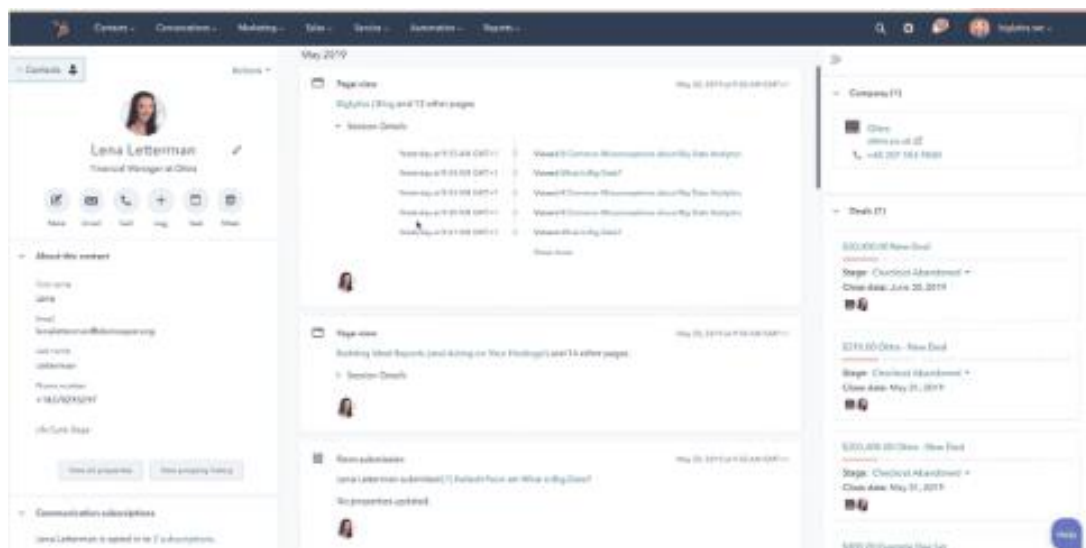


Рисунок 1.8 Скріншот головної сторінки сайту – HubSpot CRM

Платна надбудова "Центр продажів" для CRM значно підвищує попередню вартість за допомогою AI, глибших перспективних аналітик та автоматизації

високого класу.

Недоліки: Варіанти налаштування HubSpot CRM обмежені, що може бути проблемою, якщо робити щось під особливу нішу.

### 1.3.5 Zoho CRM

Zoho Corporation - індійсько-американська компанія з розробки програмного забезпечення зі штаб-квартирою в Ченнаї, Індія. Організація зосереджена на веб-інструментах для бізнесу та інформаційних технологіях, включаючи Інтернет-платформу для управління речами та набір програмного забезпечення для управління ІТ. Компанія була заснована в 1996 році Шрідхаром Вембу та Тоні Томасом у Плезантоні, штат Каліфорнія, і має свою присутність у семи місцях із головним офісом у Ченнаї та штаб-квартирою у Плезантоні. Логотип компанії Zoho зображений на рисунку 1.9.



Рисунок 1.9 Логотип компанії Zoho CRM

Переваги: CRM з соціальними медіа з автоматизацією робочих процесів та інструментами управління потенційними клієнтами. Інтегрується з Twitter, Facebook і Google+, щоб мати можливість своєчасно залучати потенційних клієнтів на кількох каналах.

Zoho CRM пропонує безкоштовну версію для двох користувачів.

Стандартний пакет Zoho забезпечує розширене прогнозування продажів та відстеження КРІ. Версія корпоративного верхнього рівня додає такі інструменти, як поетапне відстеження послідовності продажів та настроюване порівняння даних із сегментацією команд, користувачів, регіонів тощо. На рисунку 1.10 зображено головну сторінку сайту Zoho.

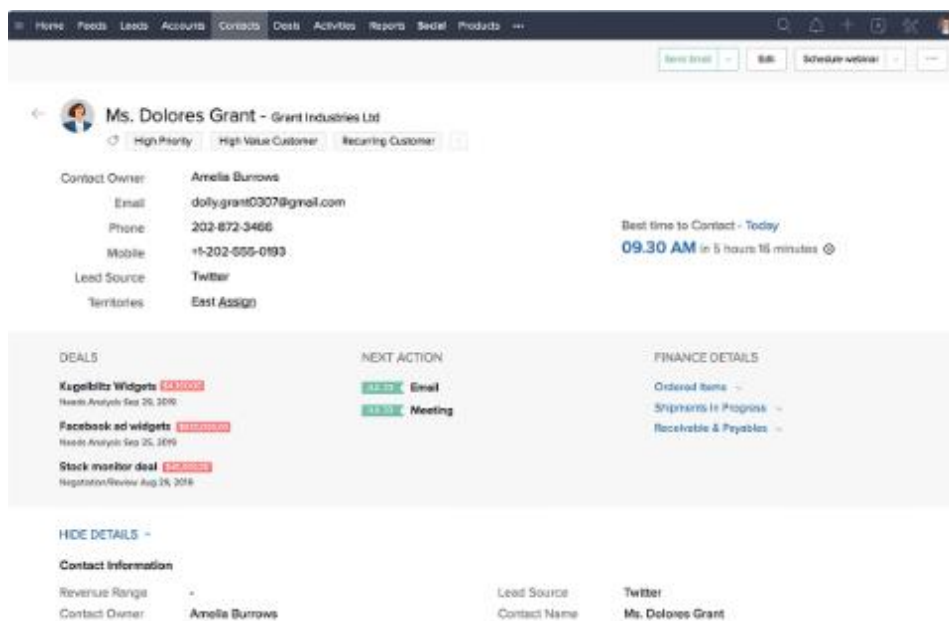


Рисунок 1.10 Скріншот головної сторінки сайту – Zoho

Недоліки: Для торгових представників, які покладаються на взаємодію один на одного та налагодження зв'язків, відсутність індивідуального відстеження електронної пошти та індивідуальних повідомлень про потенційних клієнтів може бути проблемою.

## **2 КОНСТРУЮВАННЯ ПРОГРАМНОЇ СИСТЕМИ**

### **2.1 Інструменти розробки**

Для розробки системи було вибрано середовище розробки PhpStorm, WebStorm, mySQLWorkbench, також обрано такі мови, як PHP, JavaScript, TypeScript, CSS, HTML. Для розробки системи було вибрано такі фреймворки, як Lumen, Angular.

#### **2.1.1 WebStorm**

WebStorm - це потужне та інтелектуальне середовище розробки, що надає допомогу в кодуванні JavaScript, HTML та CSS і покриває широкий спектр сучасних веб-технологій. WebStorm чудово обладнаний для складної розробки на стороні клієнта та розробки на сервері за допомогою Node.js.

Він забезпечує розумний аналіз коду, автозаповнення, функції рефакторингу, попередження помилок на ходу та багато іншого. Разом з підтримкою таких популярних фреймворків, як AngularJS та Meteor, а також інтегрованих інструментів для тестування, налагодження та аналізу коду та інтеграції з різними VCS, WebStorm підвищує продуктивність та виводить досвід розробки на абсолютно новий рівень.

#### **2.1.2 PhpStorm**

PhpStorm - комерційна, крос-платформна IDE для PHP, побудована чеською компанією JetBrains. PhpStorm надає редактор для PHP, HTML та JavaScript з

польовим аналізом коду, запобіганням помилкам та автоматизованим рефакторингом для PHP та JavaScript коду. Завершення коду PhpStorm підтримує PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1, 7.2, 7.3 і 7.4, включаючи генератори, програми, список у foreach, простори імен, закриття, риси та синтаксис короткого масиву. Він включає повноцінний редактор SQL із результатами запитів, які можна редагувати. PhpStorm написаний на Java. Користувачі можуть розширити IDE, встановивши плагіни, створені для PhpStorm, або написати власні плагіни. Програмне забезпечення також взаємодіє із зовнішніми джерелами, такими як XDebug. Усі функції, доступні в WebStorm, включені в PhpStorm, що додає підтримку PHP та баз даних. WebStorm постачається з попередньо встановленими плагінами JavaScript (наприклад, для Node.js).

### **2.1.3 mySQLWorkbench**

MySQL Workbench - це візуальний інструмент проектування баз даних, який інтегрує розробку, адміністрування, проектування, створення та обслуговування SQL в єдине цілісне середовище розробки системи баз даних MySQL. Він є наступником DBDesigner 4 від fabFORCE.net і замінює попередній пакет програм, MySQL GUI Tools Bundle.

### **2.1.4 PHP**

PHP - це мова сценаріїв загального призначення, особливо придатна для веб-розробки.

PHP-код зазвичай обробляється на веб-сервері інтерпретатором PHP. На веб-сервері результат інтерпретованого та виконаного PHP-коду - який може бути будь-

яким типом даних, наприклад, згенерований HTML або даними двійкових зображень - буде формувати цілу або частину відповіді HTTP. Існують різні системи веб-шаблонів, системи управління веб-контентом та веб-фреймворки, які можуть бути використані для організації або сприяння генерації такої відповіді. Крім того, PHP може бути використаний для багатьох завдань програмування поза веб-контекстом, таких як автономні графічні програми та роботизована система управління безпілотниками. Довільний PHP-код також може інтерпретуватися і виконуватися через інтерфейс командного рядка (CLI). [6]

### **2.1.5 JavaScript**

JavaScript часто скорочується як JS, є мовою програмування, яка відповідає специфікації ECMAScript. JavaScript є високорівневим. Він має синтаксис фігурних дужок, динамічне введення тексту, орієнтацію на об'єкти на основі прототипу та функції першого класу.

Разом з HTML та CSS, JavaScript є однією з основних технологій Всесвітньої мережі. JavaScript забезпечує інтерактивні веб-сторінки та є важливою частиною веб-додатків. Переважна більшість веб-сайтів використовують його для поведінки на стороні клієнта, і всі основні веб-браузери мають спеціальний механізм JavaScript для його виконання.

Як мова багатопарадигми, JavaScript підтримує керовані подіями, функціональні та імперативні стилі програмування. Він має інтерфейси прикладного програмування (API) для роботи з текстом, датами, регулярними виразами, стандартними структурами даних та об'єктною моделлю документа (DOM). Однак сама мова не включає жодного введення / виводу (вводу / виводу), наприклад мережевих, сховищних чи графічних засобів, оскільки середовище хоста (як правило, веб-браузер) забезпечує ці API.[7]

## 2.1.6 TypeScript

TypeScript - це мова програмування, розроблена та підтримувана корпорацією Майкрософт. TypeScript призначений для розробки великих додатків та перекомпіляції в JavaScript. Оскільки TypeScript є надмножиною JavaScript, існуючі програми JavaScript також є дійсними програмами TypeScript. TypeScript може використовуватися для розробки додатків JavaScript як для виконання на стороні клієнта, так і на стороні сервера (як у Node.js або Deno). Для перекомпіляції доступно кілька варіантів.

TypeScript підтримує файли визначень, які можуть містити інформацію про тип існуючих бібліотек JavaScript, подібно до того, як файли заголовків C++ можуть описувати структуру існуючих об'єктних файлів. Це дозволяє іншим програмам використовувати значення, визначені у файлах, ніби це статично набрані сутності TypeScript. Існують сторонні файли заголовків для популярних бібліотек, таких як jQuery, MongoDB та D3.js. Доступні також заголовки TypeScript для базових модулів Node.js, що дозволяє розробляти програми Node.js в межах TypeScript. Сам компілятор TypeScript написаний на TypeScript і компілюється в JavaScript. Він ліцензований за ліцензією Apache 2.0. TypeScript включено як першокласну мову програмування до Microsoft Visual Studio 2013 Update 2 та новіших версій, поряд із C # та іншими мовами Microsoft. Офіційне розширення дозволяє Visual Studio 2012 також підтримувати TypeScript.

## 2.1.7 CSS

Каскадні таблиці стилів (CSS) - це мова таблиці стилів, що використовується для опису презентації документа, написаного мовою розмітки, такою як HTML. CSS - це наріжна технологія Всесвітньої павутини, поряд з HTML та JavaScript.

CSS призначений для розділення презентації та вмісту, включаючи макет, кольори та шрифти. Це розділення може покращити доступність вмісту, забезпечити більшу гнучкість та контроль у специфікації характеристик презентації, дозволити декільком веб-сторінкам спільне форматування, вказавши відповідний CSS в окремому файлі .css, що зменшує складність та повторюваність структурного вмісту, а також дозволяє файл .css, який буде кешовано, щоб покращити швидкість завантаження сторінки між сторінками, що ділять файл, та його форматування. [8]

## **2.1.8 HTML**

HTML - аббревіатура, що розшифровується як Hyper Text Markup Language, яка використовується для створення веб-сторінок та веб-програм. Гіпертекст: HyperText означає "Текст у тексті". Текст має в собі посилання, є гіпертекстом. Кожного разу, коли натискати на посилання, яке приводить до нової веб-сторінки - натискати на гіпертекст. HyperText - це спосіб зв'язати між собою дві або більше веб-сторінок (документи HTML).

Мова розмітки: - це комп'ютерна мова, яка використовується для застосування конвенцій щодо розмітки та форматування текстового документа. Мова розмітки робить текст більш інтерактивним та динамічним.

## **2.2 Використані технології**

### **2.2.1 Lumen**

Lumen - це мікро-веб-фреймворк з відкритим кодом PHP, написаний на PHP, створений Тейлором Отуеллом як альтернатива Laravel. Вихідний код Lumen



розміщений на GitHub і ліцензований на умовах ліцензії MIT.

Нижче описані деякі вбудовані функції Lumen:

- Маршрутизація надається нестандартно в Lumen. Це включає основну маршрутизацію, параметри маршрутизації, іменовані маршрути та групи маршрутів, такі як проміжне програмне забезпечення.

- Аутентифікація не підтримує стан сеансу. Однак вхідні запити аутентифікуються за допомогою механізмів без громадянства, таких як маркери.

- Кешування здійснюється так само, як і в Laravel. Підтримуються драйвери кешу, такі як Database, Memcached та Redis. Наприклад, можна встановити пакет illuminate / redis через Composer, щоб використовувати кеш Redis з Lumen.

Помилки та ведення журналу реалізуються через бібліотеку Monolog, яка забезпечує підтримку різних обробників журналів.

### **2.2.2 Angular**

Angular - це платформа та фреймворк для побудови односторінкових клієнтських додатків за допомогою HTML та TypeScript. Angular написаний TypeScript. Він реалізує основну та додаткову функціональність як набір бібліотек TypeScript.

Архітектура програми Angular спирається на певні фундаментальні концепції. Основними будівельними елементами Angular framework є Angular компоненти, які організовані в NgModules. NgModules збирають відповідний код у функціональні набори;

Додаток Angular визначається набором NgModules. Додаток завжди має принаймні кореневий модуль, який дозволяє завантажувати, і зазвичай має набагато більше модулів функцій. [9]

### 2.2.3 REST API

REST - це скорочення від Representational State Transfer. Це архітектурний стиль для розподілених гіпермедійних систем.

Як і будь-який інший архітектурний стиль, у REST також є свої 6 основних обмежень, які необхідно виконати, якщо інтерфейс потрібно називати RESTful. Ці принципи наведені нижче:

Клієнт-сервер - відокремлюючи проблеми користувальницького інтерфейсу від проблем зберігання даних, покращується переносимість користувальницького інтерфейсу на декількох платформах та покращується масштабованість, спрощуючи компоненти сервера.

Stateless - кожен запит від клієнта до сервера повинен містити всю інформацію, необхідну для розуміння запиту, і не може використовувати будь-який збережений контекст на сервері. Тому стан сеансу повністю тримається на клієнті.

Кешування - обмеження кешу вимагають, щоб дані у відповіді на запит неявно або явно позначалися як кешовані чи некашовані. Якщо відповідь можна кешувати, тоді кеш-пам'ять клієнта отримує право повторно використовувати ці дані відповідей для подальших, еквівалентних запитів.

Уніфікований інтерфейс - застосовуючи принцип загальності до інтерфейсу компонента, спрощується загальна архітектура системи та покращується видимість взаємодій. Для того, щоб отримати єдиний інтерфейс, для керування поведінкою компонентів необхідні різні архітектурні обмеження. REST визначається чотирма обмеженнями інтерфейсу:

- ідентифікація ресурсів;
- маніпулювання ресурсами через представництва;
- самоописові повідомлення;
- гіпермедіа, як двигун стану застосування.

Багатошарова система - стиль багатошарової системи дозволяє архітектурі складатися з ієрархічних шарів, обмежуючи поведінку компонентів, так що кожен

компонент не може "бачити" далі безпосереднього шару, з яким вони взаємодіють.

Code on demand (необов'язково) - REST дозволяє розширити функціональність клієнта, завантажуючи та виконуючи код у формі аплетів або сценаріїв. Це спрощує клієнтів, зменшуючи кількість функцій, необхідних для попередньої реалізації. [10]

Ключовим абстрагуванням інформації в REST є ресурс. Будь-яка інформація, яку можна назвати, може бути ресурсом: документ або зображення, тимчасова служба, колекція інших ресурсів, невіртуальний об'єкт (наприклад, особа) тощо. REST використовує ідентифікатор ресурсу для ідентифікації конкретного ресурсу, який бере участь у взаємодії між компонентами.

Стан ресурсу на будь-якій конкретній митці часу називається поданням ресурсу. Представлення складається з даних, метаданих, що описують дані, та гіпермедіа-посилань, які можуть допомогти клієнтам перейти до наступного бажаного стану. Формат даних представлення відомий як тип носія. Тип носія визначає специфікацію, яка визначає спосіб обробки подання. По-справжньому RESTful API виглядає як гіпертекст. Кожна адресована одиниця інформації містить адресу або явно (наприклад, атрибути посилання та ідентифікатора), або неявно (наприклад, отримана з визначення типу носія та структури представлення).

У таблиці 2.1, наведені рекомендовані значення повернення основних методів HTTP у поєднанні з URI ресурсами:

Таблиця 2.1 – Рекомендовані значення повернення основних методів HTTP

HTTP Методи	CRUD	Вся колекція	Конкретний товар
Post	Створити	201 (Created), заголовок "місцезнаходження" із посиланням на / customers / {id}, що містить новий ідентифікатор	404 (Not Found), 409 (Conflict), якщо ресурс уже існує ..

Продовження таблиці 2.1

Get	Прочитати	200 (OK), список клієнтів. Використовуйте пагінацію, сортування та фільтрування для навігації у великих списках.	200 (OK), одиничний клієнт. 404 (Not Found), якщо ідентифікатор не знайдений або недійсний.
Put	Оновити / замінити	405 (Method Not Allowed), якщо не хочете оновити / замінити кожен ресурс у всій колекції.	200 (OK) або 204 (No Content). 404 (Not Found), якщо ідентифікатор не знайдений або недійсний.
Patch	Оновити / змінити	405 (Method Not Allowed), якщо не хочете змінити саму колекцію.	200 (OK) або 204 (No Content). 404 (Not Found), якщо ідентифікатор не знайдений або недійсний.
Delete	Видалити	405 (Method Not Allowed), якщо не хочете видалити всю колекцію, - часто не бажано.	200 (OK). 404 (Not Found), якщо ідентифікатор не знайдений або недійсний.

Методи HTTP складають основну частину обмеження “єдиного інтерфейсу” і надають відповідник дії ресурсу на основі іменника. Основними або найбільш часто використовуваними методами HTTP є POST, GET, PUT, PATCH і DELETE. Вони відповідають операціям створення, читання, оновлення та видалення (або CRUD) відповідно. Є й ряд інших методів, але вони вживаються рідше. З цих менш частих методів OPTIONS та HEAD використовуються частіше за інших.

Найчастіше метод POST використовується для «Створення» нових ресурсів. Зокрема, він використовується для створення підпорядкованих ресурсів. Тобто, підпорядкований якомусь іншому (наприклад, батьківському) ресурсу. Іншими словами, під час створення нового ресурсу, POST для батьківського і сервісу

підключається про зв'язування нового ресурсу з батьківським, присвоєння ідентифікатора (нового ресурсу URI) тощо.

Після успішного створення необхідно повернути статус HTTP 201, повернувши заголовок Location із посиланням на новостворений ресурс зі статусом HTTP 201.

POST не є ні безпечним, ні ідемпотентним. Тому рекомендується для неімпотентних запитів на ресурси. Виконання двох однакових запитів POST, швидше за все, призведе до двох ресурсів, що містять однакову інформацію.

Метод HTTP GET використовується для «Читання» (або отримання) подання ресурсу. У "щасливому" (або помилковому) шляху GET повертає подання у форматі XML або JSON та код відповіді HTTP 200 (OK). У випадку помилки він найчастіше повертає 404 (Not Found) або 400 (ПОГОРИЙ ЗАПИТ).

Відповідно до конструкції специфікації HTTP, запити GET (разом із HEAD) використовуються лише для читання даних, а не для їх зміни. Тому при такому застосуванні вони вважаються безпечними. Тобто їх можна викликати без ризику модифікації даних або пошкодження - виклик одного разу має такий самий ефект, як виклик 10 разів, або взагалі жодного. Крім того, GET (і HEAD) є ідемпотентним, що означає, що створення кількох однакових запитів має однаковий результат, як один запит.

Не потрібно передавати небезпечні операції через GET - він ніколи не повинен змінювати будь-які ресурси на сервері.

PUT найчастіше використовується для «Оновлення» можливостей, PUT-и до відомого URI ресурсу з тілом запиту, що містить нещодавно оновлене подання вихідного ресурсу.

Однак PUT також може використовуватися для створення ресурсу в тому випадку, коли ідентифікатор ресурсу вибирається клієнтом, а не сервером. Іншими словами, якщо PUT - це URI, що містить значення неіснуючого ідентифікатора ресурсу. Знову ж таки, тіло запиту містить подання ресурсу. Багато хто вважає це заплутаним. Отже, цим способом створення слід користуватися помірковано.

Крім того, необхідно використовувати POST для створення нових ресурсів та надання визначеного клієнтом ідентифікатора у поданні тіла - імовірно, до URI, який не включає ідентифікатор ресурсу.

Після успішного оновлення повернеться 200 (або 204, якщо не повертає жодного вмісту в тілі) з PUT. Якщо використовувати PUT для створення, повернеться статус HTTP 201 при успішному створенні. Тіло у відповіді є необов'язковим - якщо споживає більше смуги пропускання.

Не потрібно повертати посилання через заголовок Location у випадку створення, оскільки клієнт вже встановив ідентифікатор ресурсу.

PUT не є безпечною операцією, оскільки вона змінює (або створює) стан на сервері, але є ідемпотентною. Іншими словами, якщо створити або оновити ресурс за допомогою PUT, а потім повторити той самий виклик, ресурс все ще є і все ще має такий самий стан, як і під час першого виклику. Якщо, наприклад, виклик PUT на ресурсі збільшує лічильник у ресурсі, виклик більше не є ідемпотентним. Іноді це трапляється, і цього може бути достатньо, щоб підтвердити, що дзвінок не є ідемпотентним. Однак рекомендується залишати запити PUT ідемпотентними. Настійно рекомендується використовувати POST для неідемпотентних запитів.

PATCH використовується для «Оновлення» можливостей. Запит PATCH повинен містити лише зміни у ресурсі, а не повний ресурс. Це нагадує PUT, але тіло містить набір інструкцій, що описують, як слід модифікувати ресурс, який зараз знаходиться на сервері, для створення нової версії. Це означає, що тіло PATCH має бути не просто модифікованою частиною ресурсу, а якоюсь мовою виправлень, таких як JSON Patch або XML Patch. PATCH не є ні безпечним, ні ідемпотентним. Однак запит PATCH може бути виданий таким чином, щоб бути ідемпотентним, що також допомагає запобігти поганим результатам зіткнень між двома запитами PATCH на одному ресурсі в подібні часові рамки. Зіткнення від декількох запитів PATCH можуть бути більш небезпечними, ніж зіткнення PUT, оскільки деякі формати виправлень повинні працювати з відомої базової точки, інакше вони пошкодять ресурс. Клієнти, що використовують цей тип програм для виправлень, повинні використовувати умовний запит, таким чином, що запит не вдасться

виконати, якщо ресурс був оновлений з моменту останнього доступу клієнта до ресурсу. Наприклад, клієнт може використовувати сильний ETag у заголовку If-Match на запит PATCH.

DELETE досить легко зрозуміти. Він використовується для «Видалення» ресурсу, визначеного URI. Після успішного видалення повернеться статус HTTP 200 (OK) разом із тілом відповіді, можливо, представленням видаленого елемента (часто потрібна занадто велика пропускна здатність) або загорнутою відповіддю (див. Значення повернення нижче). Або це, або повертає статус HTTP 204 (NO CONTENT) без тіла відповіді. Іншими словами, рекомендовані відповіді - стан 204 без основного тексту, або відповідь у стилі JSEND та статус HTTP 200.

HTTP-специфікація, операції DELETE є ідемпотентними. Якщо ВИДАЛИТИ ресурс, його буде видалено. Неодноразові виклики DELETE на цьому ресурсі закінчуються тим самим: ресурс зник. Якщо виклик DELETE скаже, зменшить лічильник (усередині ресурсу), виклик DELETE перестане бути ідемпотентним. Як вже згадувалося раніше, статистика використання та вимірювання можуть бути оновлені, враховуючи ідемпотент служби, якщо дані ресурсів не змінюються. Рекомендується використовувати POST для неідемпотентних запитів ресурсів. Однак є застереження щодо DELETE ідемпотенції. Виклик DELETE на ресурсі вдруге часто повертає 404 (Not Found), оскільки його вже було видалено і тому більше не можна знайти. Це, за деякими думками, робить операції DELETE більше не ідемпотентними, однак кінцевий стан ресурсу однаковий. Повернення 404 є прийнятним і точно повідомляє статус дзвінка. [11]

## 2.2.4 MVC

Система проектувалась на основі архітектури MVC.

Model-View-Controller (MVC) - це архітектурний шаблон, який розділяє додаток на три основні логічні компоненти: модель, вигляд та контролер. Кожен із

цих компонентів створений для обробки конкретних аспектів розробки програми. MVC – один з найбільш часто використовуваних галузевих стандартів веб-розробки для створення масштабованих та розширюваних проектів.

Нижче наведені компоненти MVC:

Model - компонент Model відповідає всій логіці даних, з якою працює користувач. Це може представляти або дані, що передаються між компонентами View та Controller, або будь-які інші дані, пов'язані з бізнес-логікою. Наприклад, об'єкт "Клієнт" буде отримувати інформацію про клієнта з бази даних, маніпулювати нею та оновлювати дані назад до бази даних або використовувати її для рендерингу даних.

View - компонент View використовується для всієї логіки інтерфейсу програми. Наприклад, подання Клієнт включатиме всі компоненти інтерфейсу, такі як текстові поля, випадаючі меню тощо, з якими взаємодіє кінцевий користувач.

Controller - компонент виступає інтерфейсом між компонентами Model і View для обробки всієї бізнес-логіки та вхідних запитів, маніпулювання даними за допомогою компонента Model та взаємодії з поданнями для надання кінцевого результату. Наприклад, контролер замовника буде обробляти всі взаємодії та входи з перегляду замовника та оновлювати базу даних за допомогою моделі замовника. Той самий контролер буде використовуватися для перегляду даних Клієнта. [12] На рисунку 2.1 зображена архітектура проектування MVC.



Рисунок 2.1 – Архітектура проектування MVC



Особливостями MVC є:

- Легка тестованість;
- Повний контроль над HTML, а також над URL-адресами;
- Чітке розділення логіки: модель, вигляд, контролер;
- Поділ прикладних завдань, а саме. Бізнес-логіка, логіка UI та логіка

введення;

- Маршрутизація URL-адрес для SEO URL-адрес;
- Підтримка для тестової розробки (TDD).

У будь-чому і будь-що має недоліки, так само і ця архітектура проектування MVC при використанні має свої недоліки, а саме:

- Цю модель важко прочитати, змінити, протестувати та повторно використовувати;

- Навігаційна система може дещо ускладнюватись, оскільки вводить нові рівні абстракції, що вимагає від користувачів адаптації до критеріїв декомпозиції MVC;

- Відсутність формальної підтримки;

- Підвищена складність та неефективність даних;

- Складність використання MVC із сучасним інтерфейсом користувача;

- Існує потреба в тому, щоб кілька програмістів проводили паралельне програмування;

- Потрібні знання багатьох технологій;
- Обслуговування безлічі кодів в Controller.

### **2.2.5 OAuth**

OAuth - це відкритий стандартний протокол або рамка авторизації, який надає програмам можливість «безпечного призначеного доступу». Наприклад, можливо повідомити Facebook, що ESPN.com може отримувати доступ до профілю

користувача або публікувати оновлення на хронології користувача без необхідності вводити ESPN пароль у Facebook. Це головним чином мінімізує ризик: у разі порушення ESPN пароль користувача Facebook залишається безпечним.

OAuth не передає дані паролів, а використовує маркери авторизації, щоб підтвердити особу між споживачами та постачальниками послуг. OAuth - це протокол автентифікації, який дозволяє схвалювати одну програму, яка взаємодіє з іншою від імені користувача, не видаючи його пароль.

OAuth 1.0 вперше був випущений у 2007 році як метод авторизації для інтерфейсу програми програми Twitter. У 2010 р. Робоча група IETF OAuth опублікувала перший проект протоколу OAuth 2.0. Як і оригінальний OAuth, OAuth 2.0 надає користувачам можливість надати стороннім програмам доступ до веб-ресурсів без спільного використання пароля. Однак це абсолютно новий протокол і не сумісний із OAuth 1.0. Оновлені функції включають новий потік коду авторизації для розміщення мобільних додатків, спрощені підписи та короточасні маркери з довготривалими авторизаціями.

Потік авторизації в типовій реалізації OAuth 2.0 - це шестиступеневий процес. У наведеному нижче прикладі програма для створення онлайн-календаря повинна мати доступ до фотографій користувача, що зберігаються їх на Google Диску:

Програма створення календаря (клієнт) запитує авторизацію для доступу до захищених ресурсів, у цьому випадку до файлів зображень, що належать користувачеві (власнику ресурсу), направляючи користувача до кінцевої точки авторизації.

Власник ресурсу автентифікує та авторизує запит на доступ до ресурсу з програми, а кінцева точка авторизації повертає клієнту дозвіл авторизації. Протокол OAuth 2.0 визначає чотири типи грантів: Код авторизації, Клієнтські дані, Код пристрою та Оновлення маркера.

Потім клієнт запитує маркер доступу у сервера авторизації, представляючи дозвіл авторизації, повернений із кінцевої точки авторизації, разом із

автентифікацією власної ідентичності кінцевій точці маркера. Кінцева точка маркера - це URL-адреса, така як `https://your_domain/oauth2/token`.

Якщо ідентифікація клієнта автентифікована і надання авторизації є дійсним, сервер авторизації або постачальник автентифікації - в цьому випадку Сервер авторизації Google - видасть клієнту маркер доступу.

Тепер клієнт може запитувати захищені ресурси від сервера ресурсів - Google Drive у цьому прикладі - представляючи маркер доступу для автентифікації.

Якщо маркер доступу дійсний, сервер ресурсів повертає запитувані ресурси до програми створення календаря (клієнта).

Тепер програма для створення календаря може отримувати доступ та імпортувати фотографії користувача для створення календаря. Залежно від типу гранту, виданого на другому кроці, потік авторизації може дещо відрізнятись. Однак він все ще значною мірою слідує цим основним крокам. [13] На рисунку зображена схема роботи OAuth. Схема роботи OAuth зображена на рисунку 2.2.

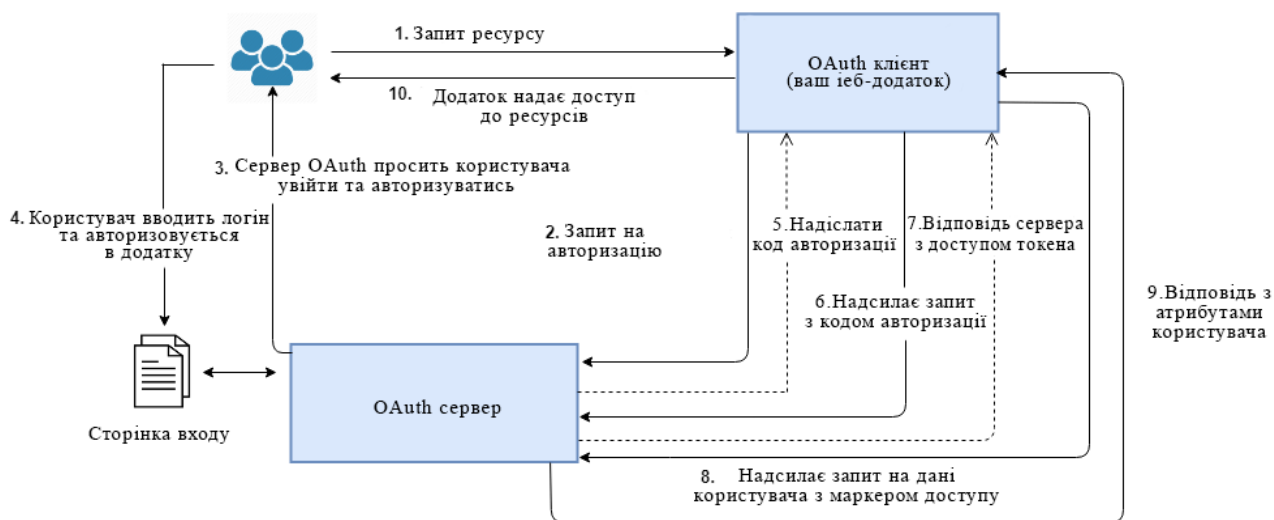


Рисунок 2.2 – Схема роботи OAuth

До OAuth HTTP був базовим стандартом автентифікації, де користувачеві пропонується ввести ім'я користувача та пароль для доступу до кожної програми. Веб-сайти запропонують ввести ім'я користувача та пароль безпосередньо у

форму, і вони ввійдуть у дані користувача(наприклад, обліковий запис Gmail), як користувач.

Базова автентифікація все ще використовується як примітивна форма автентифікації API для додатків на стороні сервера, де замість того, щоб надсилати ім'я користувача та пароль серверу з кожним запитом, користувач надсилає ідентифікатор ключа API та секрет.

На відміну від вищезазначеного, OAuth дозволяє аутентифікацію за допомогою маркерів доступу, що є більш безпечним, оскільки не використовується спільне використання паролів. [14]

### **2.2.6 JSON Web Token**

JSON Web Token (JWT) - це відкритий стандарт ( RFC 7519 ), який визначає компактний та автономний спосіб безпечної передачі інформації між сторонами як об'єкт JSON. Цю інформацію можна перевірити та довіряти їй, оскільки вона має цифровий підпис. JWT можна підписати, використовуючи секрет (з алгоритмом HMAC ) або пару відкритих / приватних ключів, використовуючи RSA або ECDSA.

Незважаючи на те, що JWT можуть бути зашифровані, щоб також забезпечити таємницю між сторонами. Підписані маркери можуть перевірити цілісність вимог, що містяться в ньому, тоді як зашифровані маркери приховують ці вимоги від інших сторін. Коли маркери підписуються за допомогою пар відкритого / приватного ключів, підпис також засвідчує, що лише сторона, що тримає закритий ключ, та, що підписала його. [15]

Ось кілька сценаріїв, коли корисні веб-маркери JSON:

Авторизація : Це найпоширеніший сценарій використання JWT. Після того, як користувач увійшов в систему, кожен наступний запит включатиме JWT, що дозволить користувачеві отримати доступ до маршрутів, послуг та ресурсів,

дозволені цим маркером. Єдиний вхід - це функція, яка в наш час широко використовує JWT, через невеликі накладні витрати та можливість легкого використання в різних доменах.

Обмін інформацією : JWT – є чудовим способом безпечної передачі інформації . Оскільки JWT можна підписувати - наприклад, використовуючи пари відкритого / приватного ключів - можна бути впевненим, що відправники є тими, хто каже, що вони є. Окрім того, оскільки підпис обчислюється за допомогою заголовка та корисного навантаження, користувач також може переконатися, що вміст не було підроблено.

У своїй компактній формі веб-маркери JSON складаються з трьох частин, розділених крапками ( . ), а саме:

Заголовок;

Корисне навантаження;

Підпис. [16]

### 3 РОЗРОБКА СИСТЕМИ

В ході написання кваліфікаційної роботи було розроблено CRM-систему для Owl Photobook.

Дана система відрізняється від усіх інших CRM-систем тим, що вона спеціалізована під певну нішу. Перед розробкою системи було прописане технічне завдання. Вона відповідає усім поставленим вимогам і виконує усі заявлені функції, а саме:

- Автентифікація користувача;
- Встановлення паролю користувача;
- Зміна паролю користувача;
- Відновлення паролю, з надсиланням листа на пошту;
- Редагування персональної інформації;
- Створення, редагування, видалення клієнтів;
- Створення, редагування, видалення типу клієнтів;
- Отримання списку клієнтів;
- Отримання списку типів клієнтів;
- Перегляд статистики сайту;
- Менеджмент замовлень;
- Створення нового замовлення;
- Редагування замовлення;
- Оновлення статусу замовлення;
- Видалення замовлення;
- Створення адреси доставки клієнтів;
- Менеджмент ролей.

Основною вимогою до системи була відмовостійкість, адже це один з найважливіших показників будь-якої сервісу. Через те, що навантаження на сервер буде постійно зростати, необхідно було забезпечити коректну обробку інформації, а також побудувати правильну архітектуру для взаємодії компонентів.

База даних на сервері здійснює швидку вибірку елементів та їх обробку.

База даних є нормалізованою, тобто відповідає наступним вимогам:

- Кожна таблиця у базі даних має свій основний ключ;
- Кожен атрибут мати одне значення;
- Групи не повторюються;
- Схема БД відповідає вимогам першої нормальної форми;
- Схема БД відповідає усім вимогам другої нормальної форми;
- Дані, які повторюються винесені в окремі таблиці;
- Поля, що не залежать від основного ключа та від будь-якого іншого поля

винесені в окрему таблицю.

В схемі БД не існує нетривіальних та багатозначних залежностей множин атрибутів та залежностей об'єднання, які не витікатимуть із обмежень ключів. Також, не функціональні вимоги були поставлені у відповідності до потреб користувача системи. Інтерфейс є інтуїтивно зрозумілим для користувача.

Також багато увагу приділено вимогам до інтерфейсу користувача. Дизайн інтерфейсу користувача - це процес, який дизайнери використовують для побудови інтерфейсів у програмному забезпеченні або комп'ютеризованих пристроях, зосереджуючи увагу на зовнішньому вигляді чи стилі. Дизайнери прагнуть створити інтерфейси, які користувачі знайдуть простими у користуванні та приємними. Дизайн інтерфейсу стосується графічних користувацьких інтерфейсів та інших форм, наприклад, інтерфейсів, що керуються голосом.

Дизайн інтерфейсу користувача (UI) зосереджується на передбаченні того, що користувачеві потрібно буде зробити, та забезпеченню того, щоб він містив елементи, до яких легко отримати доступ. Інтерфейс користувача об'єднує концепції дизайну взаємодії, візуального дизайну та інформаційної архітектури.

Інтерфейс користувача є безпечним, зручним та розширюваним. Вимоги безпеки включають необхідність захисту інформації про авторизацію від несанкціонованого доступу, забезпечення конфіденційності клієнта та відстеження поточних платежів. Інтерфейс користувача є ефективним у швидкості, і у використанні.

### 3.1 Функціонал програмного продукту

При вході у систему необхідно пройти авторизацію – скрін зі сторінкою авторизації зображений на рисунку 3.1.

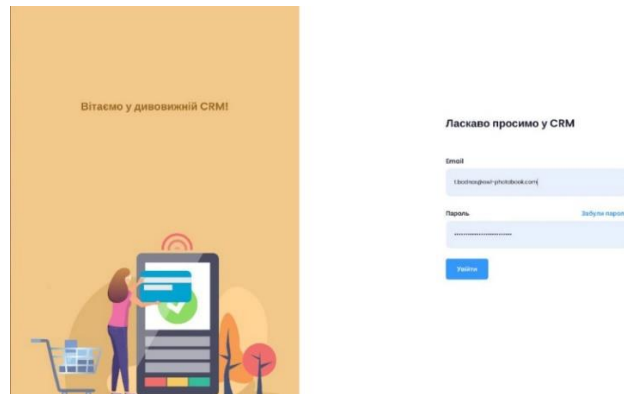


Рисунок 3.1 – Скрін сторінки авторизації

Якщо адміністратор забув пароль - він може здійснити відновлення паролю, з надсиланням листа на електронну пошту, який зображений на рисунку 3.2.

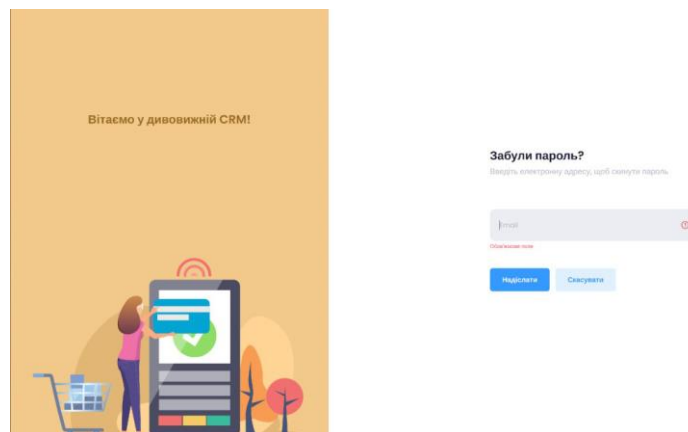


Рисунок 3.2 – Скрін сторінки відновлення паролю

Адміністратор має можливість змінити дані власного акаунту, такі як: Імя, Прізвище, мобільний телефон, Email, дата народження, пароль. Скрін персональної



сторінки зображений на рисунку 3.3.

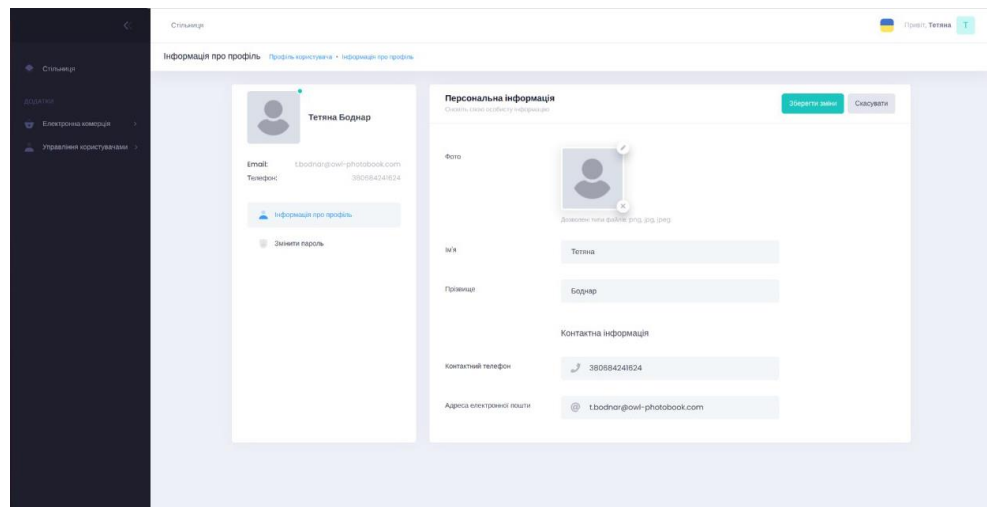


Рисунок 3.3 – Скрін персональної сторінки адміністратора

У системі можна добавляти та видаляти клієнтів, а також редагувати їхні дані. У системі можна легко здійснювати пошук клієнтів по базі даних та здійснювати фільтрування за типом. На рисунку 3.4 зображений скрін сторінки зі списком клієнтів.

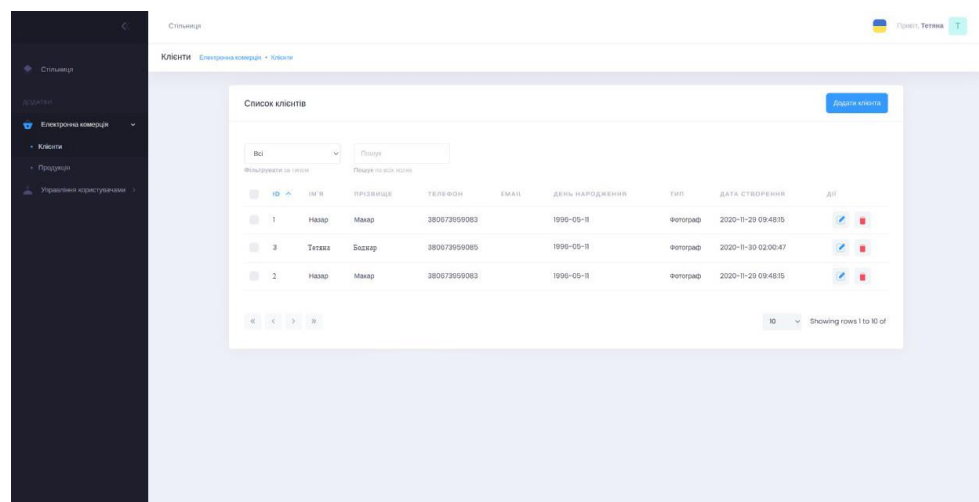


Рисунок 3.4 – Скрін сторінки «Список клієнтів»

Додавання користувачів відбувається за допомогою кнопки «Додати клієнта» в верхньому правому куті у вкладці «Клієнти». Поле «Додати клієнта» зображене на скріні 3.5.

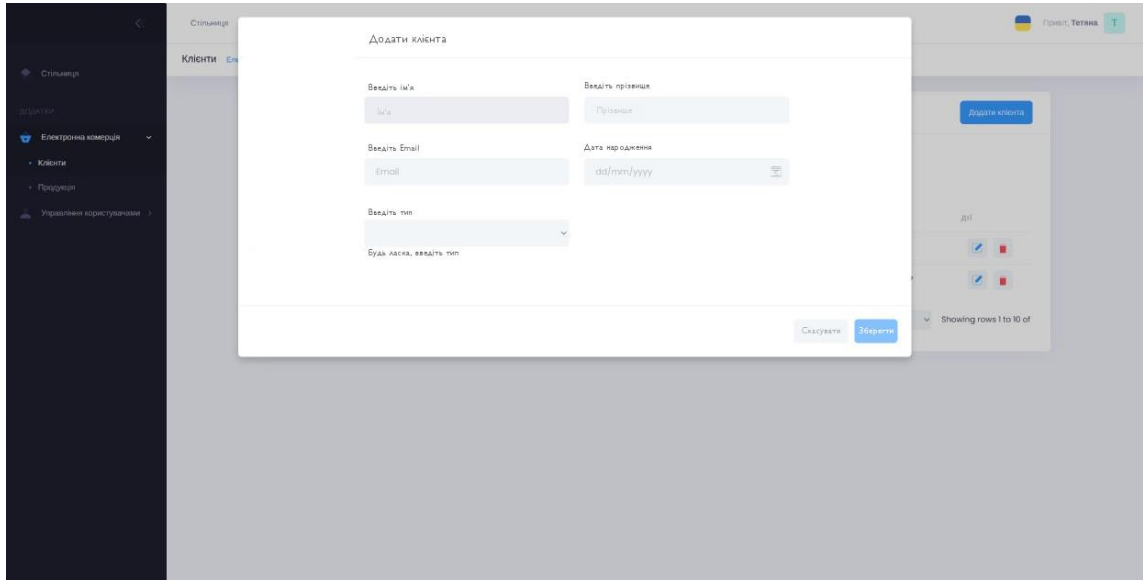


Рисунок 3.5 – Скрін вікна «Додати клієнта»

На головній сторінці – Стельниці сайту користувач може переглядати статистику замовлень, статистику відвідування сайту, поточну кількість клієнтів, щоденний, щомісячний, щорічний дохід. Також на головній сторінці користувач може бачити сповіщення щодо нових зареєстрованих користувачів, нові замовлення. При бажанні користувач системи може експортувати певні статистичні таблиці.

Користувач системи може створювати замовлення, редагувати його деталі, змінювати (оновлювати) статус, видаляти замовлення.

Також користувач системи (адміністратор) може змінювати типи користувачів, присвоювати деяким клієнтам тип «Фотограф», що при замовленні надаватиме клієнтам з цим статусом знижку.

Адміністратор може присвоювати іншим користувач ролі, які будуть визначати права доступу.

### 3.2 Тестування системи

Нижче наведені головні причини тестування CRM, та потенційні дефекти, яких можна уникнути:

- Низький рівень прийнятності користувачами;
- Неправильні або відсутні дані про клієнтів, неправильна класифікація клієнтів, а також повільний або незручний користувальницький інтерфейс можуть перешкоджати продажам, обслуговуванню клієнтів та маркетинговим працівникам переходити до ряду найбільш проблемних CRM-функцій або взагалі використовувати CRM.

Незадоволені клієнти - роботи з клієнтами можуть ставати на заваді пропущені або невиправдані повідомлення про зустрічі з клієнтами; неправильне відображення даних про повернення та скарги; затримка доставки через неправильну оплату, продаж або статус доставки; неправильні електронні листи, отримані неправильними клієнтами. Усі ці дефекти CRM можуть продовжувати розчаровувати клієнтів.

#### Відсутність видимості управління споживачами

Можливо, не вдасться отримати точну статистику від CRM через непослідовні, неадекватно змінені дані або неправильний розрахунок та класифікацію даних про покупки клієнтів, спричинені поганою інтеграцією CRM із ERP, фінансовим програмним забезпеченням, системою управління документами, сховищем даних або його несправностями.

#### Витік інформації

Якщо CRM не повністю відповідає регіональним або міжнародним нормам безпеки даних, це може призвести до того, що конфіденційна ділова інформація або конфіденційні дані клієнтів будуть піддані несанкціонованим користувачам, і призведе до судових позовів та штрафів. Чим більше відділів та службовців використовують CRM, тим гострішою стає необхідність перевірки доступу на основі ролей.

Зважаючи на усі ці можливі нюанси необхідно провести тестування системи, яке захистить як і користувачів з клієнтської сторони, так і користувачів зі сторони компанії. Нижче наведені тест-кейси, які здійснились над CRM-системою і її функціональними частинами:

Тестування юзабіліті - це метод, що використовується для оцінки того, наскільки зручним є у користуванні веб-сайт. Тести проводяться з реальними користувачами, щоб визначити, наскільки "придатним для використання" чи "інтуїтивно зрозумілим" є веб-сайт та наскільки легко користувачам досягти своїх цілей.

Ключова відмінність між тестуванням юзабіліті та традиційним тестуванням (тестування помилок, прийнятне тестування тощо) полягає в тому, що тестування юзабіліті проводиться з реальними користувачами чи замовниками продукту. Хоча традиційне тестування може проводити розробник, дизайнер або менеджер проекту, тестування зручності усуває будь-які упередження шляхом збору відгуків безпосередньо від кінцевого користувача.

Існує кілька різних типів перевірки юзабіліті або причин для проведення досліджень юзабіліті:

Порівняльне тестування юзабіліті: використовується для порівняння зручності використання одного веб-сайту з іншим. Порівняльні тести зазвичай використовуються для порівняння веб-сайту з аналогічними сайтами або сайтами-конкурентами, однак його також можна використовувати для порівняння двох дизайнів, щоб встановити, який забезпечує найкращий досвід користувачів.

Дослідницьке тестування юзабіліті - перш ніж випустити новий продукт, дослідницьке тестування юзабіліті може встановити, який зміст та функціональність повинен включати новий продукт для задоволення потреб своїх користувачів. Користувачі перевіряють цілий ряд різноманітних послуг, де їм пропонують реалістичні сценарії, які допомагають висвітлити будь-які прогалини на ринку, якими можна скористатися, та продемонструвати, де зосередити свої зусилля у розробці.

Оцінка зручності використання - це тест на нову або оновлену послугу до або після запуску. Цей тест на юзабіліті знайомить користувачів з новим дизайном, щоб переконатися, що він інтуїтивно зрозумілий у використанні та забезпечує позитивний досвід роботи. Метою оцінки зручності є забезпечення того, щоб будь-які потенційні проблеми були висвітлені та виправлені до запуску продукту.

Є багато переваг тестування зручності використання, включаючи:

- зворотній зв'язок безпосередньо з цільовою аудиторією;
- внутрішні дебати можна вирішити, протестувавши проблему, щоб побачити, як користувачі реагують на різні обговорювані варіанти проблеми та потенційні проблеми висвітлюються перед запуском продукту.

Бізнес-переваги тестування юзабіліті можна побачити в кінці проекту:

- це збільшує ймовірність використання та повторного використання;
- це мінімізує ризик виходу товару з ладу;
- користувачі можуть краще досягти своїх цілей, що призводить до досягнення бізнесом своїх цілей;

Недоліком такого методу тестування є: тестування зручності використання дає багато переваг, але є кілька недоліків у використанні цієї методології, на які слід звернути увагу. По-перше, тестування не є на 100% репрезентативним сценарієм реального життя. Крім того, тестування юзабіліті є в основному якісним, тому не дає великих зразків зворотного зв'язку, які може містити опитувальник, але відгуки можуть бути набагато точнішими та проникливішими.

Функціональне тестування - це процес забезпечення якості (QA) та різновид тестування «чорних ящиків», яке базує свої тестові приклади на специфікаціях програмного компонента, що тестується. Функції перевіряються шляхом подачі на них вхідних даних та вивчення вихідних даних, а внутрішня структура програми розглядається рідко. Функціональне тестування проводиться для оцінки відповідності системи або компонента зазначеним функціональним вимогам. Функціональне тестування зазвичай описує те, що робить система.

Оскільки функціональне тестування є різновидом чорного ящика, функціональність програмного забезпечення можна перевірити, не знаючи

внутрішньої роботи програмного забезпечення. Це означає, що тестувальникам не потрібно знати мови програмування або як впроваджено програмне забезпечення. Це, в свою чергу, може призвести до зменшення упередженості розробника (або упередження підтвердження) при тестуванні, оскільки тестер не брав участі у розробці програмного забезпечення.

Функціональне тестування перевіряє фрагмент функціональності всієї системи.

Функціональне тестування відрізняється від системного тестування тим, що функціональне тестування "перевіряє програму, перевіряючи її на відповідність ... проектним документам або специфікації (специфікаціям)", а системне тестування "перевіряє програму, перевіряючи її щодо опублікованого користувача або системні вимоги".

Тестування "чорної скриньки" - це метод тестування програмного забезпечення, який досліджує функціональність програми, не заглядаючи у її внутрішні структури або роботи. Цей метод тестування може бути застосований практично до будь-якого рівня тестування програмного забезпечення: модуля, інтеграції, системи та прийняття. Іноді це називають тестуванням на основі специфікації.

CRM-системи вимагають великих обсягів даних - їхні самі функції обертаються навколо даних клієнтів, що стає першим та центром роботи CRM. Метою тестування є забезпечити чистоту даних CRM без помилок та дублікатів, а також без втрат даних. Також важливо перевірити, чи кожен користувач бачить дані лише в дозволених зонах, щоб захистити конфіденційність.

Цілісність даних відповідає якості даних у базах даних і рівню, на якому користувачі перевіряють якість, цілісність та надійність даних. Тестування цілісності даних підтверджує, що дані в базі даних є точними та функціонують, як очікувалося, у даній програмі.

Тестування цілісності даних передбачає:

- Перевірка того, чи можна порожнє значення або значення за замовчуванням отримати з бази даних.

- Перевірка кожного значення, якщо воно успішно збережене в базі даних.
- Забезпечення сумісності даних зі старим обладнанням або старими версіями операційних систем.
- Перевірка даних у таблицях даних може бути змінена та видалена
- Запуск тестів даних для всіх файлів даних, включаючи картинки, навчальні посібники, шаблони тощо.

Тестування білого ящика - є методом тестування програмного забезпечення, яке перевіряє внутрішні структури, на відміну від функціонального тестування (тобто чорний ящик тестування ). При тестуванні білих скриньок для розробки тестових кейсів використовується внутрішня перспектива системи, а також навички програмування. Тестер вибирає входи для здійснення шляхів через код і визначає очікувані результати. Це аналогічно тестуванню вузлів у ланцюзі, наприклад тестування в ланцюзі (ІКТ). Тестування білої скриньки може бути застосовано на пристрої, інтеграції та системних рівнях процесу тестування програмного забезпечення. Тестер може тестувати шляхи всередині одиниці, шляхи між одиницями під час інтеграції та між підсистемами під час тесту на рівні системи. Хоча цей метод тестового проектування може виявити багато помилок або проблем, він може втратити нереалізовані частини специфікації або відсутні вимоги. Там, де тестування білих скриньок визначається дизайном, тобто керується виключно узгодженими специфікаціями того, як повинен поводитися кожен компонент програмного забезпечення (як у DO-178C та ISO 26262 процеси), тоді методи білої скриньки можуть здійснити оцінку невиконаних або відсутніх вимог.

Методи проектування тестів білої скриньки включають такі критерії охоплення коду: тестування контрольного потоку, тестування потоку даних, галузеве тестування, покриття заяви, висвітлення рішення, змінений стан / охоплення рішення, тестування основного шляху, тестування шляху.

В результаті тестування було проведено усіма вище наведеними методами, в ході проведення тестувань не було виявлено помилок чи функціональних недоліків. Система працює стабільно – без помилок.

## **4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ**

### **4.1 Охорона праці**

При написанні кваліфікаційної роботи та розробці CRM-системи для Owl Photobook використовувався персональний комп'ютер, тому слід зазначити, що при роботі за комп'ютерною технікою необхідно дотримуватися вимог охорони праці з ціллю збереження здоров'я.

При роботі з комп'ютером значним чином відбувається вплив на нервово-емоційний стан операторів, така робота характеризується великим навантаженням на м'язи рук при роботі з клавіатурою комп'ютера, високою інтенсивністю зорової роботи та значним розумовим перенапруженням.

Отже, раціональне планування робочого місця повинно забезпечити: зменшення втоми працівників та підвищення продуктивності праці, уникнення загального дискомфорту, якнайкраще розташування інструментів та предметів праці.

Для того, щоб виявити та проаналізувати шкідливі і небезпечні виробничі фактори необхідно почати з аналізу дотримання вимог, встановлених санітарними правилами і нормами [ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин»] для виробничих приміщень та робочих місць.

На виробництві для дійсної оцінки умов праці відбувається сертифікація робочих місць відповідно до умов праці та використовується «Гігієнічна класифікація праці» за показниками небезпечності та шкідливості факторів виробничого середовища, напруженості та тяжкості трудового процесу. На основі принципів гігієнічної класифікації умови праці поділяють на чотири класи:

- Перший клас - оптимальні умови праці - це умови, за яких створюються передумови для підтримки високого рівня працездатності;



- Другий клас - допустимі умови праці, що характеризуються такими рівнями факторів виробничого середовища та трудового процесу, які не перевищують встановлених норм;

- Третій клас - шкідливі умови праці, що характеризуються такими рівнями шкідливих виробничих факторів, які перевищують нормативи і можуть мати несприятливий вплив на організм працівника і /або його нащадків;

- Четвертий клас – небезпечні/екстремальні умови праці.

На підставі сертифікації робочого місця необхідно охарактеризувати інтенсивність робіт за такими напрямками:

- відповідність обладнання нормативно-технічним вимогам;
- документації, а також характер і обсяг виконуваних робіт;
- відповідність площі та обсягу займаного робочого місця чинним нормам;

- спеціалізоване устаткування робочого місця (засоби захисту пристроїв та їх технічний стан);

- відповідність технологічного процесу, інструментів, устаткування, засобів контролю вимогам стандартів безпеки і нормам охорони праці.

Обладнання та організація робочого місця ВДТ ЕОМ повинні задовольняти відповідність конструкції всіх елементів робочого місця та їх відносного розташування ергономічним вимогам з урахуванням особливостей та характеру трудової діяльності [ДСТУ 7299:2013 «Дизайн і ергономіка. Робоче місце оператора. Взаємне розташування елементів робочого місця. Загальні вимоги ергономіки»].

Рациональне планування робочого місця повинно забезпечити: найкраще розміщення інструментів та предметів праці, уникати загального дискомфорту, зменшити втому працівників та підвищити продуктивність праці.

Заходи щодо усунення ризику ураження електричним струмом зводяться до правильного розміщення обладнання та електричних кабелів. Також інші заходи для забезпечення електробезпеки збігаються із загальними заходами пожежної та електробезпеки.

В якості заходів профілактики для того, щоб забезпечити пожежну безпеку необхідно у приміщеннях використовувати приховану електромережу, ввімкнення та вимкнення живлення виконувати обладнанням за допомогою стандартних вимикачів, надійні розетки з пожежобезпечних матеріалів. Необхідно регулярно чистити внутрішні частини комп'ютерів та інше обладнання від пилу, комп'ютери розміщувати на окремих неспалюваних столах. Для запобігання іскроутворення необхідно рідше вставляти і виймати вилки з розеток. Щодо розташування робочих місць, то вони мають бути розміщені на відстані, що не є меншою 1,5 м від стіни з вікнами, та від інших стін на відстані близько 1 м, між бічними поверхнями ВДТ - 1,2 м; від задньої площини одного ВДТ до іншого екрану - 2,5 м.

Конструкція робочої поверхні користувача ВДТ повинна забезпечувати підтримку оптимального робочого положення. Сприятливе робоче положення в процесі роботи за комп'ютером забезпечується налаштуванням висоти робочого столу, підставки для ніг і стільця.

Важливою є форма спинки стільця, яка повинна повторювати форму спини працівника. Висота стільця має бути такою, щоб користувач не відчував тиску на стегна чи куприк. Бажано обладнати крісло підлокітниками. Потрібно встановити їх так, щоб не довелося тягнутися до клавіатури.

Порівняно з вікнами робоче місце доцільно розмістити таким чином, щоб на нього природне світло потрапляло з бокової частини, переважно зліва. Робочі зони потрібно розташовувати так, щоб пряме світло не потрапляло в очі. Рекомендовано розміщувати джерела світла по обидва боки екрану, паралельно напрямку погляду. Для того, щоб уникнути відблисків на екрані, клавіатурі в напрямку очей користувача, від загальних освітлювальних приладів або сонячного світла, потрібно використовувати спеціалізовані фільтри для екранів, захисні навіси, антипроблискові сітки та жалюзі на вікнах.

Екран ВДТ повинен бути розташований на оптимальній відстані від очей користувача, яка становить від 600 до 700 мм, але не ближче 600 мм, враховуючи розмір буквено-цифрових символів та знаків.

При обладнанні робочого місця ВДТ лазерним принтером параметри лазерного випромінювання повинні відповідати вимогам [ДСанПІН 3.3.2.007-98].

Дотримання всіх необхідних заходів з охорони праці забезпечує комфортні умови праці та відсутність шкоди для здоров'я, що сприяє підвищенню продуктивності праці, а також меншому виснаженню при роботі за персональним комп'ютером.

## **4.2 Безпека в надзвичайних ситуаціях**

У процесі написання кваліфікаційної роботи та розробки CRM-системи для Owl photobook при роботі за персональним комп'ютером існують такі види небезпеки:

- порушення опорно-рухового апарату через тривалі не динамічні навантаження при роботі з ПК;
- незадовільні ергономічні характеристики робочого місця внаслідок нерационального планування робочого місця, може призвести до уражень електричним струмом, механічних травм, та розладів опорно-рухового апарату;
- ризик ураження електричним струмом через недотримання правил електробезпеки або несправність електроприладів;
- негативний вплив недостатнього освітлення робочої зони на зір і продуктивність праці користувача ЕОМ, внаслідок несправності освітлювальних приладів або неправильної конструкції системи освітлення;
- негативний вплив незадовільних параметрів повітряного середовища робочої зони на здоров'я працівника, в результаті неправильної конструкції вентиляційної системи або її несправності;
- нервово-психічні перевантаження через постійні контакти з колегами по роботі, клієнтами, керівництвом при вирішенні робочих питань, які можуть

носити конфліктний характер і призвести до емоційної нестабільності та дискомфорту, внутрішнього подразнення, та захворювань нервової системи;

- негативний вплив підвищеного рівня шуму на психо-емоційний стан працівника, що пов'язано із використанням застарілого периферійного обладнання, освітлювальних приладів, копіювального обладнання, кондиціонерів;

- ризик пожежі внаслідок несправності електрообладнання, недотримання або порушення правил пожежної безпеки обслуговуючим персоналом, що може призвести до пожежі та неправильні дії персоналу в надзвичайних ситуаціях.

Відповідно до «Правил улаштування електроустановок» реалізовані такі групи заходів електробезпеки: конструктивні заходи забезпечують захист від випадкового контакту з струмопровідними частинами за допомогою їх ізоляції та захисних оболонок. Оскільки згідно з [НПАОП 40.1-1.32-01 "Правила встановлення електроустановок. Електричне обладнання спеціальних установок"] офісні приміщення в основному відносяться до класу пожежонебезпечної зони П-Па (приміщення, в яких розташовані тверді горючі речовини), тому ступінь забезпечується захист ізоляції обладнання IP44.

Приміщення, в яких працюють розробники програмного забезпечення, класифікуються як приміщення без підвищеного ризику ураження електричним струмом. Обладнання, яке використовується в цих кімнатах, є споживачем електроенергії, що живиться від мережі змінного струму 220 В від мережі із заземленою нейтраллю, і належить до електричних установок до 1000 В закритої конструкції.

Іншим важливим завданням безпеки в надзвичайних ситуаціях є створення заходів пожежної безпеки. Закон України «Про пожежну безпеку» визначає загальні правові, соціальні і економічні основи забезпечення пожежної безпеки на території України, регулює відносини між державними органами, юридичними та фізичними особами у цій галузі, незалежно від виду їх діяльності та форм права власності.

Пожежна безпека - це стан об'єкта, при якому можливість виникнення та розвитку пожежі та вплив її небезпечних факторів на людей виключається з регульованою ймовірністю, а також забезпечується захист матеріальних цінностей. Для забезпечення пожежної безпеки в закладах проводиться протипожежна профілактика, що включає комплекс організаційно-технічних заходів, спрямованих на забезпечення безпеки людей, запобігання пожежі, обмеження її поширення, а також створення умов для успішного подолання пожежі.

Для ліквідації пожежі на початковій стадії її розвитку персонал об'єктів повинен використовувати первинне обладнання для пожежогасіння. До них відносяться:

- вогнегасники;
- протипожежне обладнання (покривала з негорючих теплоізоляційних тканин, пожежні відра, ящики з піском, лопати, ломы, сокири і тп);
- автоматичні системи пожежогасіння.

Первинне обладнання пожежогасіння, може розташовуватися як окремо, так і у складі протипожежних щитів. Це залежить від агрегатного стану та особливостей горіння різних горючих речовин та матеріалів пожежі [ДБН В. 1.1.7-2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги»] поділяються на відповідні класи. Офісні приміщення містять дерев'яні меблі, електронне обладнання і паперові матеріали. Клас пожежі в офісному приміщенні [ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва»] - пожежі твердих речовин, переважно органічного походження, горіння яких супроводжується тлінням (деревина, пластик, папір) і визначається як клас А.

Категорія приміщень [ДСТУ Б В.1.1-36:2016 «Норми визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою»] - визначається як категорія D. Визначення типу та розрахунок кількості первинних засобів пожежогасіння обладнання [ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги»]. Крім того, адміністративні приміщення повинні бути оснащені протипожежними датчиками,

які реагують на підвищення температури, на дим та полум'я. До прикладу, такі моделі датчиків, як ДТЛ, ІТМ.

Для запобігання пожежі надзвичайно важливо правильно оцінити пожежонебезпеку будівлі, виявити небезпеку та обґрунтувати методи та засоби протипожежної охорони та захисту.

Однією з умов забезпечення пожежної безпеки є усунення можливих джерел займання.

Несправності електропроводки, електричних розеток та вимикачів можуть стати причиною запалення у лабораторії. Необхідно регулярно здійснювати плановий огляд – виявляти та усувати існуючі несправності, щоб уникнути виникнення пожежі, з цих причин.

Іншою причиною запалення у лабораторних приміщеннях може бути несправність електроприладів. Щоб усунути виникнення пожежі, із цих причин, необхідно своєчасно проводити ремонт електроприладів та не використовувати несправні електроприлади.

При обігріванні приміщень електронагрівальними приладами з відкритими нагрівальними елементами може призвести до запалення у лабораторії. З огляду на те, що в приміщенні є папір, який є легкозаймистим матеріалом - відкриті нагрівальні поверхні можуть спричинити пожежу. В лабораторії рекомендується не використовувати відкриті нагрівальні прилади задля запобігання пожежі.

Коротке замикання в електропроводці може призвести до пожежі. Електропроводка повинна бути прихованою, щоб зменшити ймовірність пожежі внаслідок короткого замикання.

Ще однією з причин запалення є влучення блискавки у будівлю. Влітку під час грози може потрапити блискавка в будинок, що призведе до можливої пожежі. Щоб уникнути цього, рекомендується встановити громовідвід на даху будинку.

Недотримання заходів пожежної безпеки та куріння в приміщенні також можуть спричинити пожежу. Для ліквідації пожежі в результаті паління в лабораторії необхідно категорично заборонити куріння і дозволяти це лише в суворо відведеному місці.

Перед початком роботи, з метою запобігання пожежі, необхідно провести інструктаж з протипожежних заходів з працівниками, що працюють у лабораторії, де необхідно ознайомити працівників з правилами пожежної безпеки, а також навчити їх користуватися первинним обладнанням для гасіння пожежі.

Перше, що необхідно зробити під час виникнення пожежі це відключити електропостачання, зателефонувати до пожежної охорони, евакуювати людей з приміщень відповідно до плану евакуації, та розпочати гасіння полум'я вогнегасниками. Якщо є незначне джерело полум'я, тоді потрібно, використавши підручні засоби, зупинити доступ повітря до джерела займання.

Під час роботи комп'ютерів забороняється ремонтувати їх на робочому місці, не допускається працювати на пошкодженому обладнанні або захарашувати робочі місця матеріалами, які не використовуються для поточних робіт.

Ремонт, будь-яке технічне обслуговування та налагодження комп'ютера, а також інші операції з цього приводу слід проводити лише тоді, коли живлення повністю вимкнено. У тих випадках, коли ремонт та інші процедури неможливо виконати при відключенні електроживлення, необхідно, щоб обладнання та периферійні пристрої були заземлені, ремонт виконували двоє або більше робітників з використанням інструментів з ізольованими ручками, а також килимки-діелектрики на підлогу.

Режим праці та відпочинку працівників електронно-обчислювальної техніки визначається [ДСанНіП 3.3.2-007-98]. Під час робочого дня кожні 40-50 хвилин роботи потрібно робити перерви для відпочинку, які тривають близько 3-5-хвилин. Загальна тривалість роботи на день не повинна перевищувати 4 години, а на тиждень - 20 годин.

Таким чином, при дотриманні правил безпеки в надзвичайних ситуаціях, можна запобігти виникненню критичних результатів.

Програмний продукт CRM-система для Owl Photobook, що описаний у кваліфікаційній роботі розроблений з урахуванням вимог техніки безпеки та пожежної безпеки.

## ВИСНОВКИ

В результаті виконання кваліфікаційної роботи була розроблена CRM-система для Owl Photobook. Система розроблена згідно усіх технічних вимог. В ході роботи, був розроблений функціонал під вимоги замовника, було проведено тестування, в результаті якого помилок не виявлено, система справно виконує усі функції, які були заявлені.

В ході виконання даної кваліфікаційної роботи отримано наступні результати:

- Проведено аналіз переваг та недоліків використання CRM-систем;
- Проведено аналіз функціональних можливостей CRM-систем;
- Проведено огляд та аналіз існуючих рішень;
- Проведено аналіз переваг та недоліків існуючих рішень;
- Проведено аналіз потреб та проблем компанії, задля вдалого їх вирішення;
- Визначено найбільш вдалі технологічні рішення, задля вирішення проблем;
- Розроблено CRM-систему для Owl Photobook;
- Розроблено REST API для CRM-систем.

Таким чином, дана CRM-система є ефективним та функціональним програмним забезпеченням, що вирішує проблеми кінцевого користувача.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Angela Haran, 2005, School of Computing, Dublin Institute of Technology, Kevin Street, Dublin 8, Ireland, Development of a framework to retain customers through customer relationship management
2. CRM-система – что это такое? [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: [https://salesap.ru/crm\\_sistemy\\_chno\\_eto/](https://salesap.ru/crm_sistemy_chno_eto/)
3. Salesforce [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://en.wikipedia.org/wiki/Salesforce>
4. Who is Close CRM for? [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.fool.com/the-blueprint/close-review/>
5. Agile CRM Review [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.fool.com/the-blueprint/agile-crm-review/>
6. PHP [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://simple.wikipedia.org/wiki/PHP>
7. JavaScript [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.tutorialspoint.com/javascript/index.htm>
8. CSS Tutorial [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.tutorialspoint.com/css/index.htm>
9. AngularJS [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://angularjs.org/>
10. Understanding And Using REST APIs [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.smashingmagazine.com/2018/01/understanding-using-rest-api/>
11. Руководство для начинающих по HTTP и REST [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://code.tutsplus.com/ru/tutorials/a-beginners-guide-to-http-and-rest--net-16340>
12. MVC Framework Introduction [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: [https://www.tutorialspoint.com/mvc\\_framework/mvc\\_framework\\_introduction.htm](https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm)

13. OAuth [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://ru.wikipedia.org/wiki/OAuth>
14. Using OAuth 2.0 to Access Google APIs [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://developers.google.com/identity/protocols/oauth2>
15. JSON Web Tokens [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://auth0.com/docs/tokens/json-web-tokens>
16. JSON Web Tokens [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: <https://www.jsonwebtoken.io/>

ДОДАТКИ

ДОДАТОК А  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ  
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

**ТЕХНІЧНЕ ЗАВДАННЯ**

на розробку проекту

«Розробка CRM-системи на базі Lumen та Angular для Owl Photobook »

Розробники:

виконавець ст. гр. СПм-61

Макар Назарій Михайлович

---

(підпис)

керівник проекту

Пастух Олег Анатолійович

---

(підпис)

Тернопіль 2020

**ЗМІСТ**

1. ПІДСТАВИ ДО РОЗРОБКИ	3
2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	3
3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ	3
3.1 Функціональні вимоги	3
3.2 Технічні вимоги	4
3.3 Програмні вимоги	4
4 ЕТАПИ РОЗРОБКИ	4
5 СУПРОВІДНА ДОКУМЕНТАЦІЯ	5
6 ПОРЯДОК ЗДАЧІ ПРОЕКТУ	5
7 ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА В ПРОЕКТІ	6

## 1 ПІДСТАВИ ДО РОЗРОБКИ

Розробка проводиться у відповідності до графіку навчального плану підготовки бакалаврів за спеціальністю 121 «Інженерія програмного забезпечення».

Тема проекту: « Розробка CRM-системи на базі Lumen та Angular для Owl Photobook».

Термін виконання: до 01.12.2020 р.

## 2 ПРИЗНАЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

Інформаційна система призначена для управління відносинами з клієнтами.

Інформаційна система буде корисною в сферах поліграфічного бізнесу, а саме друк фотокниг.

Інформаційна система дозволить менеджети замовлення, створювати клієнтів та переглядати статистичні дані сайту компанії.

## 3 ВИМОГИ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

### 3.1 Функціональні вимоги

Система повинна володіти наступним функціоналом:

- Автентифікація користувача;
- Встановлення паролю користувача;
- Зміна паролю користувача;
- Відновлення паролю, з надсиланням листа на пошту;
- Редагування персональної інформації;
- Створення, редагування, видалення клієнтів;
- Створення, редагування, видалення типу клієнтів;
- Отримання списку клієнтів;
- Отримання списку типів клієнтів;
- Перегляд статистики сайту;
- Менеджмент замовлень;
- Створення нового замовлення;

- Редагування замовлення;
- Оновлення статусу замовлення;
- Видалення замовлення;
- Створення адреси доставки клієнтів;
- Менеджмент ролей.

### 3.2 Технічні вимоги

Основною вимогою до системи є відмовостійкість, адаптивність. База даних повинна бути нормалізованою.

### 3.3 Програмні вимоги

Використання СУБД: MySQL.

Розробка серверної частини: PHP, Lumen.

Розробка клієнтської частини: HTML, CSS, SCSS, JavaScript, Typescript, Angular.

## 4 ЕТАПИ РОЗРОБКИ

Розробка системи проводиться в наступному порядку:

- аналіз предметної області, виявлення позитивних та негативних сторін;
- вибір засобів розробки та архітектури системи;
- проектування бази даних системи;
- розробка програмного забезпечення системи;
- тестування системи на реальних даних;
- оформлення супровідної документації;
- здача проекту.

Результати виконання кожного етапу проекту погоджуються з керівником проекту.

## 5 СУПРОВІДНА ДОКУМЕНТАЦІЯ

Для інформаційної системи повинні бути розроблені наступні документи:

- пояснювальна записка до проекту;
- презентація проекту;

- рецензія на проект;
- диск з проектом.

Пояснювальна записка до проекту оформляється згідно діючих вимог до нормоконтролю проектів.

### 6 ПОРЯДОК ЗДАЧІ ПРОЕКТУ

Розроблена інформаційна системи повинна відповідати вимогами, що складаються з перерахованих у п.3.1 цього документу характеристик.

Для задачі проекту необхідно підготувати весь перелік документів зазначений у п.5 цього документу.

Приймання проекту проводиться спеціально створеною комісією в термін зазначені в п.1 цього документу.

### 7 ВІДМІТКИ ПРО ВИКОНАННЯ ЕТАПІВ ТА ЗМІНИ В ПРОЕКТІ

Назва етапу	Відмітка*
Аналіз предметної області	
Архітектура системи	
Проектування база даних	
Використання системи	
Супровідна документація	

\* відмітки про виконання етапу ставляться керівником проекту



ДОДАТОК Б  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ІВАНА ПУЛЮЯ

МАТЕРІАЛИ

VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

«ІНФОРМАЦІЙНІ МОДЕЛІ,  
СИСТЕМИ ТА  
ТЕХНОЛОГІЇ»



**9–10 грудня 2020 року**

**ТЕРНОПІЛЬ 2020**

УДК 004.4

**Макар Н.М., – ст.гр.СПм-61**

(Тернопільський національний технічний університет імені Івана Пулюя)

## **ІНСТРУМЕНТИ CRM-СИСТЕМИ**

UDC 004.4

**Makar N.M.**

## **CRM SYSTEM TOOLS**

Подорож покупця змінювалась протягом багатьох років і ведення бізнесу сьогодні стало як ніколи складним. Власники бізнесу та продавці повинні підтримувати зв'язок зі своїми клієнтами, стежити за перспективами, визначати можливості продажу та перехресного продажу, а також ініціювати програми утримання клієнтів, забезпечуючи при цьому, щоб дохід компанії продовжував зростати. Для отримання високого результату необхідно докласти неабияких зусиль, щоби чи то програмний, чи (аналоговий) продукт потрапив у поле зору цільової аудиторії, для цього компанії витрачають великі кошти, аби «прощупати» нішу та зрозуміти, що саме необхідно потенційним покупцям. Таргетологи і маркетингологи усього світу борються за увагу покупця. Для того, щоб краще зрозуміти своїх покупців/клієнтів необхідно бути ближче до них, зрозуміти, що їм подобається, чим вони захоплюються. Для того, щоб бізнес приносив неабиякі плоди необхідно створити прозору статистичну структуру, яка допоможе віднайти правильний шлях.

CRM-система – це ефективний інструмент для автоматизації бізнес процесів: взаємодії з клієнтами та менеджменту замовлень. Управління взаємовідносинами з клієнтами (CRM) часто називають процесом, стратегією або програмним забезпеченням / технологією, що дозволяє організаціям управляти відносинами зі своїми клієнтами та

постачальниками. CRM-система займається менеджментом тих напрямків, які бажає менеджети бізнес. При потребі є можливість під'єднання сторонніх сервісів, які мають відповідне API для роботи з ними.

Розглянемо декілька функцій CRM-системи, які допоможуть прозоро працювати з бізнес процесами.

Статистика – дисципліна, яка займається збиранням, організацією, аналізом, інтерпретацією та представленням даних. Статистика відображає дані про прибуток за певний період, дані про найактивніших клієнтів, про найпопулярніші торгові позиції та про рекламні компанії із Facebook ads і тд. Завдяки цим даним можна проводити якісний ретаргетинг, надавати знижки постійним клієнтам, та створювати привабливі, для ЦА, рекламні компанії у Facebook ads.

Звітність та аналітика – ця функція підсумовує ефективність продажів на одній інформаційній панелі. Можна налаштувати або створити нові типи звітів відповідно до вимог та експортувати їх у різні формати.

Менеджмент замовлень, до якого відноситься створення замовлень, які були зроблені не за допомогою сайту, зміна статусу, надсилання замовлень на затвердження, відправлення в друк, оплата. Це дозволить пришвидшити та прокласифікувати роботу. Завдяки цій функціональній можливості можна легко здійснювати управління замовленням та бачити прогрес його виконання.

Управління контактами – вся необхідна інформація, пов'язана з контактами замовника, такими як ім'я, адреса електронної пошти, номер телефону, деталі роботи, попередні замовлення тощо, повинна бути легкодоступною та модифікованою.

Управління електронною поштою – можна інтегрувати свою електронну пошту в CRM, щоб не довелося переходити між кількома вкладками, щоб надіслати електронне повідомлення.

Міністерство освіти і науки України  
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет \_\_\_\_\_ ФІС \_\_\_\_\_  
(повна назва факультету)  
 Кафедра \_\_\_\_\_ Програмної інженерії \_\_\_\_\_  
(повна назва кафедри)

ЗАТВЕРДЖУЮ  
 Завідувач кафедри

\_\_\_\_\_  
(підпис)      \_\_\_\_\_  
(прізвище та ініціали)  
 «    »      20\_\_ р.

**ЗАВДАННЯ  
 НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня \_\_\_\_\_ магістр \_\_\_\_\_  
(назва освітнього ступеня)

за спеціальністю \_\_\_\_\_ 121 Інженерія програмного забезпечення \_\_\_\_\_  
(шифр і назва спеціальності)

студенту \_\_\_\_\_ Макар Назарію Михайловичу \_\_\_\_\_  
(прізвище, ім'я, по батькові)

1. Тема роботи \_\_\_\_\_ Розробка CRM – системи на базі Lumen та Angular для Owl Photobook \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

Керівник роботи \_\_\_\_\_ д.т.н., проф. Пастух Олег Анатолійович \_\_\_\_\_  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «\_\_» \_\_\_\_\_ 20\_\_ року № \_\_\_\_\_

2. Термін подання студентом завершеної роботи \_\_\_\_\_

3. Вихідні дані до роботи \_\_\_\_\_  
 \_\_\_\_\_

4. Зміст роботи (перелік питань, які потрібно розробити)

Аналіз існуючих рішень, теоретичні відомості, архітектурні рішення, використані технології

Охорона праці, безпека в надзвичайних ситуаціях  
 \_\_\_\_\_  
 \_\_\_\_\_

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_



Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

Кафедра \_\_\_\_\_

**В І Д Г У К**

**на кваліфікаційну роботу на здобуття освітнього ступеня «магістр»**

Студента \_\_\_\_\_

На тему: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Спеціальність і група \_\_\_\_\_

Обсяг кваліфікаційної роботи: графічного (ілюстративного) матеріалу – \_\_\_\_ аркушів формату А1 (слайдів); кількість сторінок роботи (без додатків) – \_\_\_\_ аркушів формату А4; кількість сторінок додатків – \_\_\_\_ аркушів формату А4.

Висновок про відповідність кваліфікаційної роботи завданню \_\_\_\_\_

Актуальність теми \_\_\_\_\_

\_\_\_\_\_

Наукова новизна, теоретичне та практичне значення роботи \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Зауваження, побажання \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Позитивні сторони роботи \_\_\_\_\_

\_\_\_\_\_

Оцінка текстової та графічної частин (ілюстративного матеріалу) роботи: \_\_\_\_\_

\_\_\_\_\_

Загальна оцінка кваліфікаційної роботи: \_\_\_\_\_

Відгук склав: \_\_\_\_\_

(науковий ступінь, посада, прізвище та ініціали керівника)

« \_\_\_\_ » \_\_\_\_\_ 20\_\_ р. \_\_\_\_\_

(підпис керівника)

Міністерство освіти і науки України  
Тернопільський національний технічний університет імені Івана Пулюя

**РЕЦЕНЗІЯ**

**на кваліфікаційну роботу на здобуття освітнього ступеня «магістр»**

Студента \_\_\_\_\_

На тему: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Спеціальність і група \_\_\_\_\_

Обсяг кваліфікаційної роботи: графічного (ілюстративного) матеріалу – \_\_\_\_ аркушів формату А1 (слайдів); кількість сторінок роботи (без додатків) – \_\_\_\_\_ аркушів формату А4; кількість сторінок додатків – \_\_\_\_ аркушів формату А4.

Короткий зміст кваліфікаційної роботи та отриманих результатів \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Висновок про відповідність кваліфікаційної роботи завданню \_\_\_\_\_

\_\_\_\_\_

Актуальність теми \_\_\_\_\_

\_\_\_\_\_

Наукова новизна, теоретичне та практичне значення роботи \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Короткий критичний огляд змісту кожного розділу роботи з виділенням найбільш важливих і значущих моментів, ступінь використання здобувачем прогресивних методів досліджень, повнота викладення матеріалу, оцінка отриманих результатів: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

