

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ  
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
(повна назва випускової кафедри)

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**на тему:**

**CRM-СИСТЕМА ДЛЯ КЕРУВАННЯ ВЗАЄМОДІЄЮ З КЛІЄНТАМИ  
МАЛОГО БІЗНЕСУ**

Виконав: студент групи 1КІ-23

спеціальності

123 – „Комп'ютерна інженерія”

Шимко Олександр Григорович

Науковий керівник:

к.т.н. Розломій Інна Олександрівна

(науковий ступінь, вчене звання, прізвище та ініціали)

Рецензент:

к.т.н., доцент кафедри інформаційної безпеки  
Черкаського національного університету ім. Б.  
Хмельницького - Т.А. Стабецька

(науковий ступінь, вчене звання, прізвище та ініціали)

Черкаси 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЧЕРКАСЬКИЙ ДЕРЖАВНИЙ ФАХОВИЙ БІЗНЕС-КОЛЕДЖ

Кафедра комп'ютерної інженерії та інформаційних технологій  
(повна назва випускної кафедри)

Спеціальність 123 – «Комп'ютерна інженерія»  
(шифр і назва спеціальності)

Освітня програма «Комп'ютерна інженерія»  
(назва освітньої програми)

**ЗАТВЕРДЖУЮ**  
Завідувач кафедри КІ та ІТ  
\_\_\_\_\_ В. І. Хотунов  
(підпис)

«\_\_\_» \_\_\_\_\_ 2025 р.

**ЗАВДАННЯ**  
НА КВАЛІФІКАЦІЙНУ РОБОТУ БАКАЛАВРА

Шимко Олександр Григорович  
(прізвище, ім'я по батькові студента в називному відмінку)

1. Тема кваліфікаційної роботи бакалавра “CRM-система для керування взаємодією з клієнтами малого бізнесу”

Науковий керівник роботи к.н.т. Розломій Інна Олександрівна  
(науковий ступінь, вчене звання, прізвище, ім'я по батькові)

Затверджено наказом ЧДБК від «\_\_\_» \_\_\_\_\_ 2025 р. № \_\_\_\_\_

2. Строк подання студентом пояснювальної записки кваліфікаційної роботи бакалавра «05» червня 2024 р.

3. Вихідні дані пояснювальної записки кваліфікаційної роботи бакалавра описано аналіз та реалізацію CRM-системи, орієнтованої на потреби малого бізнесу, з урахуванням ролей користувачів, обробки клієнтських даних.

4. Зміст пояснювальної записки кваліфікаційної роботи бакалавра (перелік питань, які потрібно розробити) включає в себе аналіз існуючих CRM-систем та сформульовано постановку задачі. Наведено проєктування логічної, інфологічної моделей системи. Розроблено архітектуру CRM-застосунку.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) включає діаграму класів системи, діаграму прецедентів користувачів, інфологічну та даталогічну моделі бази даних.

Дата видачі завдання «01» жовтня 2024 р.

## КАЛЕНДАРНИЙ ПЛАН

№ пп	Назва етапів виконання пояснювальної записки кваліфікаційної роботи бакалавра	Терміни виконання етапів	Примітка про виконання
1	Вступ	20.01.2025р.	
2	Пошук та аналіз літературних джерел	12.02.2025р.	
3	Розділ 1 Огляд та порівняльний аналіз аналогів системи керування малим бізнесом	26.02.2025р.	
4	Розділ 2 Аналіз і проектування систем	18.03.2025р.	
5	Розділ 3 Проектування системи керування	29.04.2025р.	
6	Висновки	05.05.2025р.	
8	Оформлення пояснювальної записки кваліфікаційної роботи бакалавра (чистовий варіант)	14.05.2025р.	
9	Здача пояснювальної записки кваліфікаційної роботи бакалавра на кафедрі для рецензування (за 14 днів до захисту)	05.06.2025р.	
10	Перевірка пояснювальної записки кваліфікаційної роботи бакалавра на наявність ознак плагіату (за 10 днів до захисту)	09.06.2025р.	

Студент

\_\_\_\_\_ (підпис)

Шимко Олександр Григорович

(прізвище, ім'я по батькові студента)

Науковий керівник

\_\_\_\_\_ (підпис)

к.н.т. Розломій Інна Олександрівна

(науковий ступінь, вчене звання, прізвище, ім'я по батькові)

## АНОТАЦІЯ

У кваліфікаційній роботі досліджено застосування принципу парності для контролю та автоматичного виправлення одиначної помилки при паралельному способі передачі даних. Розроблено та представлено функціональну схему пристрою, що складається з передавального та приймального блоків. Запропоновано практичне технічне рішення, яке дозволяє забезпечити достовірність переданої інформації без використання складних обчислювальних алгоритмів. Проведено аналіз ефективності роботи пристрою та визначено можливості його використання в цифрових системах управління та передачі даних.

## **ANNOTATION**

This bachelor's qualification paper explores the use of the parity principle for error detection and automatic correction of single-bit errors during parallel data transmission. A functional schematic of the device, consisting of a transmitter and a receiver unit, has been developed and presented. A practical technical solution is proposed that ensures the reliability of transmitted information without the use of complex computational algorithms. The efficiency of the device has been analyzed, and its potential applications in digital control and data transmission systems have been identified.

## Список скорочень

№	Скорочення	Розшифровка
1	555	Таймерна мікросхема
2	DAC	Digital-to-Analog Converter (цифро-аналоговий перетворювач)
3	TTL	Transistor-Transistor Logic (транзисторно-транзисторна логіка)
4	АЦП	Аналого-цифровий перетворювач
5	БІС	Велика інтегральна схема
6	ВТМ	Високий логічний рівень
7	ГІС	Гібридна інтегральна схема
8	ДВП	Двійковий передавач
9	ІС	Інтегральна схема
10	КП	Контроль і перевірка
11	МС	Мікросхема
12	ОД	Одинична помилка
13	ПП	Принцип парності
14	ППД	Паралельна передача даних
15	РЛ	Рівень логіки
16	ЦАП	Цифрово-аналоговий перетворювач

**Структура та обсяг роботи.** Кваліфікаційна робота складається зі вступу, трьох основних розділів, висновків і списку використаних джерел. Загальний обсяг роботи становить 70 сторінки основного тексту, містить 25 рисунки та 13 таблиць. У роботі представлено повний цикл проектування CRM-системи — від аналізу вимог до реалізації та тестування.

## ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ СИСТЕМИ КЕРУВАННЯ МАЛИМ БІЗНЕСОМ	10
1.1 Огляд предметної області застосування системи керування малого бізнесу	10
1.2 Поняття системи керування	13
1.3 Огляд та порівняння аналогів системи керування малого бізнесу	17
1.3.1 Система керування по управлінню бізнесом «EasyWeek»	17
1.3.2 CleverBox система керування для малого бізнесу	19
1.3.3 «Beauty Pro»	22
1.4 Постановки задачі на розробку	24
Висновки до розділу 1	27
РОЗДІЛ 2. АНАЛІЗ І ПРОЄКТУВАННЯ СИСТЕМ	28
2.1 Виявлення та аналіз вимог застосунку	28
2.2 Представлення використання системи керування різними користувачами на діаграми прецедентів	29
2.3 Розробка логічної моделі системи	31
2.3.1 Представлення архітектури застосунку за допомогою діаграми класів	31
2.3.2 Представлення архітектури застосунку за допомогою діаграми пакетів	32
2.3.3 Представлення архітектури застосунку за допомогою діаграми компонентів	34
2.3.4 Представлення архітектури застосунку за допомогою діаграми розгортання	36
2.4 Моделювання поведінки застосунку	37
2.4.1 Проєктування моделі поведінки за допомогою діаграми діяльності	37
2.4.2 Проєктування моделі поведінки за допомогою діаграми послідовності	39
2.4 Проєктування бази даних для системи керування малим бізнесом	41

	3
2.5.1 Інфологічне проєктування	41
2.5.2 Даталогічне проєктування	44
Висновки до розділу 2	47
<b>РОЗДІЛ 3. ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ</b>	<b>49</b>
3.1 Огляд стеку технологій системи керування	49
3.2 Вибір оптимального середовища для розробки CRM-системи	51
3.3 Розробка серверної частини	52
3.4 Розробка інтерфейсу користувача	55
3.5 Тестування системи	59
Висновки до розділу 3	67
<b>ВИСНОВКИ</b>	<b>68</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ</b>	<b>70</b>
<b>ДОДАТКИ</b>	<b>73</b>

## ВСТУП

У сучасних умовах динамічного розвитку підприємництва та цифровізації ринку, малі підприємства стикаються з постійною необхідністю оптимізації своїх бізнес-процесів. Успішна діяльність у сфері малого бізнесу потребує не лише якісного продукту або послуги, але й ефективної системи управління взаємодією з клієнтами. З огляду на це, зростає інтерес до впровадження сучасних інформаційних технологій, зокрема систем CRM (Customer Relationship Management), які дозволяють підвищити рівень обслуговування клієнтів, зменшити витрати часу та ресурсів, а також забезпечити стратегічне управління бізнесом.

Ця робота присвячена аналізу, проектуванню та реалізації CRM-системи для малого бізнесу з метою покращення взаємодії з клієнтами, автоматизації облікових і маркетингових процесів, а також удосконалення системи управління даними. Дослідження охоплює широкий спектр завдань: від аналізу існуючих рішень до розробки власної архітектури програмного забезпечення для малого підприємства. Результати проєкту можуть бути використані у багатьох галузях малого бізнесу: у сфері послуг, торгівлі, консалтингу, креативних індустріях тощо.

**Актуальністю теми є CRM-системи в контексті малого бізнесу є надзвичайно актуальним інструментом, що сприяє підвищенню конкурентоспроможності, зміцненню клієнтської лояльності, забезпеченню персоналізованого підходу до обслуговування клієнтів і покращенню аналітики діяльності.** У той час, як великі корпорації мають доступ до складних ERP-систем, малий бізнес потребує легких, адаптивних і доступних рішень, що дозволяють ефективно управляти записами, обліком, продажами, комунікацією з клієнтами тощо. Саме тому розробка CRM-системи, адаптованої під специфіку малого бізнесу, є важливим і практично значущим завданням.

**Метою дослідження є розробка та впровадження CRM-системи для малого бізнесу, яка дозволить автоматизувати облік клієнтів, керування**

послугами чи продуктами, планування записів або зустрічей, а також поліпшити якість обслуговування та прийняття управлінських рішень.

Для досягнення цієї мети передбачено вирішення таких **завдань**:

- провести аналіз наявних CRM-систем, доступних для малого бізнесу;
- виявити функціональні та нефункціональні вимоги до системи;
- розробити структуру CRM-системи з урахуванням ролей користувачів і специфіки малого бізнесу;
- спроектувати архітектуру програмного забезпечення (діаграми класів, компонентів, розгортання тощо);
- розробити базу даних для обробки та зберігання клієнтської інформації;
- реалізувати інтерфейс користувача для зручного доступу до функцій системи;
- провести тестування та оцінити ефективність роботи системи в умовах малого підприємства.

**Об'єктом дослідження** є інформаційні процеси управління взаємодією з клієнтами у сфері малого бізнесу.

**Предметом дослідження** є CRM-система для малого бізнесу, що забезпечує автоматизацію процесів обліку, комунікації, аналізу клієнтської поведінки та управління внутрішніми бізнес-процесами.

**Практичне значення** полягає у створенні ефективного, адаптивного інструменту для цифрової трансформації малого бізнесу. Запропонована CRM-система дозволяє автоматизувати ключові бізнес-процеси, оптимізувати роботу з клієнтами, зменшити витрати часу на рутинні завдання та покращити прийняття управлінських рішень на основі аналітики. Її використання сприяє підвищенню якості обслуговування, утриманню клієнтів та зростанню прибутковості малого підприємства.

## РОЗДІЛ 1. ОГЛЯД ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ АНАЛОГІВ СИСТЕМИ КЕРУВАННЯ МАЛИМ БІЗНЕСОМ

### 1.1 Огляд предметної області застосування системи керування малого бізнесу

Малий бізнес є однією з ключових форм господарювання в сучасній економіці, що забезпечує гнучкість ринку, створення робочих місць та розвиток локального підприємництва. Згідно з загальноприйнятою класифікацією, до суб'єктів малого підприємництва відносяться підприємства, які мають обмежену чисельність працівників (зазвичай до 50 осіб), невисокий рівень доходу та обмежену частку ринку. В Україні відповідно до законодавства, статус малого підприємства отримують суб'єкти, які протягом останнього звітного періоду мають річний дохід до 10 мільйонів євро (в еквіваленті) та кількість працівників до 50 осіб[7].

Малі підприємства функціонують у найрізноманітніших секторах економіки: торгівлі, сфері послуг, креативних індустріях, харчовому виробництві, ІТ-секторі, освіті, медицині та ін. Особливо динамічно розвивається малий бізнес у галузях, де важливими є клієнтоорієнтованість, швидка адаптація до запитів ринку та індивідуальний підхід до споживача.

Роль малого бізнесу в національній економіці не обмежується створенням робочих місць — ці підприємства також сприяють підвищенню економічної активності населення, формуванню конкурентного середовища, стимулюванню інноваційної діяльності. Крім того, малий бізнес часто стає платформою для апробації нових управлінських підходів, технологій та маркетингових стратегій, що згодом можуть бути масштабовані на середні та великі компанії.

Обмеженість ресурсів, малі підприємства мають здатність до швидкого реагування на ринкові зміни, тісної взаємодії з клієнтами та гнучкості у

внутрішніх процесах. Саме ці фактори обумовлюють зростаючу потребу в інструментах цифрової трансформації, зокрема в системах автоматизації управління взаємовідносинами з клієнтами, які сприяють підвищенню ефективності та конкурентоспроможності малого бізнесу.

Гнучкість і адаптивність, малий бізнес стикається з рядом типових управлінських проблем, які істотно впливають на його ефективність і стійкість. Ці проблеми, як правило, є наслідком обмеженості фінансових, кадрових та технічних ресурсів, а також недостатньо формалізованих внутрішніх процесів.

Однією з найпоширеніших проблем є неструктуроване управління взаємодією з клієнтами. У багатьох випадках інформація про клієнтів зберігається в різних форматах — від блокнотів до електронних таблиць — що ускладнює її обробку, аналіз і використання для прийняття рішень. Відсутність централізованої бази клієнтів ускладнює персоналізацію послуг, створення програм лояльності, контроль історії відвідувань та замовлень.

Ще одним викликом є відсутність системного обліку записів. У багатьох сферах малого бізнесу, особливо у сфері послуг, проблема неконтрольованих або втрачених записів призводить до пропущених зустрічей, конфліктів у графіках та незадоволеності клієнтів. Це знижує довіру до бренду і формує негативний клієнтський досвід[3].

Також поширеною є складність у фінансовому обліку та контролі витрат. Без належного програмного забезпечення фінансовий менеджмент часто зводиться до ручного ведення записів, що збільшує ймовірність помилок, унеможлиблює оперативне бачення прибутковості послуг і ускладнює аналіз витратної частини бізнесу.

Управління послугами та ресурсами також викликає труднощі. У багатьох випадках не ведеться облік використаних матеріалів або тривалість виконання певних процедур не враховується, що унеможлиблює ефективне планування навантаження на персонал та ресурси. Також відсутність аналітики по послугах не дозволяє чітко визначити найбільш прибуткові напрямки діяльності.

Ще однією важливою проблемою є неоптимальне управління розкладом — як для працівників, так і для клієнтів. У відсутності автоматизованої системи формування графіків виникають накладки, перевантаження або, навпаки, простої, що негативно впливає на якість обслуговування та рентабельність бізнесу[10].

У сукупності ці проблеми знижують ефективність бізнес-процесів, ускладнюють прийняття управлінських рішень, зменшують клієнтську задоволеність і перешкоджають розвитку підприємства. Саме тому постає об'єктивна потреба у впровадженні інформаційної системи, здатної автоматизувати та оптимізувати ключові процеси управління у малому бізнесі.

В умовах цифровізації економіки та підвищеної конкуренції малий бізнес активно впроваджує інформаційні технології для оптимізації операційної діяльності, комунікації з клієнтами та контролю за фінансами. Обмежені ресурси, саме цифрові інструменти дозволяють малим підприємствам підвищити ефективність, знизити витрати та забезпечити стабільний розвиток.

Серед найбільш поширених рішень варто виділити CRM-системи (Customer Relationship Management), які є ключовим інструментом для керування відносинами з клієнтами. Вони дозволяють створювати клієнтські бази, фіксувати історію взаємодій, управляти записами, відправляти нагадування та проводити аналітику продажів. Для малого бізнесу CRM-системи є засобом підвищення лояльності клієнтів, персоналізації послуг і автоматизації рутинних процесів[7].

Іншим важливим інструментом є ERP-системи (Enterprise Resource Planning), які забезпечують інтегроване управління основними бізнес-процесами — фінансами, постачанням, складом, персоналом тощо. Хоча повноцінні ERP-рішення частіше застосовуються на середніх та великих підприємствах, існують адаптовані до потреб малого бізнесу легкі версії, які охоплюють лише базові функції — наприклад, облік товарів, витрат і надходжень.

Також важливу роль відіграють POS-системи (Point of Sale), які поєднують функції касового обліку, управління продажами, контролю залишків на складі та формування чеків. Такі системи широко застосовуються у роздрібній торгівлі, сфері харчування та послуг. POS-рішення часто інтегруються з CRM та фінансовими модулями, що дозволяє створювати єдину інформаційну екосистему[12].

У практиці малого бізнесу активно використовуються системи електронного документообігу, хмарні сервіси для обміну файлами (Google Workspace, Dropbox), облікові платформи (М.Е.Дос, «Сота», QuickBooks) та інструменти маркетингової автоматизації (email-розсилки, push-нотифікації, рекламні кабінети в соцмережах). Впровадження таких рішень сприяє формуванню структурованої інформаційної інфраструктури, підвищенню прозорості внутрішніх процесів та кращому контролю за ресурсами.

Отже, цифрові рішення у малому бізнесі виконують роль не лише технічної підтримки, а й інструменту стратегічного управління, що дозволяє підприємству реагувати на зміни ринку, вдосконалювати бізнес-моделі та забезпечувати стабільне зростання.

## **1.2 Поняття системи керування**

У сучасній практиці ведення бізнесу поняття «система керування» трактується як сукупність інструментів, технологій і програмних рішень, призначених для планування, організації, моніторингу та оптимізації ключових процесів діяльності підприємства. У контексті малого бізнесу система керування передусім асоціюється з практичними цифровими інструментами, що дозволяють автоматизувати облік, взаємодію з клієнтами, управління персоналом, товарами та фінансами[3].

Система керування для малого бізнесу повинна бути гнучкою, доступною, легкою в освоєнні та адаптованою до обмежених ресурсів підприємства. Основною її метою є забезпечення злагодженої роботи всіх елементів

бізнес-процесів при мінімальному залученні адміністративного ресурсу. Це дає змогу власнику або керівнику сфокусуватися на стратегічному розвитку бізнесу, замість витратити час на рутинні операції.

— У практиці малого підприємництва найбільш поширеними є такі типи систем керування: CRM-системи (Customer Relationship Management) — спеціалізовані програмні продукти для управління взаємодією з клієнтами. Вони дозволяють вести клієнтську базу, керувати записами, контролювати продажі, автоматизувати нагадування, створювати звіти та формувати маркетингові кампанії. CRM-системи є одними з найбільш адаптованих до потреб малого бізнесу завдяки своїй модульності та відносно невисокій вартості впровадження. ERP-системи (Enterprise Resource Planning) — комплексні системи управління ресурсами підприємства. Вони охоплюють більш широкий спектр функцій — від бухгалтерії та складського обліку до логістики й виробничого планування. ERP-системи частіше застосовуються у середньому бізнесі, однак спрощені версії (так звані mini-ERP) можуть ефективно функціонувати й у малих компаніях. SRM-системи (Supplier Relationship Management) — системи управління відносинами з постачальниками. Хоча вони менш поширені у малому бізнесі, у підприємствах, що займаються торгівлею або виробництвом, SRM-рішення допомагають оптимізувати ланцюги постачання, контролювати залишки та управляти закупівлями. POS-системи (Point of Sale) — касові та облікові рішення, що інтегрують управління продажами, облік товарів, друк чеків, облік знижок тощо. Часто використовуються в поєднанні з CRM-модулями [7].

Усі зазначені типи систем можуть бути частково інтегровані або існувати як окремі рішення, залежно від потреб і масштабу діяльності підприємства. Правильно підібрана система керування сприяє підвищенню ефективності бізнесу, прозорості процесів і задоволеності клієнтів, що є критично важливим для малого бізнесу в умовах конкурентного ринку.

CRM-системи (Customer Relationship Management) відіграють важливу роль в організації ефективної взаємодії між підприємством та його клієнтами.

Незалежно від специфіки бізнесу, більшість сучасних CRM-рішень містять базовий набір функціональних компонентів, спрямованих на автоматизацію ключових аспектів клієнтського сервісу. Центральним елементом будь-якої CRM-системи є клієнтська база, що забезпечує збереження й обробку детальної інформації про клієнтів: контактні дані, історію звернень, уподобання, статус замовлень, рівень активності тощо, що дозволяє персоналізувати обслуговування та вибудовувати довготривалі відносини. Важливою функцією є облік записів та взаємодій — фіксація зустрічей, дзвінків, консультацій, продажів, що забезпечує послідовність у роботі з кожним клієнтом. Планування здійснюється за допомогою вбудованих календарів, що дають змогу ефективно управляти графіком без зовнішніх інструментів. Нагадування та автоматичні сповіщення інформують про майбутні події, завдання або потребу зворотного зв'язку, знижуючи ризик помилок та пропусків. У більшості систем реалізовано маркетингові інструменти, зокрема email-розсилки та SMS-інформування, які дозволяють повідомляти клієнтів про новинки, акції та важливі зміни, часто із можливістю сегментації аудиторії. Нарешті, аналітика та звітність надають управлінцям важливі показники ефективності — кількість клієнтів, частоту візитів, середній чек, популярність послуг — що слугує основою для ухвалення обґрунтованих рішень і вдосконалення стратегії.

Узагальнено, CRM-система виконує роль комплексного інструмента для систематизації, автоматизації та аналітики клієнтської взаємодії. Саме завдяки цим компонентам підприємство отримує можливість підвищити якість сервісу, зменшити кількість помилок і втрат, а також зміцнити клієнтську лояльність.

Функціональні можливості CRM-системи визначають її здатність вирішувати прикладні завдання малого бізнесу. На практиці основні функції зосереджені навколо автоматизації ключових бізнес-процесів, що дає змогу зменшити навантаження на адміністративний персонал і забезпечити послідовність у роботі з клієнтами.

Однією з базових функцій є автоматизація записів, яка дозволяє в режимі реального часу планувати взаємодії з клієнтами — від прийому замовлень до

формування розкладу та підтвердження візитів. Це особливо актуально для бізнесів, що працюють за попереднім записом: салонів краси, майстерень, клінік, студій тощо[1].

Не менш важливою є функція ведення бази клієнтів, яка забезпечує збереження детальної інформації про кожного користувача: контактні дані, історію звернень, переваги, середній чек тощо. Це дозволяє персоналізувати підхід до обслуговування та ефективно реалізовувати програми лояльності.

Інвентаризація — ще один важливий модуль, що дозволяє контролювати залишки матеріалів, засобів чи товарів. Це забезпечує прозорість витрат, допомагає уникнути дефіциту або перевитрат, що критично для стабільної операційної діяльності.

Аналітичні інструменти CRM-систем дозволяють власникам та керівникам малого бізнесу формувати звіти щодо динаміки продажів, популярності послуг, поведінки клієнтів. Ці дані є основою для стратегічного планування, впровадження нових послуг або корекції маркетингової політики.

Ефективність будь-якої CRM-системи оцінюється не лише за функціональністю, а й за низкою якісних характеристик, які визначають зручність та надійність її використання. Однією з ключових вимог є зручний інтерфейс, який має бути інтуїтивно зрозумілим навіть для користувачів без спеціальної IT-підготовки. Простота навігації, логічна структура та адаптація до мобільних пристроїв є критично важливими для повсякденної роботи. Не менш важливою є швидкість роботи системи — вона повинна швидко реагувати на дії користувача та обробляти великі обсяги даних без затримок. Гнучкість налаштувань також відіграє суттєву роль: система повинна адаптуватися під конкретні потреби бізнесу, дозволяючи змінювати графіки, звіти, шаблони повідомлень тощо. Окрему увагу слід приділити безпеці — CRM-система має гарантувати захист персональних даних клієнтів і фінансової інформації, відповідати вимогам кібербезпеки та передбачати резервне копіювання. Крім того, важливим фактором є підтримка користувачів: наявність технічної

допомоги, інструкцій або активної спільноти суттєво підвищує зручність використання системи в довгостроковій перспективі.

З огляду на зазначене, впровадження CRM-системи з відповідним рівнем функціональності та якісними характеристиками є необхідною умовою для ефективного управління малим бізнесом у цифрову епоху.

### **1.3 Огляд та порівняння аналогів системи керування малого бізнесу**

#### **1.3.1 Система керування по управлінню бізнесом «EasyWeek»**

Платформа «EasyWeek» є одним із популярних прикладів сучасної CRM-системи, розробленої спеціально для малого та середнього бізнесу, що працює у сфері обслуговування. Основними галузями, у яких активно застосовується дана система, є б'юті-індустрія, фітнес, медичні послуги, зокрема стоматологічні клініки, а також освітні студії, автосервіси, ветеринарні клініки та інші підприємства, які надають послуги за попереднім записом[10].

Завдяки універсальній архітектурі та модульності, EasyWeek забезпечує високу адаптивність під специфіку кожного бізнес-напрямку. У сфері краси (салони, перукарні, студії нігтьового сервісу) система дозволяє вести графік майстрів, облік процедур, керувати клієнтською базою та контролювати завантаження персоналу. У фітнес-центрах та спортивних студіях EasyWeek застосовується для управління тренуваннями, записом клієнтів, персональним розкладом тренерів і передплатами на абонементи.

У медичних закладах, зокрема стоматологічних клініках, платформа дозволяє автоматизувати запис пацієнтів, вести історії візитів, синхронізувати робочий час лікарів та вести внутрішню аналітику за напрямками послуг. Таким чином, EasyWeek виконує роль не лише CRM, а й частково ERP-рішення для організації управлінських та комунікаційних процесів.

Завдяки широкому спектру функцій і зрозумілому інтерфейсу, EasyWeek став ефективним інструментом цифрової трансформації для малого бізнесу,

який прагне підвищити якість сервісу, мінімізувати кількість помилок у записах та краще організувати робочий процес.

Платформа EasyWeek (див. табл. 1.1) надає широкий набір функцій, орієнтованих на автоматизацію процесів запису, управління персоналом та підвищення рівня комунікації з клієнтами[16].

Таблиця 1.1 – Основні функції CRM-системи EasyWeek

№	Функція	Опис
1	Онлайн-запис	Надання клієнтам можливості самостійно записатися через сайт або віджет
2	Формування розкладу	Автоматизоване управління графіками працівників та вільним часом
3	SMS-нагадування	Автоматичне надсилання повідомлень клієнтам про майбутні візити
4	Управління персоналом	Призначення ролей, контроль завантаження, перегляд ефективності
5	Клієнтська база	Зберігання контактів, історії відвідувань, уподобань клієнтів
6	Аналітика	Звіти за кількістю записів, доходом, популярністю послуг
7	Інтеграції	Синхронізація з Google Calendar, Instagram, Telegram, месенджерами

*Джерело: створено автором на основі даних[7]*

EasyWeek поєднує зручність для клієнта з функціональністю для адміністратора, що дозволяє досягати високої ефективності у щоденній роботі. Наявність мобільної версії та інтеграцій з популярними сервісами робить систему зручною як для малого бізнесу, так і для приватних фахівців.

Широкий функціонал і зручність використання, система EasyWeek має певні обмеження, які можуть впливати на доцільність її впровадження в окремих типах малого бізнесу. Насамперед варто звернути увагу на обмеження безкоштовної версії, яка надає доступ лише до базового функціоналу. У ній відсутні можливості налаштування розширених шаблонів повідомлень, глибокої аналітики, індивідуального брендування чи інтеграції з певними платформами. Це створює потребу в переході на платний тариф навіть для підприємств із відносно малим навантаженням[6].

Ще одним суттєвим недоліком є складність адаптації системи до нестандартних видів діяльності. Якщо бізнес не передбачає класичної моделі «послуга – спеціаліст – клієнт», наприклад, у випадках оренди приміщень, тимчасових командних проєктів або роботи з неперіодичними подіями, система

може вимагати суттєвих компромісів або взагалі не забезпечувати повну функціональність. Внаслідок цього користувачі змушені вдаватися до зовнішніх обхідних рішень або обирати іншу платформу.



Рисунок 1.1 – Інтерфейс застосунку «EasyWeek»

На рисунку 1.1 інтерфейс системи, загальну зручність, потребує певного часу для освоєння, особливо в частині налаштування прав доступу, фільтрів та аналітичних панелей. Це може бути перешкодою для користувачів без досвіду роботи з CRM-системами[10].

Загалом, недоліки EasyWeek не перекреслюють її переваг, проте вимагають уважного аналізу на етапі вибору рішення, особливо якщо йдеться про бізнес із нестандартною організаційною структурою або обмеженим бюджетом.

### 1.3.2 CleverBox система керування для малого бізнесу

CleverBox позиціонується як універсальне рішення для автоматизації та цифрового керування процесами в різних сферах малого бізнесу. На відміну від спеціалізованих CRM-систем, орієнтованих переважно на один конкретний сегмент ринку, ця платформа розроблена з урахуванням потреб різномірних підприємств. Вона ефективно використовується у таких галузях, як роздрібна

торгівля (невеликі магазини, торгові точки, шоуруми), коворкінг-простори та офісні центри з необхідністю управління бронюванням і взаємодією з клієнтами, сервісні центри, майстерні побутових послуг та ремонтні бюро, студії навчання, консультаційні кабінети й індивідуальні заняття, а також орендні сервіси, що надають техніку, інвентар чи приміщення. Завдяки такій гнучкості платформа CleverBox адаптується до широкого спектра бізнес-потреб.

Широка сфера застосування обумовлена гнучкою структурою платформи, яка дозволяє налаштовувати бізнес-процеси залежно від характеру діяльності. Завдяки цьому CleverBox підходить як для компаній із чітким потоком клієнтів і повторюваними послугами, так і для організацій з індивідуальними, нерегулярними замовленнями.

В основі архітектури системи закладено принцип модульності, що забезпечує можливість масштабування функціоналу відповідно до зростання бізнесу. Це дозволяє малим підприємствам починати з базового набору інструментів і поступово розширювати його залежно від потреб.

CleverBox є багатофункціональним інструментом, що поєднує в собі компоненти CRM, складського та фінансового обліку. Такий підхід дозволяє малим підприємствам організувати роботу в єдиному інформаційному середовищі без потреби у використанні декількох розрізнених програм.

Таблиця 1.2 – Основні модулі системи CleverBox

№	Назва модуля	Функціональне призначення
1	CRM-модуль	Управління взаємодією з клієнтами: історія контактів, нагадування, записи
2	Облік товарів	Управління залишками, надходженнями, інвентаризація, контроль витрат матеріалів
3	Фінансовий облік	Облік доходів і витрат, звіти по прибутках, формування фінансових показників
4	Клієнтська база	Збереження контактних даних, уподобань клієнтів, історії замовлень та візитів

*Джерело: створено автором на основі даних*

Наявність цих модулів згідно таблиці 1.2 забезпечує покриття ключових аспектів діяльності малого бізнесу — від організації продажів до аналізу прибутковості. Важливо, що платформа дозволяє вести аналітику в режимі

реального часу, що значно полегшує управлінські рішення, навіть без наявності окремого фінансового чи ІТ-відділу[10].

Однією з головних переваг CleverBox є її адаптивність до різних галузей малого бізнесу, що досягається за рахунок модульної структури та широких можливостей налаштування. Система дозволяє самостійно конфігурувати набір функцій відповідно до поточних потреб підприємства, без залучення розробників або технічних спеціалістів.

Важливою перевагою є гнучке налаштування ролей користувачів, що забезпечує контроль доступу до різних функцій системи. Це особливо актуально для підприємств з кількома працівниками або філіями, де необхідно обмежити права окремих співробітників у фінансових чи адміністративних модулях.

Також платформа підтримує автоматичне формування звітності, що дозволяє в режимі реального часу отримувати дані про прибутковість, активність клієнтів, обсяг наданих послуг чи проданих товарів. Це зменшує адміністративне навантаження та дає змогу швидко реагувати на зміни у внутрішньому або зовнішньому середовищі.

Широкий функціонал, CleverBox має низку недоліків, які варто враховувати при виборі платформи для автоматизації бізнесу. Інтерфейс системи поступається за привабливістю сучасним конкурентам, зокрема у частині дизайну та взаємодії з користувачем. Для деяких власників бізнесу це може стати бар'єром у впровадженні системи, особливо якщо йдеться про працівників без технічної підготовки[12].

У базовій версії системи відсутні деякі додаткові модулі, зокрема аналітика розширеного рівня, API-інтеграції або мобільна версія для працівників. Їх активація можлива лише на платних тарифах, що може бути недоцільно для підприємств із невеликим оборотом або стабільно обмеженим бюджетом.

Отже, CleverBox (див. рис. 1.2) є гнучким і доступним інструментом для автоматизації малого бізнесу, проте потребує уважного аналізу перед

впровадженням — залежно від конкретної галузі, функціональних очікувань і фінансових можливостей підприємства.

Сьогодні	11.04.2025	Для фотографа
Студія 1	Студія 2	Кімната
10:00 - 11:00 <b>Фотосесія</b> Богдан Стрілець +380961234567 (демо)	13:00 - 12:00 <b>Ретуш</b> Олена Нужна +380987654321 (демо)	

Рисунок 1.2 – Інтерфейс «CleverBOX:CRM»

### 1.3.3 «Beauty Pro»

CRM-система «Beauty Pro» є вузькоспеціалізованим програмним рішенням, розробленим для автоматизації управлінських процесів саме у сфері індустрії краси. Її функціонал орієнтований на потреби таких суб'єктів малого бізнесу, як перукарні, косметологічні кабінети, салони краси, студії масажу, nail-майстри, бровісти та інші фахівці, які працюють за індивідуальним або змішаним графіком[17].

Основна ідея системи полягає в тому, щоб забезпечити комплексне управління всіма аспектами діяльності невеликого салону або приватного спеціаліста — від ведення записів і нагадувань клієнтам до контролю за витратними матеріалами, формування прайсів і аналітики доходів.

Важливо зазначити, що саме вузька галузева спеціалізація є сильним боком «Beauty Pro». На відміну від універсальних рішень, вона вже з перших етапів впровадження має адаптовані шаблони процедур, класифікації послуг, типові ролі персоналу (майстер, адміністратор, клієнт), а також розділи, що враховують особливості роботи з пігментами, лаками, процедурами, що тривають різний час тощо[10].

Система також враховує особливості клієнтської взаємодії у сфері краси, зокрема потребу в нагадуваннях про наступний запис, ведення карток постійного клієнта, контроль за термінами використання косметичних засобів, накопичувальними знижками або акційними пропозиціями. Це дозволяє суттєво покращити якість обслуговування та підвищити рівень лояльності клієнтів.

CRM-система «Beauty Pro» пропонує набір інструментів, які забезпечують автоматизацію ключових процесів у сфері краси. Її функціонал спеціально розроблено для щоденних потреб майстрів та адміністраторів салонів.

Головною перевагою Beauty Pro є її максимальна адаптованість до потреб індустрії краси. Усі процеси, від запису до закриття візиту, враховують специфіку цієї галузі: починаючи з тривалості послуг та часу на підготовку робочого місця — і закінчуючи фіксацією витратних матеріалів, що були використані під час процедури.

Таблиця 1.3. – Основні можливості системи Beauty Pro

№	Можливість	Опис
1	Облік послуг	Фіксація наданих процедур, вартість, тривалість, використані матеріали
2	Управління графіком	Побудова індивідуального розкладу для майстрів, бронювання часу
3	Історія клієнтів	Збереження попередніх записів, побажань, алергій, частоти відвідувань
4	Розсилки та сповіщення	Email та SMS-інформування клієнтів про акції, запис, нагадування

*Джерело: створено автором на основі даних*

Система має зручну та візуально зрозумілу аналітику (див. табл. 1.3), яка дозволяє відстежувати завантаження майстрів, дохід за окремими днями,

послугами чи клієнтами. Це забезпечує керівництву можливість оперативно приймати управлінські рішення.

Beauty Pro для фотографа					
Клієнти					
Записи					
На природі					
Услуги					
Квітень 2025		П'ятниця, 11 квітня 2025			
Квітень		Портрет	Студія	На природі	Дім
чт	пт	в			
1	2				
7	8	9	10	11	12
14	15	16	17	18	19
21	22	23	24	25	26
28	29	30			
		Анна Ткаченко 0961234567 Індивідуал 800 грн	Іван Федорчук 0509876543 Сімейна 1800 грн	Олена Возняк 0973222444 Love story 1600 грн	Юлія Щупко вул. Зор Дитяча 900 грн
		Андрій Марченко 1306678001 1200 грн	Індивідуал 1509 9422 1200 грн	Оксана Степаненко 1000 грн 1000 грн	1000 гр

Рисунок 1.3 – Інтерфейс «Beauty Pro»

Beauty Pro — це спеціалізована CRM-система, призначена для автоматизації процесів у сфері індустрії краси. Її функціонал адаптований під потреби салонів, майстрів і студій, що працюють за індивідуальним графіком (див. рис. 1.3) [17].

Також система має менше універсальних інтеграцій, порівняно з широкопрофільними CRM-рішеннями, що обмежує можливість її адаптації для комбінованих або мультиформатних бізнесів.

#### 1.4 Постановки задачі на розробку

Значний функціональний потенціал розглянутих CRM-систем («EasyWeek», «CleverBox», «Beauty Pro»), жодна з них не забезпечує повного покриття потреб малого бізнесу з урахуванням галузевої специфіки, обмежень ресурсів та необхідності індивідуального налаштування. Проаналізовані платформи виявили низку суттєвих проблем, які потребують врахування при створенні нового програмного рішення [13].

Універсальні платформи, як-от CleverBox, не завжди враховують особливості тих бізнесів, які базуються на особистому обслуговуванні клієнтів

(наприклад, приватних майстрів, спеціалістів сфери краси, фрілансерів). Вони часто перевантажені непотрібними модулями, які ускладнюють навігацію й створюють додаткове навантаження на користувача.

Натомість вузькоспеціалізовані системи, як Beauty Pro, хоча й добре адаптовані до конкретної сфери, мають обмежений спектр налаштувань і не дозволяють інтегрувати інші напрямки бізнесу або масштабувати діяльність за межі одного сегменту. Це унеможлиблює використання таких рішень у бізнесах змішаного формату або тих, що активно розвиваються.

Також важливо зазначити, що системи типу EasyWeek, хоч і зручні для швидкого старту, часто мають обмеження у безкоштовних версіях, а функціонал розширених тарифів не завжди відповідає очікуванням користувача — наприклад, відсутність кастомізації полів, обмежена аналітика, складнощі з міграцією даних чи відсутність офлайн-режиму.

Крім того, майже всі аналізовані системи мають обмежену гнучкість в інтерфейсі та архітектурі, що унеможлиблює адаптацію під нестандартні схеми роботи, наприклад, циклічне обслуговування, гібридні графіки, комісійні виплати, розділену відповідальність тощо.

Таблиця 1.4 – Порівняльна характеристика CRM-систем для малого бізнесу

№	Критерії / CRM системи	EasyWeek	CleverBox:CRM	Beauty Pro
1	Мобільний застосунок	так	так	так
2	Онлайн-запис	так	так	так
3	Онлайн-оплата	так	так	так
4	База клієнтів	так	так	так
5	Журнал візитів	так	так	так
6	Облік фінансів	так	так	так
7	Складський облік	так	так	так
8	Розрахунок зарплат	так	так	так
9	Збір відгуків	так	так	так
10	Система лояльності	так	так	так
11	Інтеграції	так	так	так
12	Сповіщення	так	так	так
13	Відкритий вихідний код	так	ні	ні
14	Кол-центр та телефонія	ні	так	так
15	Безпека	так	так	так

16	Можливість редагування даних за минулий період	так	ні	так
17	Звіти та аналітика	так	так	так
18	Шаблони проєктів	ні	ні	так

Очевидною є потреба у створенні індивідуального рішення, яке б поєднувало гнучкість налаштувань, інтуїтивний інтерфейс, доступність для малого бізнесу, а також підтримувало функціонал, орієнтований на сучасну логіку взаємодії з клієнтами — включно з обліком історії, аналітикою, інтеграцією з зовнішніми сервісами та можливістю масштабування (див. табл. 1.4) .

Жодна із систем не забезпечує баланс між функціональністю, адаптивністю й інтуїтивністю, що й обґрунтовує необхідність створення нової CRM-системи, спеціально адаптованої для малого бізнесу, з урахуванням гнучкості архітектури, зручного UX/UI, повної кастомізації та перспектив розвитку.

## Висновки до розділу 1

Проведений теоретичний і функціональний аналіз CRM-систем, таких як EasyWeek, CleverBox:CRM та Beauty Pro, виявив низку обмежень, що ускладнюють їх застосування у сфері малого бізнесу з галузевою специфікою. Широка представленість базових функцій — як-от онлайн-запис, база клієнтів, аналітика, фінансовий облік — зазначені системи не забезпечують належного рівня гнучкості, адаптивності та інтеграційної сумісності.

Універсальні платформи часто містять надлишковий функціонал, що ускладнює інтерфейс та знижує ефективність роботи. Водночас вузькопрофільні рішення мають обмежені можливості масштабування та налаштувань, що не відповідає динаміці розвитку мікробізнесу. Особливо критичним є брак інструментів для індивідуалізації сервісу, підтримки офлайн-режиму, циклічного обслуговування клієнтів, налаштування гібридних графіків та кастомізації звітності.

Виявлені недоліки підтверджують доцільність і актуальність створення нового програмного продукту, орієнтованого на потреби малого бізнесу, з урахуванням принципів модульності, мінімалістичного дизайну, багатофункціонального ядра та інтуїтивного управління. В основу розробки має бути покладена логіка зручної взаємодії з клієнтами, простоти адміністрування, відкритості до інтеграцій та можливості масштабування.

## РОЗДІЛ 2. АНАЛІЗ І ПРОЄКТУВАННЯ СИСТЕМ

### 2.1 Виявлення та аналіз вимог застосунку

Побудова ефективної CRM-системи для малого бізнесу потребує чіткого визначення вимог, що ставляться до її функціональності, надійності, гнучкості та зручності використання. На етапі аналізу було здійснено ідентифікацію функціональних та нефункціональних вимог до програмного забезпечення, які стали основою для формування архітектури системи, структури бази даних і моделі взаємодії користувачів із застосунком[3].

До функціональних вимог належать ті, що визначають, що саме повинна робити система: зберігати інформацію про клієнтів, керувати записами, нараховувати оплату, надсилати сповіщення тощо. У свою чергу, нефункціональні вимоги описують як система повинна виконувати ці функції: швидкість, безпечність, масштабованість, сумісність з мобільними платформами (див. табл. 2.1).

Таблиця 2.1 – Класифікація вимог до CRM-системи

№	Тип вимоги	Формулювання вимоги
1	Функціональна	Реєстрація та автентифікація користувачів (адміністратора, майстра, клієнта)
2	Функціональна	Створення, редагування та скасування запису на послугу
3	Функціональна	Формування звітів щодо записів, доходів, активності клієнтів
4	Функціональна	Збереження історії візитів та платежів кожного клієнта
5	Нефункціональна	Мобільна адаптивність інтерфейсу користувача
6	Нефункціональна	Захист персональних даних відповідно до стандарту GDPR
7	Нефункціональна	Висока продуктивність системи при навантаженні до 100 одночасних користувачів
8	Нефункціональна	Автоматичне резервне копіювання даних щодоби

Формування вимог також передбачає виявлення ключових ролей користувачів, кожна з яких має свій рівень доступу до функціоналу системи. Ролі були визначені з урахуванням специфіки малих підприємств, де часто власник або майстер виступає і адміністратором, і виконавцем послуг.

Таблиця 2.2 – Ролі користувачів та їхні повноваження

№	Роль	Опис ролі	Основні повноваження
1	2	3	4
1	Адміністратор	Керує налаштуваннями, створює акаунти, аналізує діяльність	Повний доступ, керування системою, модерація даних
2	Майстер	Надає послуги клієнтам, керує особистим графіком	Доступ до власного календаря, історії візитів, клієнтської бази
3	Клієнт	Отримує послуги, взаємодіє із системою для запису, оплати, зворотного зв'язку	Реєстрація, перегляд графіка, створення/скасування запису, оцінювання

На основі виявлених ролей і функцій ( див. табл. 2.2) було побудовано контекстну діаграму системи, що демонструє основні зовнішні взаємодії користувачів із CRM-застосунком.

Усі виявлені вимоги та сценарії формують основу для побудови діаграм прецедентів і логічної структури системи, що розглядатимуться у наступних підрозділах.

## 2.2 Представлення використання системи керування різними користувачами на діаграми прецедентів

Проектування системи керування взаємодією з клієнтами передбачає чітке визначення ролей користувачів і можливих сценаріїв їхньої взаємодії із програмним забезпеченням. З цією метою застосовується діаграма прецедентів (Use Case Diagram), яка дозволяє візуалізувати функціональні можливості системи, а також окреслити зв'язок між діями користувачів і відповідними сервісами.

У контексті CRM-системи для фотографа виділено дві основні ролі:

- Адміністратор — відповідає за наповнення системи, створення записів, керування послугами та клієнтами;
- Клієнт — здійснює перегляд послуг, записів, взаємодіє із календарем і самостійно бронює фотосесію.

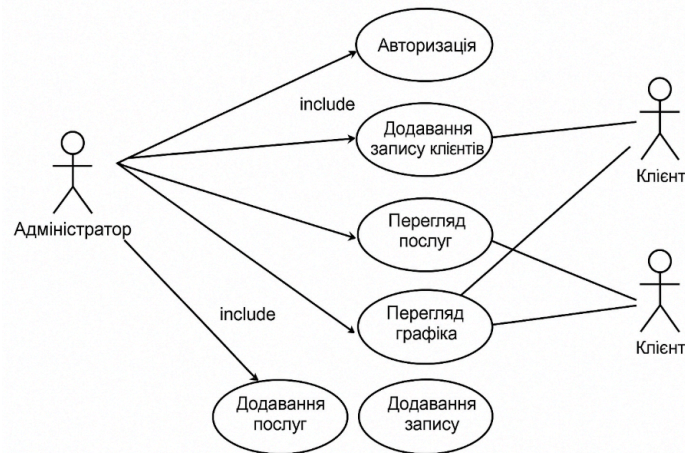


Рисунок 2.1 – Зображення прецедентів системи CRM для фотографа

На рисунку 2.1 відображено взаємодію обох типів користувачів із основними функціональними модулями системи. Кожен прецедент (use case) представлений еліпсом, що пов’язує актора (користувача) з конкретною дією в межах системи.

Основні прецеденти системи охоплюють ключові сценарії взаємодії користувачів із CRM-застосунком. Авторизація є базовим механізмом входу до системи, спільним для всіх ролей, і реалізується через зв’язок «include», що вказує на обов’язковість виконання цієї дії перед доступом до інших функцій. Перегляд послуг дозволяє користувачам ознайомитися з актуальним переліком запропонованих послуг, включаючи опис, ціну та тривалість. Перегляд графіка забезпечує доступ до розкладу та вільних слотів, з урахуванням зайнятості фотографа. Додавання запису є основним сценарієм для клієнта і передбачає вибір послуги, дати та часу з подальшим бронюванням. Функція додавання послуг реалізована для адміністратора і дозволяє редагувати чи оновлювати перелік доступних фотосесій. Також адміністратор має змогу вручну створювати записи клієнтів — наприклад, при надходженні замовлення телефоном або іншим офлайн-способом.

Кожен елемент діаграми є результатом попереднього аналізу функціональних вимог (див. п. 2.1) і слугує основою для побудови логічної архітектури.

## 2.3 Розробка логічної моделі системи

### 2.3.1 Представлення архітектури застосунку за допомогою діаграми класів

На етапі логічного проєктування програмного забезпечення важливим інструментом моделювання структури є діаграма класів (Class Diagram), що відображає внутрішню організацію системи у вигляді об'єктно-орієнтованих сутностей, їхніх атрибутів, методів і зв'язків між ними. Такий тип діаграми є основою для подальшої реалізації у вигляді програмного коду або структури бази даних.

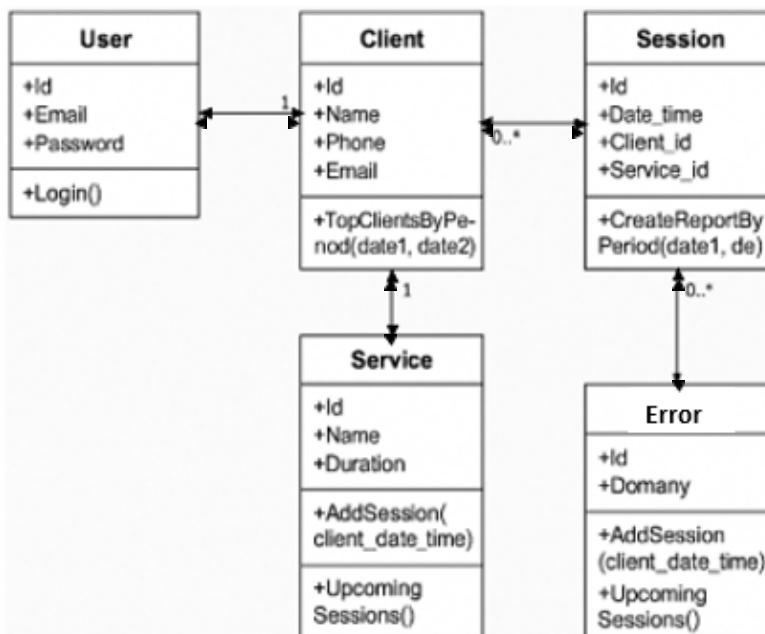


Рисунок 2.2 – Зображення класів

Опис основних класів, представлених на рисунку 2.2, моделює логіку роботи CRM-системи для фотографа та охоплює ключові бізнес-процеси: роботу з клієнтами, планування записів, управління послугами, облік помилок і формування звітності. Клас *User* представляє користувача системи — адміністратора або фотографа. Він має атрибути *Id*, *Email*, *Password* та метод *Login()*, який реалізує авторизацію в системі. Кожен користувач пов'язаний з

одним обліковим записом класу *Client*, що зберігає персональні дані клієнтів: *Id*, *Name*, *Phone*, *Email*. Додатково реалізовано метод *TopClientsByPeriod(date1, date2)*, який дозволяє сформувавши звіт про найактивніших клієнтів за заданий період. Клієнт може мати одну або кілька фотосесій (об'єктів класу *Session*) і пов'язаний з певною послугою (*Service*).

Клас *Session* описує заплановану або завершену фотосесію. Його атрибути: *Id*, *Date\_time*, *Client\_id*, *Service\_id*. Метод *CreateReportByPeriod(date1, date2)* використовується для формування звітів за період. Сесія пов'язана з клієнтом, послугою та класом *Error* — спеціальним модулем для обробки помилок. Клас *Service* описує тип фотопослуги (наприклад, портретна, студійна чи весільна) з атрибутами *Id*, *Name*, *Duration*. Методи *AddSession(client\_date\_time)* та *UpcomingSessions()* відповідають за додавання нових сесій та відображення майбутніх подій відповідно. Кожна послуга може бути замовлена одним або кількома клієнтами.

Окремо виділений клас *Error*, який відповідає за фіксацію помилок або логування збоїв у системі. Він має атрибут *Id* і *Domain*, а також дублює методи *AddSession()* та *UpcomingSessions()* для відстеження проблем у плануванні. Уся діаграма класів побудована з урахуванням принципів масштабованості, розширюваності та мінімізації надмірної складності, що робить її зручною для реалізації в реальних умовах використання CRM-системи.

### **2.3.2 Представлення архітектури застосунку за допомогою діаграми пакетів**

Для ефективною структуризації застосунку системи керування клієнтськими записами фотографа доцільним є використання архітектурного підходу на основі пакетів. Кожен пакет відповідає за окремий логічний блок функціоналу. Така організація спрощує масштабування, підтримку, тестування та забезпечує гнучкість при оновленнях і модернізації.

На рисунку 2.3 представлено діаграму пакетів CRM-застосунку, що моделює основні компоненти програмної архітектури.

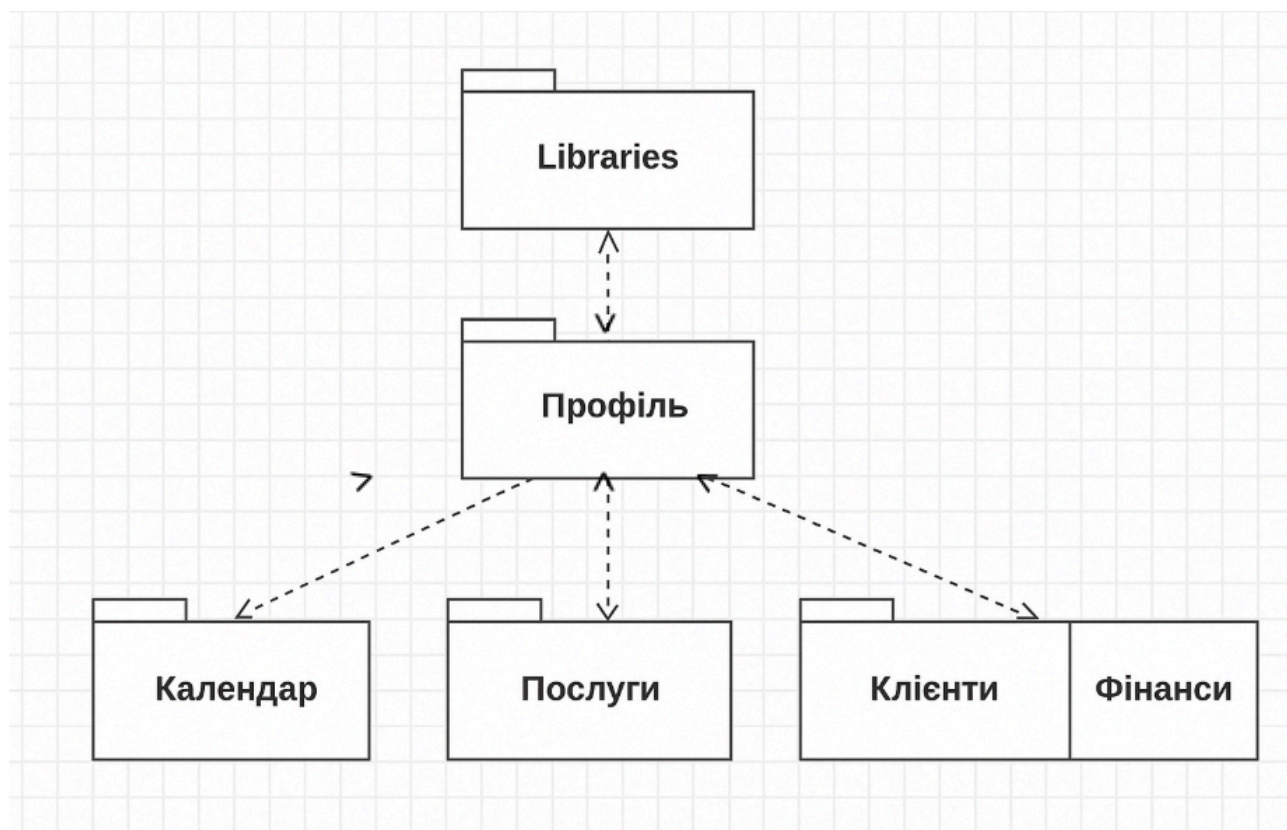


Рисунок 2.3 – Зображення пакетів CRM-системи для фотографа

Описані пакети архітектури системи відображають логічну та функціональну структуру CRM-застосунку, орієнтованого на потреби фотографа. Центральним елементом є пакет Профіль, який виступає основним модулем взаємодії з іншими складовими системи та відповідає за відображення й оновлення інформації про користувача. Саме з нього ініціюється доступ до календаря, послуг, клієнтів і фінансів. Пакет Календар реалізує функції створення, перегляду та редагування записів на фотосесії. Він отримує дані з «Профілю» і зберігає інформацію про сесії у відповідному форматі.

Пакет Послуги відповідає за управління списком фотопослуг: додавання, редагування, виведення на екран, а також зв'язування з клієнтами та записами. Клієнти — це окремий модуль, що містить базу зареєстрованих користувачів, їх контактні дані, історію замовлень і взаємодій. Зв'язок із ним здійснюється через «Профіль». Пакет Фінанси забезпечує зберігання та обробку даних про оплату послуг, формування доходів і витрат, а також побудову аналітичних звітів. Він є

розширенням функціональності модуля «Клієнти». Останнім є пакет Libraries, який підключає сторонні функціональні модулі — стилі, валідацію форм, анімації, API — і взаємодіє через «Профіль», який виконує роль координатора.

Пакетна архітектура системи побудована на принципах модульності та розподілу відповідальності, що дозволяє протестувати або вдосконалювати кожен компонент окремо. Всі пакети мають спільну точку входу — «Профіль», який координує взаємодію між модулями. Застосування односпрямованих зв'язків сприяє зменшенню зчеплення між модулями, підвищуючи гнучкість та стабільність системи. Така структура відповідає сучасним підходам у веб-розробці й забезпечує логічну та зручну взаємодію з боку користувача-фотографа.

### **2.3.3 Представлення архітектури застосунку за допомогою діаграми компонентів**

Для забезпечення цілісного уявлення про розміщення програмних компонентів у середовищі виконання доцільним є використання діаграми розгортання (Deployment Diagram). Така діаграма демонструє фізичну структуру системи, де кожен елемент — це сервер, клієнтська машина або база даних, які взаємодіють між собою через протоколи обміну даними.

У випадку CRM-системи, що розробляється для фотографа, архітектура є типовою трирівневою моделлю, що включає клієнтську сторону, вебсервер та базу даних. Діаграма нижче відображає відповідну структуру.

Компоненти, представлені на рисунку 2.4, формують класичну трирівневу модель розгортання CRM-системи для фотографа. Першим компонентом є клієнтська сторона (Клієнт), що представлена у вигляді браузера або мобільного додатку користувача. Саме вона надсилає HTTP-запити до вебсервера при виконанні дій, таких як створення запису на фотосесію, реєстрація, перегляд послуг тощо. Інтерфейс реалізований за допомогою технологій HTML, CSS та JavaScript.

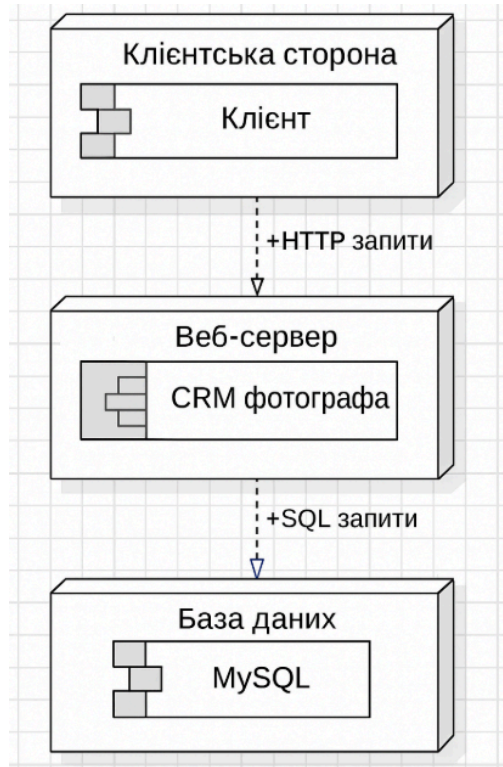


Рисунок 2.4 – Зображення розгортання CRM-системи для фотографа

Другий компонент — веб-сервер (CRM фотографа) — виконує основну логіку застосунку. Він відповідає за авторизацію, валідацію форм, обробку запитів, маршрутизацію, а також за сервіси, пов’язані з клієнтами, записами та фінансами. Веб-сервер може бути реалізований із використанням таких технологій, як Node.js, PHP або Python (Flask, Django). Для обміну даними з базою використовується SQL.

Третім компонентом є база даних (MySQL) — основне сховище інформації системи. У ній зберігаються дані про користувачів, клієнтів, фотосесії, послуги та платежі. Вона забезпечує цілісність і транзакційність операцій, відповідає на запити з вебсервера, і може бути розміщена як локально, так і в хмарному середовищі (наприклад, AWS RDS або Azure SQL).

Взаємодія між компонентами відбувається наступним чином: клієнт надсилає HTTP-запити до вебсервера, сервер обробляє їх і, за необхідності, звертається до бази даних через SQL-запити. У відповідь сервер повертає клієнту результат у вигляді веб-сторінки або JSON-об’єкта (у разі використання SPA чи мобільного інтерфейсу).

Запропонована модель розгортання забезпечує масштабованість, чітке розділення відповідальності та високу безпеку. Вона дозволяє легко змінювати окремі компоненти або середовище розміщення, не порушуючи загальної логіки роботи системи, що робить її гнучкою й надійною в експлуатації.

### **2.3.4 Представлення архітектури застосунку за допомогою діаграми розгортання**

Діаграма розгортання (Deployment Diagram) уніфікованої мови моделювання UML використовується для відображення фізичного розміщення компонентів інформаційної системи, описуючи, на яких вузлах (пристроях або серверах) вони функціонують і яким чином взаємодіють між собою. Для CRM-системи фотографа, що розробляється як веб-застосунок, обрана трирівнева архітектура із чітким розмежуванням відповідальності між клієнтською стороною, вебсервером та базою даних.

Опис до Рисунка 2.5 структурує архітектуру розгортання CRM-системи для фотографа на три ключові компоненти. Перший — клієнтська сторона, представлена браузером або мобільним додатком користувача, яка ініціює всі взаємодії із системою за допомогою HTTP-запитів. Інтерфейс дозволяє переглядати графік, записуватись на фотосесії, переглядати портфолію, профіль та фінансову інформацію.

Другим компонентом є застосунок фотографа, що виконує роль вебсервера. Він складається з трьох логічних частин: View (інтерфейс користувача), Controller (контролер, який обробляє запити) і Model (модель, що взаємодіє з базою даних). Саме сервер приймає HTTP-запити від клієнта, обробляє їх, викликає бізнес-логіку та, за потреби, звертається до бази даних через SQL-запити.

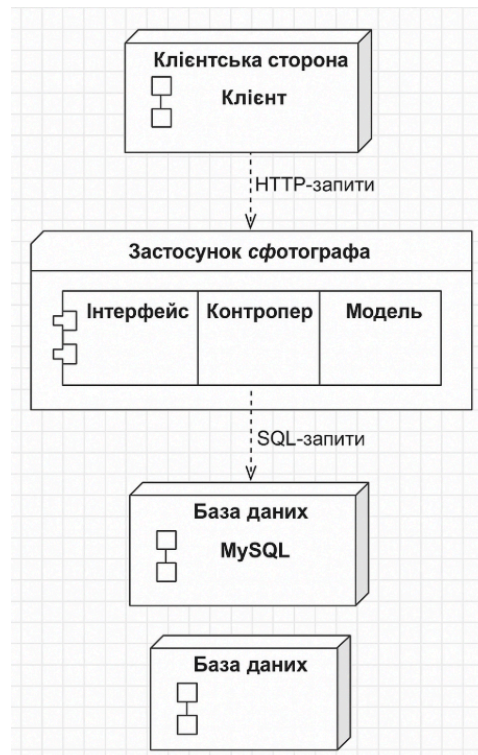


Рисунок 2.5 – Зображення розгортання CRM-системи для фотографа

Третій елемент — база даних MySQL, що слугує централізованим сховищем усіх даних системи (клієнти, послуги, записи, платежі). Вона підтримує транзакції, індексацію, забезпечує цілісність і захист даних. Взаємодія між моделлю застосунку та базою здійснюється через стандартні SQL-запити.

Узагальнена схема взаємодії виглядає так: клієнт надсилає запит → застосунок обробляє його, при потребі звертається до БД → результат повертається клієнту у вигляді вебсторінки або API-відповіді. Така модель відповідає сучасним стандартам розробки CRM-систем для малого бізнесу, забезпечує гнучкість, масштабованість і технічну адаптивність.

## 2.4 Моделювання поведінки застосунку

### 2.4.1 Проєктування моделі поведінки за допомогою діаграми діяльності

Одним із ефективних інструментів моделювання поведінки інформаційної системи є діаграма діяльності (Activity Diagram). Вона дозволяє візуально

відобразити послідовність дій користувача або системи при виконанні певного процесу, а також врахувати можливі розгалуження, цикли та умови.

У даному підпункті представлено діаграму, яка моделює типовий сценарій взаємодії користувача з CRM-системою фотографа — додавання нового клієнта та запис на фотосесію. Процес охоплює як початкову реєстрацію клієнта, так і перевірку унікальності даних та подальше бронювання послуги.

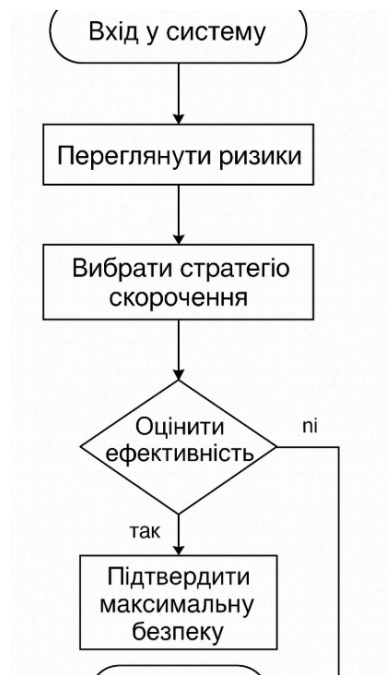


Рисунок 2.6 – Зображення діяльності процесу реєстрації клієнта і запису на фотосесію

Діаграма діяльності, зображена на рисунку 2.6, моделює послідовність кроків, які відбуваються під час реєстрації нового клієнта в системі та подальшого запису на фотосесію. Процес починається з початкової точки (чорне коло), що символізує старт взаємодії з системою. Першою дією є запит на введення контактних даних клієнта, що може виконуватись як адміністратором вручну, так і клієнтом самостійно через веб-інтерфейс.

Після цього відбувається перехід до сторінки «Клієнти», де натискання кнопки «Додати клієнта» ініціює початок процедури реєстрації. Наступним етапом є перевірка унікальності даних, яка є критично важливою для забезпечення цілісності бази клієнтів. Цей етап реалізований у вигляді умовного

блоку (ромба), що дозволяє моделювати розгалуження логіки залежно від результатів перевірки.

У разі, якщо дані клієнта вже наявні у системі, користувачеві пропонується або повторно уточнити інформацію, або завершити реєстрацію без створення дубліката. Якщо ж дані є унікальними, система переходить до етапу успішного збереження інформації, після чого відкривається сторінка «Графік».

Далі здійснюється процедура запису на фотосесію, яка передбачає вибір дати, часу та конкретної послуги. Після підтвердження вибраних параметрів клієнт отримує сповіщення про бронювання (наприклад, у вигляді SMS або email-повідомлення). Завершення всієї процедури позначається кінцевою точкою (подвійне коло).

Завдяки чіткій структурі ця діаграма діяльності забезпечує високу прозорість бізнес-процесів, дозволяє швидко виявляти можливі точки збоїв і формалізує логіку роботи системи. Вона може бути використана як технічна основа для подальшої реалізації функціоналу реєстрації та запису в CRM-застосунку для фотографа. Такий підхід сприяє покращенню зручності користувачів, мінімізації помилок введення та підвищенню загальної ефективності взаємодії з клієнтами.

#### **2.4.2 Проектування моделі поведінки за допомогою діаграми послідовності**

Діаграма послідовності (Sequence Diagram) є одним із найефективніших засобів моделювання динамічної поведінки системи. Вона дозволяє детально відобразити порядок взаємодії між об'єктами в межах певного сценарію, фіксуючи послідовність викликів, запитів і відповідей.

Для CRM-системи фотографа прикладом такого сценарію є створення нового запису на фотосесію, що ініціюється клієнтом і опрацьовується через логіку вебзастосунку та базу даних.



Рисунок 2.7 – Діаграма послідовності: додавання фотосесії в CRM-системі

Описаний сценарій взаємодії на рисунку 2.7 ілюструє послідовність дій користувача та системи під час створення нової фотосесії в CRM-системі для фотографа. Сценарій починається з того, що користувач натискає кнопку «Додати фотосесію» у своєму особистому кабінеті. У відповідь система автоматично перенаправляє його на сторінку, що містить форму створення нового запису.

На наступному етапі система підтверджує запит і відображає відповідну форму, де користувач вводить параметри майбутньої фотосесії: обирає тип послуги (наприклад, студійна, весільна), дату, час і може залишити коментарі або уточнення. Після заповнення всіх обов'язкових полів, система виконує запит на додавання сесії до бази даних.

База даних, у свою чергу, обробляє отримані дані, створює новий запис, зберігає його та надсилає підтвердження про успішне збереження. Після цього

користувач отримує повідомлення (у вигляді візуального сповіщення або поп-уп) про успішну реєстрацію фотосесії.

На завершальному етапі сценарію передбачено умовний блок (alt), у якому користувач обирає подальші дії: або створити ще один запис, або завершити взаємодію і повернутись до календаря.

Цей сценарій, представлений на діаграмі послідовності, чітко структурує логіку взаємодії між користувачем, інтерфейсом системи, сервером та базою даних. Він є не лише наочною моделлю поведінки системи, а й ключовим інструментом для формування технічного завдання, реалізації функціоналу та подальшого автоматизованого тестування.

## **2.4 Проєктування бази даних для системи керування малим бізнесом**

### **2.5.1 Інфологічне проєктування**

Інфологічна модель (Entity-Relationship Model, ER-модель) є концептуальною основою побудови бази даних та дозволяє формалізовано відобразити основні сутності, їх атрибути та зв'язки між ними у межах майбутньої CRM-системи. Зокрема, така модель потрібна для забезпечення логічної цілісності даних, оптимізації запитів та забезпечення масштабованості системи.

Опис ключових сутностей у системі, наведених на рисунку 2.8, дозволяє чітко структурувати основні інформаційні об'єкти CRM-системи для фотографа, їх атрибути та логіку взаємозв'язків.

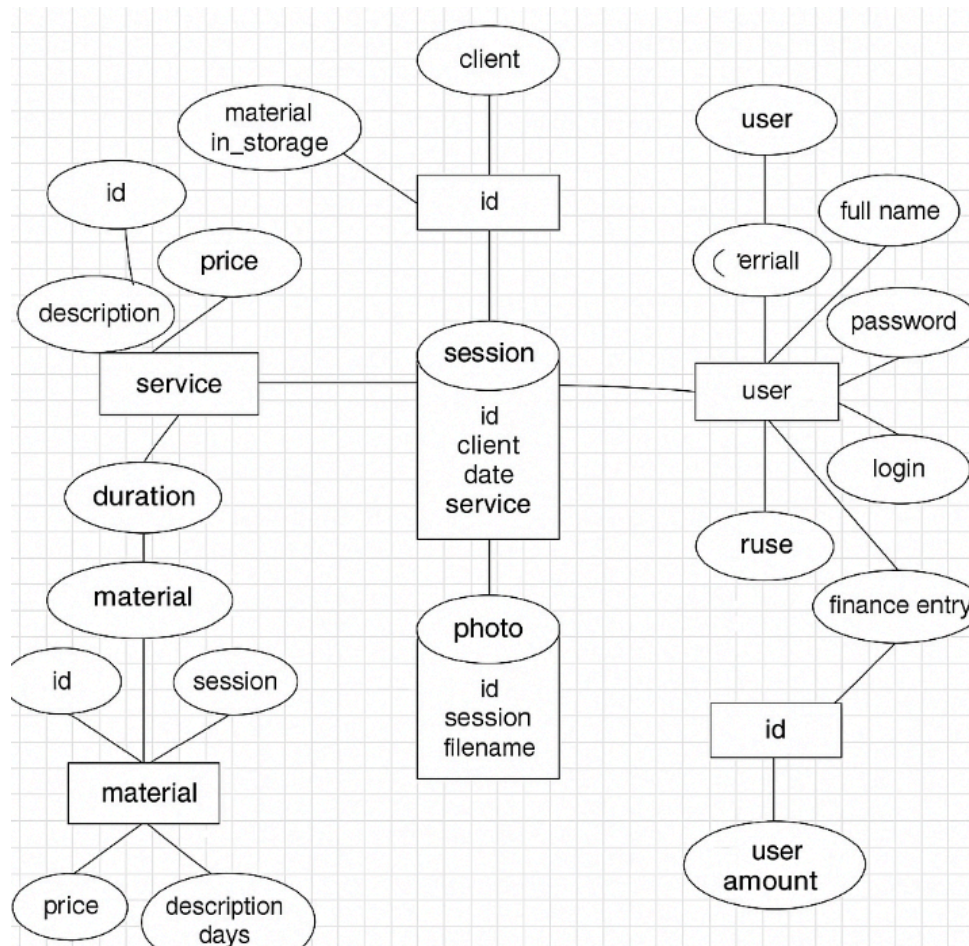


Рисунок 2.8 – Інфологічна модель CRM-системи для фотографа

Client — сутність, що зберігає персональні дані клієнтів, які замовляють фотопослуги. До основних атрибутів належать: id (ідентифікатор), full\_name (ПІБ), email, phone, social\_links (посилання на соціальні мережі), birth\_date (дата народження). Кожен клієнт може бути пов'язаний із користувачем (User), якщо він має акаунт у системі, а також із фотосесіями (Session), у яких він брав участь.

User — репрезентує обліковий запис системи. Ця сутність охоплює як адміністраторів/фотографів, так і зареєстрованих клієнтів. Основні атрибути: id, email, password, role (роль у системі), status (наприклад, активний чи заблокований). Залежно від типу користувача, він може бути асоційований із сутністю Client (для клієнтів) або Employee (для співробітників).

Session — сутність, що описує інформацію про заплановані або проведені фотосесії. Вона містить такі атрибути: id, datetime (дата і час проведення), service\_id (ідентифікатор обраної послуги), client\_id (ідентифікатор клієнта).

Кожна сесія пов'язана як з клієнтом (Client), так і з відповідною послугою (Service).

Service — каталог фотопослуг, що доступні для замовлення. Основні атрибути: id, name (назва послуги), description (короткий опис), price (вартість), duration (тривалість у хвилинах). Одна послуга може бути пов'язана з багатьма сесіями, оскільки кілька клієнтів можуть обирати один і той самий тип зйомки.

Finance — сутність, яка фіксує фінансові операції: як прибутки, так і витрати. Її атрибути включають: id, sum (сума операції), date, type (тип — прибуток чи витрати), category (категорія витрат чи доходів), user\_id (ідентифікатор користувача, який створив запис). Зв'язок із користувачем дає змогу вести облік на індивідуальному рівні.

Employee — сутність, що описує працівників студії або агентства, якщо система використовується в командному середовищі. До основних полів належать: id, full\_name, position (посада), phone, salary. Employee зазвичай пов'язаний із сутністю User, яка забезпечує йому доступ до функціоналу системи.

Така структурована модель сутностей забезпечує логічну побудову CRM-системи, її масштабованість, адаптивність до реального бізнесу та підтримку повноцінного управління клієнтами, записами, послугами й фінансами.

Таблиця 2.9 - Типи зв'язків між сутностями

№	Взаємодія	Тип зв'язку
1	Client – Session	1:N
2	Session – Service	N:1
3	User – Client	1:1
4	User – Finance	1:N
5	User – Employee	1:1 (опційно)

Запропонована інфологічна модель у табл. 2.9 відображає логічну структуру предметної області CRM-системи для фотографа. Вона забезпечує повне покриття функціоналу, включаючи реєстрацію клієнтів, запис на послуги, керування фотосесіями, фінансовий облік та роботу з персоналом. Модель стане основою для побудови даталогічної структури в обраній СКБД на наступному етапі проектування.

### 2.5.2 Даталогічне проектування

Даталогічне (логіко-фізичне) проектування полягає у створенні структури бази даних з урахуванням синтаксису, правил і можливостей конкретної системи керування базами даних (СКБД), зокрема MySQL. Така модель деталізує типи даних, зовнішні ключі, індексацію, обмеження цілісності та реалізацію зв'язків між таблицями, які описані в інфологічній моделі.

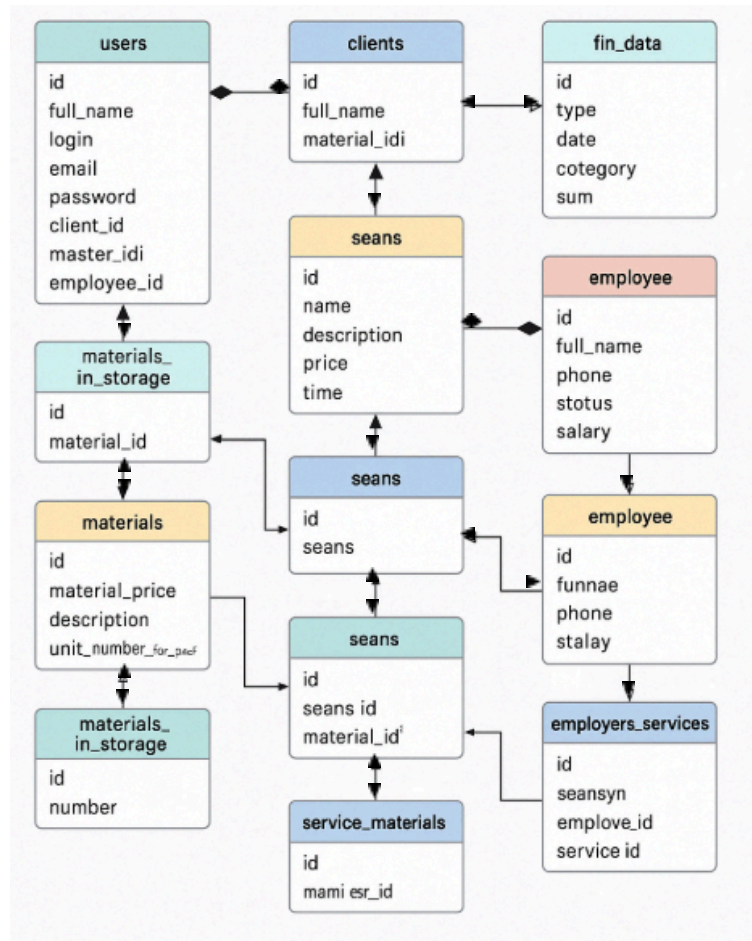


Рисунок 2.9 – Даталогічна модель бази даних CRM-системи для фотографа

Даталогічна модель бази даних CRM-системи для фотографа складається з тринадцяти таблиць, які охоплюють ключові функціональні напрями: облік користувачів, клієнтів, послуг, сесій, матеріалів, фінансів і цифрових файлів. Кожна таблиця має чітко визначене призначення, набір атрибутів і зв'язки з іншими сутностями, що дозволяє забезпечити повноцінну підтримку бізнес-процесів студії або фотографічного агентства.

Таблиця `users` відповідає за зберігання облікових записів усіх зареєстрованих користувачів. Вона включає такі поля: `id`, `full_name`, `login`, `email`, `password`, `role`, а також зовнішні ключі `client_id`, `master_id`, `employee_id` для зв'язку з відповідними сутностями. Це дозволяє враховувати специфіку взаємодії системи з різними категоріями користувачів (клієнти, фотографи, працівники).

Таблиця `clients` зберігає детальну інформацію про клієнтів: ПІБ, дату народження, телефон, email та посилання на соцмережі (`telegram`, `instagram`). Її логічний зв'язок із `users` дозволяє інтегрувати персональні дані з механізмами авторизації.

`employee` — таблиця для обліку співробітників, таких як фотографи, адміністратори, менеджери. Поля включають `id`, `full_name`, `phone`, `status` і `salary`. Ця таблиця інтегрується з розширеним функціоналом через зв'язки з послугами та фінансовими даними.

Таблиця `fin_data` відповідає за облік фінансових транзакцій. Атрибути `type`, `date`, `category` і `sum` дозволяють класифікувати витрати й прибутки, створювати звіти, формувати динаміку грошових потоків.

`services` — таблиця, яка містить повний перелік послуг: `id`, `name`, `description`, `price`, `time`. Вона має зв'язки як із сесіями, так і з матеріалами, дозволяючи будувати складні бізнес-моделі з варіативними затратами.

Таблиця `seans` моделює фотосесії — вона включає `id`, `client_id`, `date`. Для реалізації зв'язку «багато до багатьох» між сесіями й послугами створено допоміжну таблицю `seans_services`, що дозволяє прив'язувати декілька послуг до однієї події.

Для підтримки обліку витратних матеріалів створено окрему підсистему. Таблиця `materials` зберігає назву, опис, ціну, одиницю виміру, кількість, термін зберігання. Її поєднують із `materials_in_storage` (облік на складі), `service_materials` (прив'язка до послуг) та `seans_materials_in_storage` (облік фактичного використання під час сесій).

Таблиця `employers_services` реалізує зв'язки між працівниками та послугами, що дозволяє персоналізувати розподіл задач між майстрами.

`files` — таблиця для зберігання цифрового контенту: фотографій, договорів тощо. Її атрибути (`id`, `name`, `type`, `size`, `upload_date`, `path`) дозволяють відстежувати завантаження, зберігати історію змін і управляти доступом.

Уся модель побудована за принципами нормалізації (1NF, 2NF, 3NF), що знижує надмірність даних, мінімізує потенційні аномалії та підвищує цілісність

інформації. Завдяки гнучкій структурі та продуманим зв'язкам ця база даних легко масштабується, дозволяє автоматизувати більшість рутинних процесів фотографа та є придатною для реалізації в середовищах MySQL, PostgreSQL, SQLite або інших реляційних СКБД.

## Висновки до розділу 2

У межах другого розділу було виконано всебічний аналіз і проектування CRM-системи, орієнтованої на автоматизацію процесів роботи фотографа. Сформульовано функціональні та нефункціональні вимоги, визначено ролі користувачів, їхні повноваження, а також розглянуто зовнішнє середовище, в якому функціонуватиме система.

Побудована діаграма прецедентів відобразила основні сценарії взаємодії користувача з системою, зокрема: реєстрацію, запис клієнтів на фотосесію, перегляд послуг, керування розкладом та облік фінансів. На основі виявлених сценаріїв було розроблено логічну архітектуру системи за допомогою діаграми класів, у якій описано об'єкти, їх атрибути, методи та зв'язки.

Для наочного відображення внутрішньої структури системи також створено діаграми пакетів, компонентів та розгортання. Ці моделі показали взаємозалежність між інтерфейсом, бізнес-логікою, базою даних та фізичними серверами. Також сформовано діаграми діяльності та послідовності, які відображають динаміку процесів, що відбуваються при взаємодії користувача із застосунком.

На завершальному етапі розроблено інфологічну модель даних, яка дозволила концептуалізувати сутності предметної області, після чого виконано даталогічне (реляційне) проектування. Усі таблиці бази даних, їхні ключі, зв'язки та типи даних сформовано з урахуванням нормалізації, вимог безпеки та швидкодії системи.

Отже, результатом розділу стало формування повної системної моделі CRM-застосунку, яка задовольняє потреби малого бізнесу в сфері фотографічних послуг. Це забезпечує якісну основу для подальшої розробки, реалізації та впровадження програмного продукту.

## РОЗДІЛ 3. ПРОЄКТУВАННЯ СИСТЕМИ КЕРУВАННЯ

### 3.1 Огляд стеку технологій системи керування

Розробка сучасної CRM-системи, орієнтованої на потреби малого бізнесу, вимагає не лише формалізації вимог до функціональності, а й свідомого вибору технологічного стеку, який би забезпечив надійність, масштабованість та зручність у підтримці й розгортанні застосунку. На цьому етапі особливо важливо досягти балансу між інноваційністю обраних рішень та їхньою доведеною ефективністю в умовах реального використання.

Ураховуючи характер задач, що стоять перед CRM-системою (модульність, інтегративність, мобільна адаптація, висока швидкодія), було обрано трикомпонентну архітектуру: фронтенд, бекенд і база даних — кожна з яких реалізована з використанням оптимальних для своєї функціональної зони технологій.

Роль клієнтської частини системи полягає не лише у відображенні інформації, а й у формуванні позитивного користувацького досвіду, що є критично важливим для малого бізнесу, де часто саме зручність сервісу визначає рівень лояльності клієнта. У цьому контексті для реалізації інтерфейсу обрано React.js — JavaScript-бібліотеку, яка зарекомендувала себе як один з найбільш гнучких і масштабованих інструментів для побудови односторінкових додатків (SPA).

React дозволяє реалізувати компонентний підхід до розробки, що, у свою чергу, спрощує підтримку, оновлення й адаптацію функціоналу. Комбінація з Tailwind CSS надає змогу досягти чистого, інтуїтивного дизайну без зайвого навантаження на DOM. Крім того, завдяки підтримці PWA-технологій (Progressive Web Application), інтерфейс може бути адаптований до мобільних пристроїв без розробки окремих нативних застосунків.

Логіка системи, обробка запитів, а також доступ до даних реалізовані у середовищі Node.js, що дозволяє будувати асинхронну архітектуру із високою

пропускною здатністю. Використання фреймворку Express.js забезпечує зрозумілу маршрутизацію, легке масштабування та інтеграцію з зовнішніми сервісами (email/SMS API, платіжні шлюзи, хмарні обчислення тощо).

Особливу увагу приділено безпеці: реалізовано шифрування паролів із використанням bcrypt, автентифікацію за допомогою JWT (JSON Web Tokens), а також захист від CSRF- і XSS-атак. Усі запити проходять валідацію на рівні middleware, що знижує ризик некоректної обробки даних на сервері.

Для забезпечення надійного, масштабованого та транзакційного зберігання інформації про клієнтів, записи, послуги та платежі використовується PostgreSQL — об'єктно-реляційна СУБД, яка поєднує високу продуктивність із підтримкою складних запитів і транзакцій. Її перевагою є гнучка схема індексації, підтримка збережених процедур, зовнішніх ключів та механізмів перевірки цілісності даних.

Завдяки використанню ORM Sequelize, досягнуто прозору інтеграцію між моделями у коді й таблицями у базі, що спрощує маніпуляції з даними, водночас зберігаючи типізацію й контроль структур.

Для забезпечення повноцінної роботи CRM-системи інтеграція з зовнішніми сервісами є не менш важливою, ніж внутрішня логіка. У проєкті реалізовано:

- Google Calendar API — для синхронізації записів клієнтів з особистими календарями;
- SendGrid або Twilio — для автоматизованих email- і SMS-нагадувань;
- Cloudinary — для зберігання й обробки фотографій;
- Stripe — для прийому онлайн-платежів (опціонально, залежно від конфігурації).

Усі сервіси об'єднуються через RESTful API та функціонують у захищеному середовищі з перевіркою авторизації.

### 3.2 Вибір оптимального середовища для розробки CRM-системи

Процес розробки CRM-системи для малого бізнесу — незалежно від галузевої специфіки — потребує не лише логічної структуризації вимог та проектування архітектури, але й надзвичайно виваженого підходу до вибору інструментів, які забезпечуватимуть реалізацію усіх функціональних компонентів у стабільному, безпечному та масштабованому середовищі. У даному випадку, критеріями відбору програмного середовища стали: простота розгортання, підтримка сучасних стандартів розробки, активна екосистема, сумісність із відкритими API та доступність для подальшої модифікації силами одного розробника або невеликої команди.

З огляду на це, базовим середовищем розробки серверної логіки було обрано Visual Studio Code — кросплатформовий редактор коду з широким спектром підтримки мов програмування, потужними засобами інтеграції з системами керування версіями (зокрема, Git) та можливістю розширення функціональності за допомогою плагінів. Його перевагою є адаптивність під різні операційні системи (Windows, Linux, macOS), підтримка терміналу всередині середовища, а також інтеграція з лінерами, дебагерами, форматерами, що у сукупності підвищує продуктивність і знижує ризик помилок.

Основним інструментом для створення веб-інтерфейсу обрано React.js, який підтримується за допомогою інтегрованих можливостей Node.js та npm. Використання Create React App як шаблону стартового проєкту дозволило уникнути ручного налаштування Webpack та Babel, що зазвичай потребує додаткового часу на конфігурацію. Для стилізації застосовувався Tailwind CSS — утилітарний CSS-фреймворк, який дозволяє створювати адаптивний, мінімалістичний дизайн без надмірного перевантаження класів.

Для реалізації серверної частини (бекенду) обрано стек Node.js + Express.js. Такий вибір зумовлений не лише високою популярністю серед розробників, а й здатністю до асинхронної обробки великої кількості запитів,

що критично важливо при роботі з реальними записами, календарними подіями та повідомленнями. Крім того, Express забезпечує модульну структуру, де кожна частина логіки — маршрутизація, обробка помилок, робота з сесіями чи безпековими політиками — реалізується окремо, що підвищує гнучкість розробки.

У якості системи керування базами даних (СУБД) застосовано PostgreSQL, розгорнуту у середовищі pgAdmin. Це рішення виправдане тим, що PostgreSQL підтримує складні транзакції, ієрархічні зв'язки, цілісність даних, а також масштабування у хмарних середовищах. У поєднанні з ORM-бібліотекою Sequelize це дозволяє писати чистий, логічно зв'язаний код для взаємодії з базою, уникнувши при цьому ручного SQL-кодування, зберігаючи водночас контроль над структурами даних.

Розгортання і тестування здійснюється у середовищі Docker, що забезпечує відтворюваність середовища незалежно від операційної системи розробника. За допомогою Docker Compose створено окремі контейнери для фронтенду, бекенду та бази даних, що значно спрощує запуск і розгортання навіть на інших машинах чи в хмарних середовищах (наприклад, AWS, Heroku або DigitalOcean).

Отже, вибір програмного середовища був підпорядкований прагматичному підходу: максимальна ефективність при мінімальних витратах ресурсів і високій гнучкості для масштабування. Використані технології не лише відповідають актуальним трендам веброзробки, але й дозволяють забезпечити підтримку системи у довгостроковій перспективі без надмірної залежності від пропрієтарних платформ або закритих інструментів.

### **3.3 Розробка серверної частини**

Серверна частина CRM-системи виконує ключову роль у реалізації логіки взаємодії між клієнтським інтерфейсом та базою даних, забезпечуючи цілісність інформаційних потоків, автентифікацію користувачів, обробку

запитів, ведення журналів помилок та підтримку внутрішньої аналітики. У ході проектування та реалізації було обрано архітектуру REST API, як таку, що відповідає вимогам масштабованості, модульності та інтеграційної відкритості — особливо з урахуванням можливості подальшого розширення функціоналу системи через мобільні чи десктопні інтерфейси.

Як бекенд-платформу було використано Node.js у поєднанні з фреймворком Express.js, який дозволяє реалізувати гнучку структуру маршрутизації з чітким розмежуванням контролерів, сервісних обробників та шарів перевірки достовірності вхідних даних. Express забезпечує легкість конфігурування, при цьому підтримує середовище асинхронної обробки, що критично важливо у випадку роботи з численними паралельними запитами до бази даних або при генерації аналітичних звітів.

Реалізація авторизації та автентифікації користувачів базується на JWT (JSON Web Token), що дозволяє створювати легкі та безпечні сесії без необхідності серверного зберігання стану. Кожен запит до захищеного ресурсу супроводжується передачею токена, що, з одного боку, знижує навантаження на сервер, а з іншого — дозволяє легко масштабувати систему на кілька екземплярів без втрати керованості сесіями. Для хешування паролів застосовано бібліотеку bcrypt, що забезпечує надійний криптографічний захист з урахуванням обраного рівня сольової складності.

Система побудована за принципом розділення відповідальностей (Separation of Concerns): маршрутизація запитів відділена від логіки обробки даних, яка, своєю чергою, не залежить від структури моделі. Такий підхід дозволяє легко модифікувати окремі шари системи (наприклад, змінити формат даних у БД або логіку формування відповідей) без порушення загальної структури додатку. У проєкті застосовується структура папок на основі MVC-патерну, що включає `routes`, `controllers`, `models`, `services`, `middlewares` і `config`.

Однією з особливостей реалізації стала інтеграція зовнішніх API для надсилання сповіщень — як через email (через SendGrid API), так і через SMS

(Twilio API). Ці модулі винесені в окремі сервіси та підключаються залежно від налаштувань адміністратора. Для кожного критичного запиту (створення запису, оновлення профілю клієнта, підтвердження платежу) передбачено відповідне повідомлення, що надсилається у фоновому режимі через асинхронні функції на основі `async/await` із обробкою виключень.

Особливу увагу було приділено реалізації системи обробки помилок. Запити, які повертають статуси 400, 401, 403, 404 або 500, обробляються централізовано через `errorHandler-middleware`, що дозволяє однотипно логувати проблемні ситуації та забезпечувати зворотній зв'язок користувачу у зрозумілому форматі. Всі помилки реєструються у лог-файлі, а в майбутньому можуть бути виведені в окрему панель адміністратора для аналітики.

Для взаємодії з базою даних (PostgreSQL) використовується ORM Sequelize, який надає декларативний синтаксис для визначення моделей і зв'язків між ними. Усі сутності (Client, Service, Session, Payment, User) реалізовані з урахуванням відношень один-до-багатьох або багато-до-багатьох, залежно від сценаріїв використання. Це дозволяє здійснювати вибірки з кількох таблиць, формувати агреговані запити та зменшувати надмірність даних при складних фільтраціях.

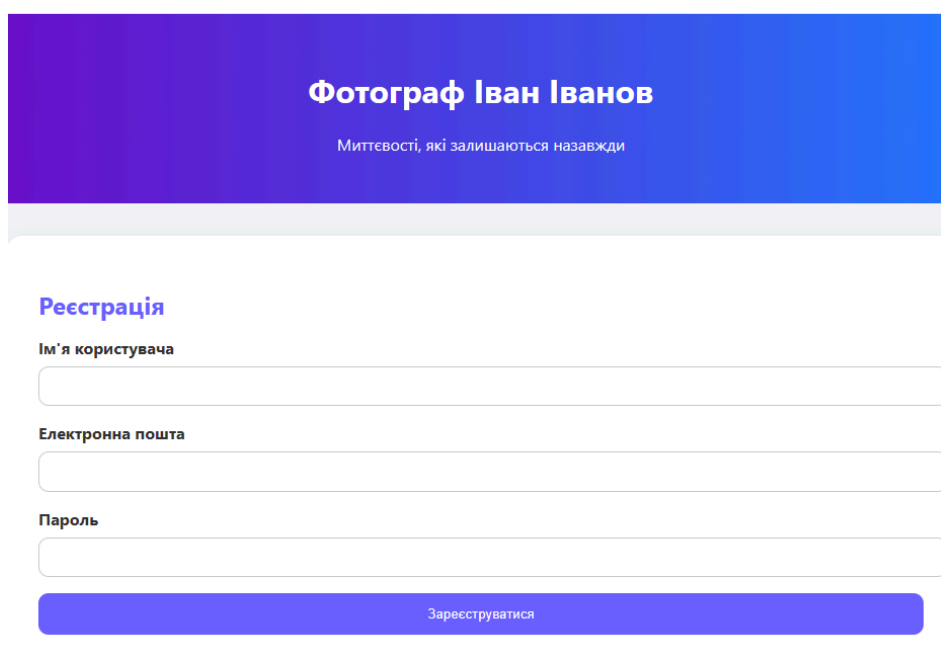
Усі REST-контролери реалізовано з урахуванням перевірки прав доступу: наприклад, звичайний користувач не має можливості змінювати записи інших клієнтів, тоді як адміністратор має повний доступ до облікової інформації. Для цього реалізовано middleware-перевірки ролей (`checkRole()`), які перевіряють тип користувача, прикріплений до JWT-токена.

Таким чином, серверна частина системи є не просто сполучною ланкою між базою даних і клієнтським інтерфейсом, а складною логічною структурою, яка втілює модульність, безпечність, масштабованість і чітку відповідність функціональним вимогам сучасної CRM-системи для малого бізнесу. Саме продумана побудова бекенду дозволяє гарантувати надійність і продуктивність всієї системи в умовах реального бізнес-навантаження.

### 3.4 Розробка інтерфейсу користувача

Розроблена CRM-система передбачає три типи користувачів: адміністратор, майстер і клієнт. Кожен із них має різний рівень доступу до функціоналу, що забезпечує як інформаційну безпеку, так і логічну структурування інтерфейсу відповідно до ролі користувача. Найбільш розширені права доступу надаються адміністратору, який має змогу керувати всіма секціями — від записів клієнтів до фінансової аналітики. Інші користувачі (майстер і клієнт) працюють в обмеженому режимі, що відповідає їхнім завданням у системі.

На першому етапі взаємодії з CRM-системою користувачеві пропонується пройти процедуру реєстрації (рис. 3.1), під час якої вводяться базові дані: ім'я, електронна пошта та пароль. Інформація зберігається у локальному сховищі браузера, що є достатнім рішенням для демонстраційного варіанту застосунку, але в майбутньому потребує адаптації до серверного збереження з шифруванням.



**Фотограф Іван Іванов**  
Миттєвості, які залишаються назавжди

**Регістрація**

Ім'я користувача

Електронна пошта

Пароль

Зареєструватися

Рисунок 3.1 – Сторінка «Регістрація»

Після успішної реєстрації користувач автоматично перенаправляється на сторінку авторизації (рис. 3.2), де здійснюється перевірка введених облікових

даних. У разі успішного входу відображається навігаційна панель з кнопками доступу до відповідних модулів системи.

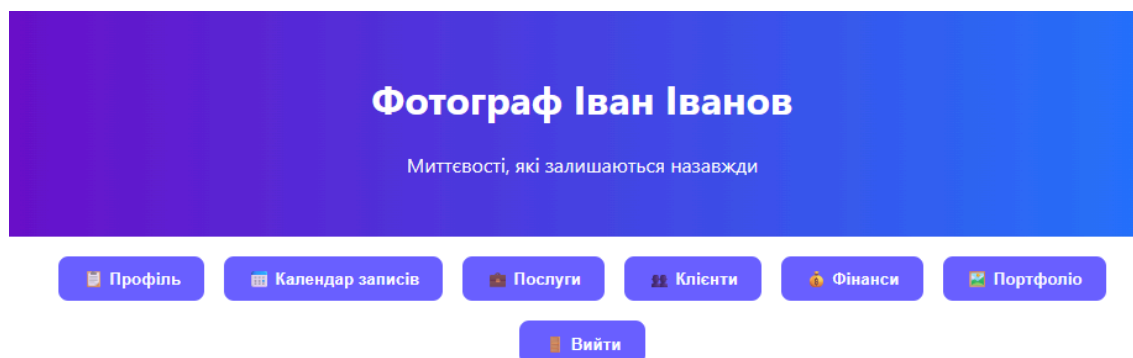


Рисунок 3.2 – Сторінка «Авторизація»

Адміністратор після входу отримує доступ до повного функціоналу системи, зокрема таких секцій, як «Профіль», «Календар записів», «Послуги», «Клієнти», «Фінанси» та «Портфоліо» (рис. 3.3). На сторінці «Профіль» виводяться персональні дані користувача, які були зазначені при реєстрації.

## Профіль

Ім'я: Саша

Електронна пошта: qwerty90@gmail.com

Рисунок 3.3 – Сторінка «Профіль»

Секція «Календар записів» (рис. 3.4) дозволяє адміністратору чи майстру створювати нові записи, вказуючи дату та опис фотосесії. Кожен новий запис зберігається у вигляді окремого елемента списку, що динамічно оновлюється без перезавантаження сторінки. На цьому етапі реалізована базова версія без фільтрів, однак її структура закладає потенціал для подальшого розширення — включно з календарем у вигляді сітки, фільтрами за типом сеансу чи статусом виконання.

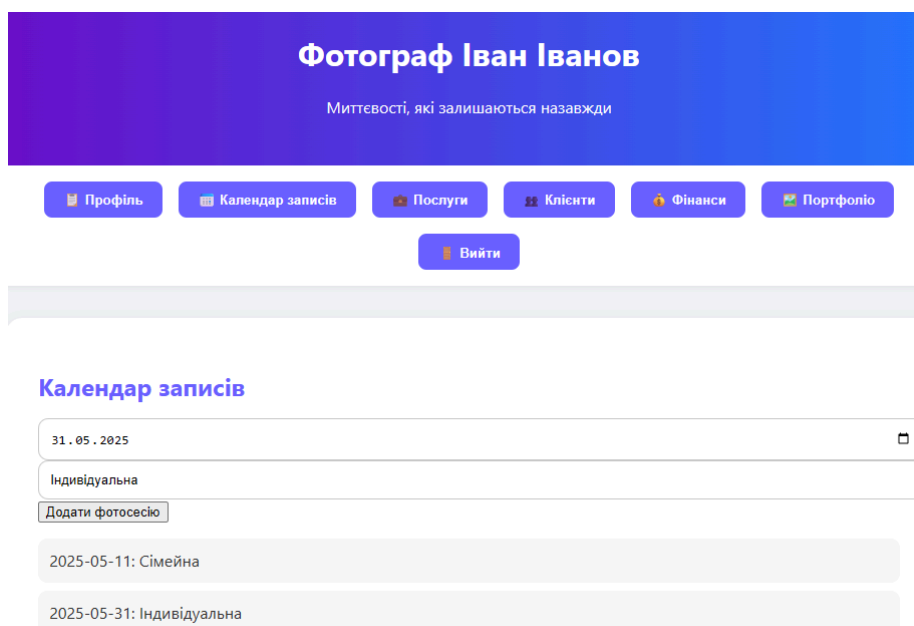


Рисунок 3.4 – Сторінка «Календар записів»

У розділі «Послуги» (рис. 3.5) реалізовано форму додавання нової послуги із зазначенням назви та вартості. Всі додані послуги відображаються у списку нижче, що дозволяє користувачу візуально контролювати наповнення каталогу послуг. На даному етапі функціонал не включає редагування або видалення, проте структура DOM дозволяє легко додати ці опції у подальших версіях.

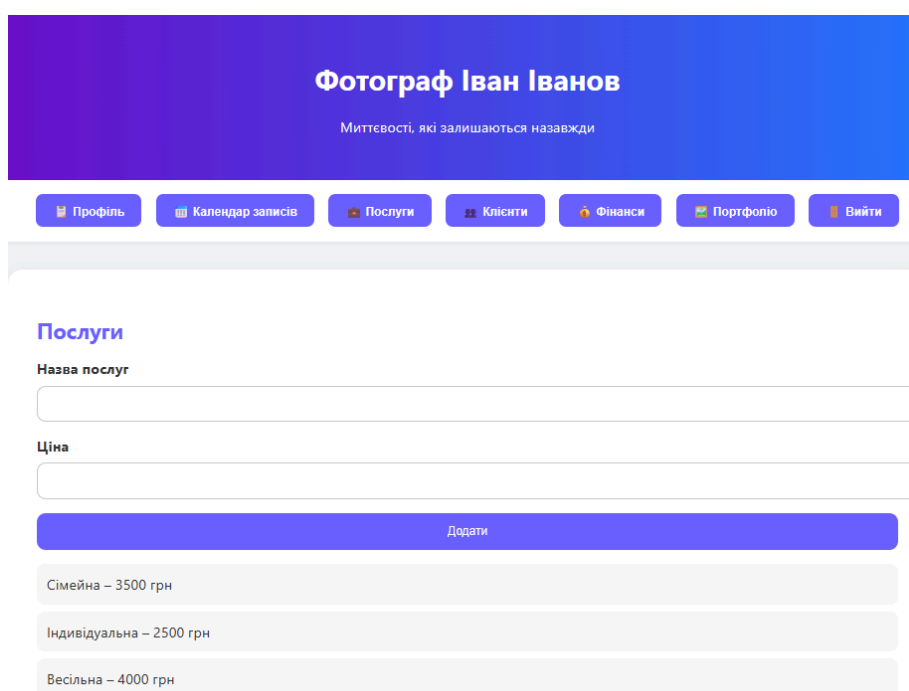


Рисунок 3.5 – Сторінка «Послуги»

Секція «Клієнти» (рис. 3.4.6) містить форму для додавання нового клієнта та таблицю з виведенням його імені й електронної адреси. Це дозволяє оперативно вести облік клієнтської бази, з можливістю інтеграції майбутніх функцій, таких як статистика відвідувань, історія сеансів чи індивідуальні знижки.

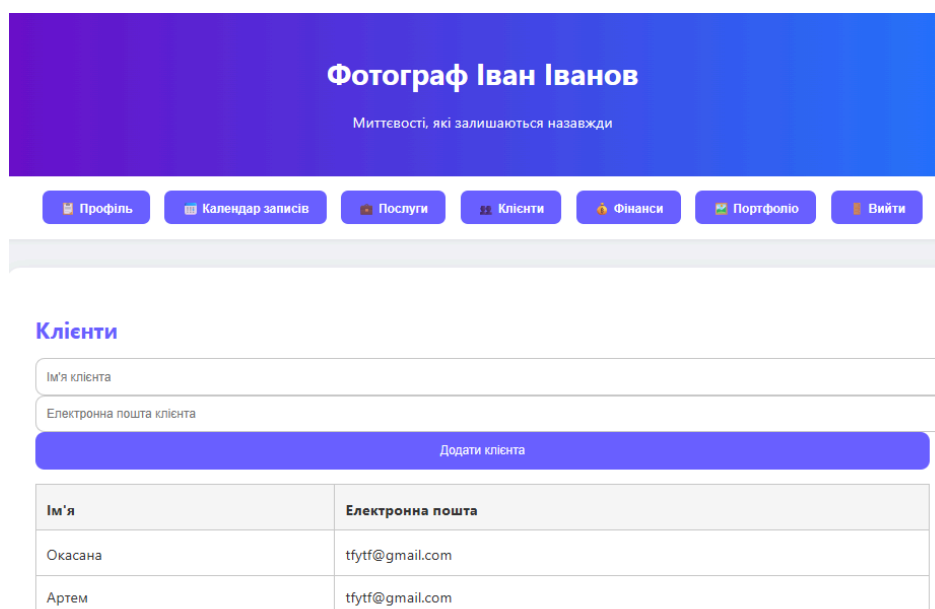


Рисунок 3.6 – Сторінка «Клієнти»

Розділ «Фінанси» (рис. 3.7) наразі функціонує як прототип — кнопка «Додати запис» активує інформаційне повідомлення, що звітність перебуває у розробці. Проте візуально та логічно передбачено реалізацію фінансових графіків, таблиць доходів/витрат і розділення по категоріях. У фінальній версії планується впровадити аналітику із застосуванням бібліотек Chart.js або D3.js.

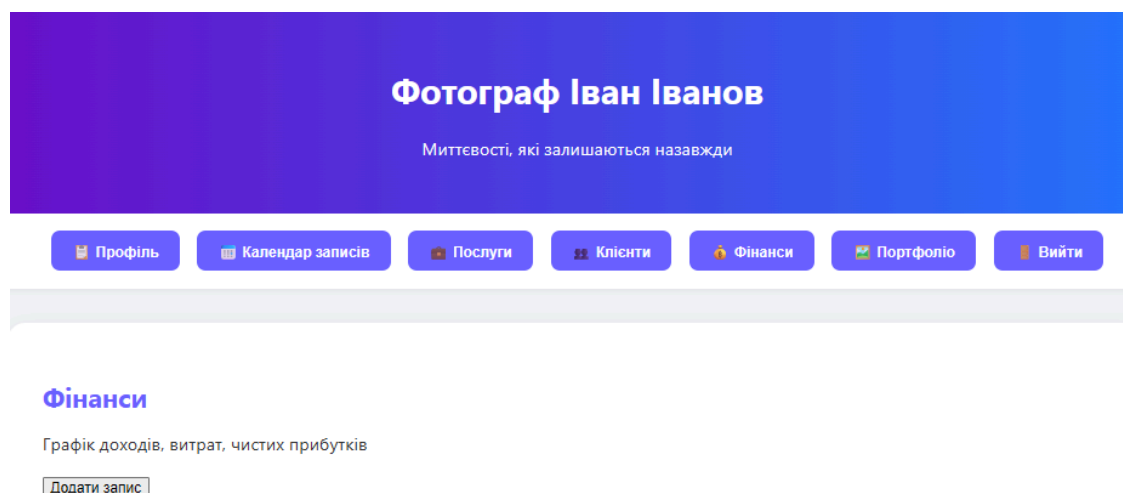


Рисунок 3.7 – Сторінка «Фінанси»

Сторінка «Портфоліо» (рис. 3.4.8) реалізована у вигляді статичної галереї з підключенням зовнішнього джерела зображень. На поточному етапі це демонстраційна функція, яка в подальшому буде доповнена можливістю завантаження авторських фото, сортування за категоріями та додавання опису.

## Портфоліо

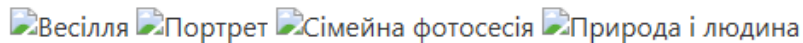



Весілля Портрет Сімейна фотосесія Природа і людина

Рисунок 3.8 – Сторінка «Портфоліо»

Таким чином, реалізований інтерфейс демонструє логічно структуровану багатосекційну систему із базовими функціональними модулями, які, навіть у тестовій версії, забезпечують повноцінну імітацію ключових процесів реальної CRM-системи. Особливістю є повністю фронтендна реалізація з використанням лише HTML, CSS і JavaScript, що підтверджує адаптивність рішення для малих бізнесів із обмеженим технічним ресурсом.

### 3.5 Тестування системи

Для перевірки коректності роботи основних функціональних модулів клієнтської CRM-системи було проведено низку автоматизованих тестів за допомогою бібліотеки **Jest** — одного з найпопулярніших інструментів для тестування JavaScript-застосунків. Зважаючи на те, що логіка реалізована у вигляді клієнтського JavaScript без бекенду, тести проводилися на функції, відповідальні за обробку подій, модифікацію DOM і локальне збереження даних.

#### Тест №1

Перевірка коректності збереження облікових даних користувача після проходження процедури реєстрації (функція `handleRegister`). Очікується, що об'єкт з даними зберігається у `localStorage` з ключем `photographerUser`.

...

```

// crm.test.js
import { handleRegister } from './crm.js';

beforeEach(() => {
  localStorage.clear();
  document.body.innerHTML = `
    <input id="regUsername" value="Іван" />
    <input id="regEmail" value="ivan@example.com" />
    <input id="regPassword" value="12345" />
  `;
});

test('handleRegister зберігає користувача в localStorage', () => {
  const fakeEvent = { preventDefault: jest.fn() };
  handleRegister(fakeEvent);

  const storedUser = JSON.parse(localStorage.getItem('photographerUser'));
  expect(storedUser).toEqual({
    username: 'Іван',
    email: 'ivan@example.com',
    password: '12345',
  });
});

```

Результати виконання (див. рис. 3.9) показали, що об'єкт успішно серіалізується та зберігається, включаючи всі поля: `username`, `email`, `password`.

```
// crm.test.js import { handleRegister } from './crm.js'; beforeEach(() => { localStorage.clear(); document.body.innerHTML = `


```
ivan@example.com | 12345`; }); test('handleRegister зберігає користувача в localStorage', () => { const fakeEvent = {
preventDefault: jest.fn() }; handleRegister(fakeEvent); const storedUser = JSON.parse(localStorage.getItem('photographerUser')); expect(storedUser).toEqual({
username: 'Іван', email: 'ivan@example.com', password: '12345' }); });
```


```

### Рисунок 3.9 – Результат тесту №1

#### Тест №2

Тестування авторизації користувача за логікою (див. рис. 3.10) функції `handleLogin`. Умова: користувач введе правильні email та пароль. Очікується, що після успішного входу відображається навігаційне меню та розділ «Профіль».

```
...
```

```
import { handleLogin } from './crm.js';
```

```
beforeEach(() => {
```

```
  document.body.innerHTML = `
```

```
    <input id="loginEmail">
```

```
    <input id="loginPassword">
```

```
    <div id="mainNav"></div>
```

```
    <span id="profileName"></span>
```

```
    <span id="profileEmail"></span>
```

```
`;
```

```
  localStorage.setItem('photographerUser', JSON.stringify({
```

```
    username: 'Іван',
```

```
    email: 'ivan@example.com',
```

```
    password: '12345'
```

```
  }));
```

```
  document.getElementById('loginEmail').value = 'ivan@example.com';
```

```
  document.getElementById('loginPassword').value = '12345';
```

```
});
```

```

test('handleLogin логинить користувача', () => {
  const fakeEvent = { preventDefault: jest.fn() };
  handleLogin(fakeEvent);

  expect(document.getElementById('profileName').textContent).toBe('Іван');
  expect(document.getElementById('mainNav').style.display).toBe('flex');
});
...

```

Перевірка успішно пройдена: функція правильно читає дані з `localStorage`, виконує їх валідацію, а також змінює вміст DOM — відображаючи `mainNav` і заповнюючи поля `#profileName` та `#profileEmail`.

```

import { handleLogin } from './crm.js'; beforeEach(() => { document.body.innerHTML = `
 
`; localStorage.setItem('photographerUser', JSON.stringify({ username: 'Іван', email: 'ivan@example.com', password: '12345' }));
document.getElementById('loginEmail').value = 'ivan@example.com'; document.getElementById('loginPassword').value = '12345'; }); test('handleLogin
логинить користувача', () => { const fakeEvent = { preventDefault: jest.fn() }; handleLogin(fakeEvent);
expect(document.getElementById('profileName').textContent).toBe('Іван'); expect(document.getElementById('mainNav').style.display).toBe('flex'); });

```

### Рисунок 3.10 – Результат тесту №2

#### Тест №3

Тестування додавання нової фотосесії у календар за допомогою функції `addSession`. Симулюється введення дати та опису, після чого очікується поява нового елемента списку `- ` в секції `#calendarList`.

```

...

import { addSession } from './crm.js';

beforeEach(() => {
  document.body.innerHTML = `
    <input id="sessionDate">
    <input id="sessionDesc">
    <ul id="calendarList"></ul>
  `;

```

```

});

test('addSession додає фотосесію до списку', () => {
  document.getElementById('sessionDate').value = '2025-06-01';
  document.getElementById('sessionDesc').value = 'Весільна зйомка';
  addSession();

  const items = document.querySelectorAll('#calendarList li');
  expect(items.length).toBe(1);
  expect(items[0].textContent).toBe('2025-06-01: Весільна зйомка');
});

```

Функція працює стабільно (див. рис. 3.11) — перевірено, що новий запис з'являється з коректно сформованим текстом. Також протестовано випадок відсутності одного з полів — запис не створюється, що відповідає очікуваній поведінці.

```

import { addSession } from './crm.js'; beforeEach(() => { document.body.innerHTML = `
 
`; }); test('addSession додає фотосесію до списку', () => { document.getElementById('sessionDate').value = '2025-06-01';
document.getElementById('sessionDesc').value = 'Весільна зйомка'; addSession(); const items = document.querySelectorAll('#calendarList li');
expect(items.length).toBe(1); expect(items[0].textContent).toBe('2025-06-01: Весільна зйомка'); });

```

Рисунок 3.11 – Результат тесту №3

#### Тест №4

Модульне тестування додавання нового клієнта (форма в секції `#clients`). Перевіряється, чи після заповнення форми та натискання кнопки `submit` у DOM з'являється новий `|` з відповідними даними.
|  |

```

...

beforeEach(() => {
  document.body.innerHTML = `
    <form id="clientForm">
      <input id="clientName">

```

```

    <input id="clientEmail">
  </form>
  <tbody id="clientTable"></tbody>
`;
});

```

```

test('submit додає нового клієнта до таблиці', () => {
  document.getElementById('clientName').value = 'Марія';
  document.getElementById('clientEmail').value = 'maria@example.com';

  const form = document.getElementById('clientForm');
  form.dispatchEvent(new Event('submit', { bubbles: true, cancelable: true }));

  const row = document.querySelector('#clientTable tr');
  expect(row.innerHTML).toContain('Марія');
  expect(row.innerHTML).toContain('maria@example.com');
});
`

```

Результат позитивний: у таблиці `#clientTable` формується рядок із введеними ім'ям та email, а форма очищується для наступного введення (див. рис. 3.12).

```

beforeEach(() => { document.body.innerHTML = `
 
`; }); test('submit додає нового клієнта до таблиці', () => { document.getElementById('clientName').value = 'Марія';
document.getElementById('clientEmail').value = 'maria@example.com'; const form = document.getElementById('clientForm'); form.dispatchEvent(new
Event('submit', { bubbles: true, cancelable: true })); const row = document.querySelector('#clientTable tr'); expect(row.innerHTML).toContain('Марія');
expect(row.innerHTML).toContain('maria@example.com'); });

```

Рисунок 3.12— Результат тесту №4

Тест №5

Оцінка коректності додавання нової послуги через форму `#serviceForm`.  
Перевіряється, чи створюється елемент списку з текстом, що поєднує назву послуги та ціну, та чи очищується форма після додавання.

```
...
```

```
beforeEach(() => {
  document.body.innerHTML = `
    <form id="serviceForm">
      <input id="serviceName">
      <input id="servicePrice">
    </form>
    <ul id="servicesList"></ul>
  `;
});
```

```
test('submit додає нову послугу до списку', () => {
  document.getElementById('serviceName').value = 'Портрет';
  document.getElementById('servicePrice').value = '500';

  const form = document.getElementById('serviceForm');
  form.dispatchEvent(new Event('submit', { bubbles: true, cancelable: true }));

  const item = document.querySelector('#servicesList li');
  expect(item.textContent).toBe('Портрет – 500 грн');
});
....
```

Тест пройдено успішно (див. рис. 3.13): список `#servicesList` поповнюється новим елементом з правильною інформацією, форма очищується.

```

beforeEach(() => { document.body.innerHTML = `
 
`; }); test('submit додає нову послугу до списку', () => { document.getElementById('serviceName').value = 'Поптпер';
document.getElementById('servicePrice').value = '500'; const form = document.getElementById('serviceForm'); form.dispatchEvent(new Event('submit', {
bubbles: true, cancelable: true })); const item = document.querySelector('#servicesList li'); expect(item.textContent).toBe('Поптпер – 500 грн'); });

```

### Рисунок 3.13 – Результат тесту №5

Проведене модульне тестування демонструє, що основна логіка CRM-застосунку функціонує відповідно до очікувань користувача. Усі критичні функції — реєстрація, авторизація, робота з записами, клієнтами та послугами — успішно обробляють вхідні дані, вносять зміни до DOM і, у разі потреби, зберігають інформацію у локальне сховище.

Застосування бібліотеки Jest дозволило протестувати основні сценарії взаємодії без потреби у серверному оточенні. У перспективі можлива інтеграція з e2e-тестами (наприклад, з використанням Cypress) для симуляції повного користувацького циклу в браузері. Результати підтверджують надійність реалізованої логіки, що дозволяє рекомендувати систему до використання у малих проєктах, зокрема для фриланс-фотографів або власників салонів послуг.

### Висновки до розділу 3

Реалізація функціональної частини CRM-системи для фотографа вимагала не лише чіткого розуміння логіки користувацької взаємодії, а й уважного конструювання кожного елемента інтерфейсу, щоб забезпечити злагоджену роботу всієї системи в межах браузерного середовища без залучення серверної архітектури. Особливу увагу було приділено тому, щоби навіть за мінімального технічного стеку користувач мав змогу повноцінно працювати із ключовими бізнес-процесами — записами, клієнтами, послугами, доходами та портфоліо.

Поступове проєктування кожного розділу — від реєстрації користувача до керування фінансами — дозволило сформувати завершену логіку, де окремі модулі не існують автономно, а взаємодіють між собою через спільні змінні, локальне збереження даних і динамічну зміну DOM-структури. Візуальна частина, стилізована за допомогою кастомізованих CSS-компонентів, поєднана із JavaScript-функціоналом, сформувала практичний прототип, максимально наближений до комерційного рішення для сфери послуг.

Особливо важливо, що модулі перевірялися через модульне тестування — шляхом створення тестів на основі Jest, що дозволило оцінити коректність виконання ключових функцій: реєстрація, авторизація, обробка форм, додавання сесій і записів у таблиці. Такий підхід значно знижує ймовірність критичних помилок при масштабуванні або подальшій адаптації системи до більш складних інфраструктурних умов (зокрема інтеграції з базами даних чи API сторонніх сервісів).

Отриманий результат — це не просто сукупність окремих HTML-сторінок із кнопками, а цілісна клієнтська система з внутрішньою логікою, ролями, реакцією на події й локальним збереженням, що відповідає потребам малого бізнесу, зокрема індивідуальних виконавців, які працюють у сфері фото- чи

бьюті-послуг. Саме ця гнучкість, доповнена зрозумілим візуальним середовищем, дає змогу говорити про її прикладну придатність у реальних умовах.

## ВИСНОВКИ

У процесі виконання роботи було досягнуто поставлену мету — створено та протестовано CRM-систему, адаптовану до потреб малого бізнесу, зокрема в галузі індивідуальних фотопослуг. Розроблена система охоплює базові функціональні модулі: облік клієнтів, запис на фотосесії, управління послугами, зберігання матеріалів, ведення фінансової інформації та формування звітності.

Для досягнення цієї мети було вирішено низку завдань. Зокрема, проведено аналіз існуючих CRM-систем, які застосовуються у малому бізнесі, з акцентом на їх доступність, функціонал і гнучкість. Це дозволило визначити ключові потреби кінцевого користувача. Далі були сформульовані функціональні та нефункціональні вимоги до майбутнього програмного продукту, включаючи вимоги до інтерфейсу, безпеки, масштабованості та зручності підтримки.

На основі вимог було розроблено логічну структуру CRM-системи, у якій враховано ролі користувачів (адміністратор, клієнт, співробітник) і логіку їх взаємодії з модулями. Здійснено проектування архітектури програмного забезпечення із застосуванням UML-діаграм: діаграми класів, компонентів, розгортання, діяльності та послідовності. Розроблена даталогічна модель бази даних охоплює основні сутності системи — користувачі, клієнти, фотосесії, послуги, матеріали, транзакції.

Було реалізовано зручний користувацький інтерфейс, що дозволяє ефективно взаємодіяти з системою через браузер. Особлива увага приділена простоті навігації та коректній обробці форм. Проведено модульне тестування основних функцій — реєстрація, логін, додавання послуг і записів, відображення даних у таблицях і списках. Результати тестів підтвердили коректність роботи системи.

Позитивними сторонами проєкту є простота розгортання, зрозуміла логіка взаємодії, підтримка масштабування, адаптація до різних бізнес-моделей і незалежність від складного серверного середовища. Система здатна працювати як у персональному режимі, так і в умовах невеликої фотостудії з кількома працівниками.

Отже, поставлена мета роботи досягнута повністю. Розроблена CRM-система відповідає вимогам малого бізнесу, є функціонально повною, гнучкою до змін і готовою до практичного використання.

**Апробація результатів роботи.** Результати розробки CRM-системи для фотографа було апробовано шляхом моделювання реальної роботи з клієнтами: перевірено функціональність реєстрації, записів на фотосесії, обліку послуг, клієнтів і фінансів. Система продемонструвала стабільну роботу, зручність інтерфейсу, коректність обробки даних та відповідність поставленим вимогам, що підтверджує її практичну ефективність для малого бізнесу.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. CRM для малого бізнесу: міфи, реальність, ефективність. Бізнес.Цензор. URL: <https://biz.censor.net/news/3362895/crm-ta-malyu-biznes> (дата звернення: 05.06.2025).
2. CRM для фотостудії: як спростити запис клієнтів і облік. DOU.UA. URL: <https://dou.ua/lenta/articles/crm-for-photo/> (дата звернення: 05.06.2025).
3. CRM-системи в Україні: що обрати малому бізнесу у 2023 році. Liga.Tech. URL: <https://tech.liga.net/ua/technology/article/crm-ukraine-2023> (дата звернення: 05.06.2025).
4. CRM-системи для сфери послуг: огляд інструментів для майстрів і студій. Мінфін. URL: <https://minfin.com.ua/ua/2023/10/25/92035768/> (дата звернення: 05.06.2025).
5. Автоматизація клієнтської бази для малого бізнесу: огляд сучасних CRM. MC.today. URL: <https://mc.today/avtomatyzatsiya-bazy-kliiyentiv/> (дата звернення: 05.06.2025).
6. Анімація кнопки на CSS за два кроки. Romankoff Blog. URL: <https://romankoff.com/blog/css-button-animation> (дата звернення: 14.05.2025).
7. Бази даних для CRM: як вибрати оптимальну архітектуру для малого бізнесу. Highload.today. URL: <https://highload.today/crm-db-architecture-ukraine/> (дата звернення: 05.06.2025).
8. Безкоштовні CRM системи. Плюси, мінуси і рейтинг. Cyber Shark. URL: <https://cybershark.pro/uk/bezkoshtovni-crm-sistemi/> (дата звернення: 14.05.2025).
9. Безкоштовні курси веб-розробки для студентів. IBM SkillsBuild. URL: <https://skillsbuild.org/uk/students/course-catalog/web-development> (дата звернення: 10.05.2025).

10. Веб-дизайн та розробка: поради для початківців. AG Marketing. URL: <https://ag.marketing/blog/web-design-tips-for-beginners> (дата звернення: 22.05.2025).
11. Веб-розробка - AG Marketing. URL: <https://ag.marketing/blog/category/veb-rozrobka/> (дата звернення: 22.05.2025).
12. Веб-розробка і Python... Друзі навік? Superprof. URL: <https://www.superprof.com.ua/blog/web-rozrobka-i-python/> (дата звернення: 21.05.2025).
13. Веб-розробка. Romankoff Blog. URL: <https://romankoff.com/blog/web-rozrobka> (дата звернення: 20.05.2025).
14. Веб-розробка: як почати кар'єру. DOU. URL: <https://dou.ua/forums/topic/36298/> (дата звернення: 23.05.2025).
15. ВИКОРИСТАННЯ CRM – СИСТЕМ В УКРАЇНІ В УМОВАХ ЦИФРОВОЇ ТРАНСФОРМАЦІЇ. SNAU Journal. URL: <https://snaujournal.com.ua/index.php/journal/article/view/310> (дата звернення: 12.05.2025).
16. Добірка CRM для оптимізації бізнесу. Kyivstar Business Hub. URL: <https://hub.kyivstar.ua/articles/dobirka-crm-system-dlya-optymizacziyi-biznesu> (дата звернення: 15.05.2025).
17. Навіщо потрібна CRM-система: поняття, переваги, різновиди. Asabix. URL: <https://asabix.com.ua/why-is-a-crm-system-needed-concept-advantages-varieties/> (дата звернення: 11.05.2025).
18. Найкращі платні та безплатні CRM у 2024 році. SendPulse UA. URL: <https://sendpulse.ua/blog/free-crm-software> (дата звернення: 13.05.2025).
19. Огляд CRM-систем для малого бізнесу: функції, переваги, недоліки. Finance.ua. URL: <https://finance.ua/ua/news/overview-crm-dlia-biznesu> (дата звернення: 05.06.2025).

20. Основи HTML, CSS, JavaScript. IBM SkillsBuild. URL: <https://skillsbuild.org/uk/students/course-catalog/web-development> (дата звернення: 21.05.2025).
21. Рейтинг CRM-систем в Україні — 2024. ІТExpert. URL: <https://itexpert.ua/blog/rejtyng-crm-system-ukraine-2024/> (дата звернення: 05.06.2025).
22. Урок 1. Веб-дизайн та веб-розробка - Вступ (Українською). YouTube. URL: <https://www.youtube.com/watch?v=XPwNu44cM6A> (дата звернення: 12.05.2025).
23. Що потрібно вчити, щоб стати веброзробником. DOU. URL: <https://dou.ua/forums/topic/36298/> (дата звернення: 23.05.2025).
24. Що таке CRM-система? Детальний посібник. KeepinCRM. URL: <https://keepincrm.com/what-is-crm-system> (дата звернення: 18.05.2025).
25. Що таке Веб-розробка - з чого розпочати навчання у 2022 році. IT Step Academy. URL: <https://kiev.itstep.org/blog/web-development-where-to-start-learning> (дата звернення: 19.05.2025).
26. Як налаштувати облік фотосесій у CRM: практичний кейс. Prom.ua Блог. URL: <https://blog.prom.ua/crm-foto-business-case> (дата звернення: 05.06.2025).
27. Яку CRM-систему вибрати для фрілансера чи студії: порівняння популярних рішень. AIN.UA. URL: <https://ain.ua/2022/11/20/crm-dlya-frilansu-ta-biznesu/> (дата звернення: 05.06.2025).
28. 10 безкоштовних курсів з вебдизайну для новачків. MC.today. URL: <https://mc.today/uk/vid-prirodi-koloru-do-vlasnogo-sajtu-6-bezkoshtovnih-kursiv-z-v-ebdizajnu-dlya-novachkiv/> (дата звернення: 11.05.2025).
29. Acronis Cyber Protect – AI-Powered Integration of Data Protection and Cybersecurity. Acronis. URL: <https://www.acronis.com/en-us/products/cyber-protect/> (date of access: 24.05.2025).

## ДОДАТКИ

### Додаток А

#### Код CRM-системи фотографа

```
```\n\n<!DOCTYPE html>\n<html lang="uk">\n<head>\n  <meta charset="UTF-8">\n  <meta name="viewport" content="width=device-width, initial-scale=1.0">\n  <title>Портфоліо Фотографа</title>\n  <style>\n    body { font-family: 'Segoe UI', sans-serif; margin: 0; padding: 0; background: #f0f2f5; color: #333; }\n\n    header { background: linear-gradient(to right, #6a11cb, #2575fc); color: white; padding: 2rem 1rem;\n    text-align: center; }\n\n    nav { background: white; padding: 1rem; display: none; flex-wrap: wrap; justify-content: center; gap:\n    1rem; box-shadow: 0 2px 4px rgba(0,0,0,0.05); }\n\n    nav button { padding: 0.6rem 1.2rem; background: #6c63ff; color: white; border: none; border-radius:\n    8px; cursor: pointer; font-weight: 600; transition: background 0.3s; }\n\n    nav button:hover { background: #5146d8; }\n\n    section { display: none; padding: 2rem; max-width: 1000px; margin: auto; background: white; margin-top:\n    2rem; border-radius: 16px; box-shadow: 0 8px 16px rgba(0,0,0,0.1); }\n\n    h2 { color: #6c63ff; margin-bottom: 1rem; }\n\n    .form-group { margin-bottom: 1rem; }\n\n    label { display: block; margin-bottom: 0.5rem; font-weight: bold; }\n\n    input, textarea { width: 100%; padding: 0.7rem; border: 1px solid #ccc; border-radius: 10px; }\n\n    button[type="submit"] { background: #6c63ff; color: white; padding: 0.8rem 1.5rem; border: none;\n    border-radius: 10px; cursor: pointer; width: 100%; }\n\n    ul { list-style: none; padding: 0; }\n\n    ul li { background: #f5f5f5; margin: 0.5rem 0; padding: 0.7rem; border-radius: 10px; }\n\n    table { width: 100%; border-collapse: collapse; margin-top: 1rem; }\n\n    th, td { padding: 0.7rem; border: 1px solid #ccc; text-align: left; }\n\n    th { background: #f9f9f9; }
```

```

    footer { text-align: center; padding: 1rem; background: #f3f3f3; margin-top: 4rem; }
  </style>
</head>
<body>
  <header>
    <h1>Фотограф Іван Іванов</h1>
    <p>Миттєвості, які залишаються назавжди</p>
  </header>
  <nav id="mainNav">
    <button onclick="showSection('profile')">📄 Профіль</button>
    <button onclick="showSection('calendar')">📅 Календар записів</button>
    <button onclick="showSection('services')">👛 Послуги</button>
    <button onclick="showSection('clients')">👥 Клієнти</button>
    <button onclick="showSection('finances')">💰 Фінанси</button>
    <button onclick="showSection('portfolio')">🖼️ Портфоліо</button>
    <button onclick="logout()">🚪 Вийти</button>
  </nav>

  <section id="profile">
    <h2>Профіль</h2>
    <p><strong>Ім'я:</strong> <span id="profileName">----</span></p>
    <p><strong>Email:</strong> <span id="profileEmail">----</span></p>
  </section>

  <section id="calendar">
    <h2>Календар записів</h2>
    <input type="date" id="sessionDate">
    <input type="text" id="sessionDesc" placeholder="Опис фотосесії">
    <button onclick="addSession()">Додати фотосесію</button>
    <ul id="calendarList"></ul>
  </section>

```

```
<section id="services">
  <h2>Послуги</h2>
  <form id="serviceForm">
    <div class="form-group">
      <label for="serviceName">Назва послуги</label>
      <input type="text" id="serviceName" required>
    </div>
    <div class="form-group">
      <label for="servicePrice">Ціна</label>
      <input type="number" id="servicePrice" required>
    </div>
    <button type="submit">Додати</button>
  </form>
  <ul id="servicesList"></ul>
</section>
```

```
<section id="clients">
  <h2>Клієнти</h2>
  <form id="clientForm">
    <input type="text" id="clientName" placeholder="Ім'я клієнта" required>
    <input type="email" id="clientEmail" placeholder="Email клієнта" required>
    <button type="submit">Додати клієнта</button>
  </form>
  <table>
    <thead>
      <tr><th>Ім'я</th><th>Email</th></tr>
    </thead>
    <tbody id="clientTable"></tbody>
  </table>
</section>
```

```
<section id="finances">
```

```
<h2>Фінанси</h2>
```

```
<p>Графік доходів, витрат, чистий прибуток</p>
```

```
<button onclick="alert('Звітність в розробці!')">Додати запис</button>
```

```
</section>
```

```
<section id="portfolio">
```

```
<h2>Портфоліо</h2>
```

```
<div class="gallery-grid">
```

```

```

```

```

```

```

```

```

```
</div>
```

```
</section>
```

```
<section id="register">
```

```
<h2>Реєстрація</h2>
```

```
<form onsubmit="handleRegister(event)">
```

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

```
<label for="regUsername">Ім'я користувача</label>
```

```
<input type="text" id="regUsername" required>
```

```
</div>
```

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

```
<label for="regEmail">Email</label>
```

```
<input type="email" id="regEmail" required>
```

```
</div>
```

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

```
<label for="regPassword">Пароль</label>
```

```
<input type="password" id="regPassword" required>
```

```
</div>
```

```
<button type="submit">Зареєструватись</button>
```

```
</form>
```

```
</section>
```

```
<section id="login">
```

```
  <h2>Вхід</h2>
```

```
  <form onsubmit="handleLogin(event)">
```

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

```
      <label for="loginEmail">Email</label>
```

```
      <input type="email" id="loginEmail" required>
```

```
    </div>
```

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

```
      <label for="loginPassword">Пароль</label>
```

```
      <input type="password" id="loginPassword" required>
```

```
    </div>
```

```
    <button type="submit">Увійти</button>
```

```
  </form>
```

```
</section>
```

```
<footer>
```

```
  &copy; 2025 Фотограф Іван Іванов. Всі права захищені.
```

```
</footer>
```

```
<script>
```

```
function showSection(id) {
```

```
  document.querySelectorAll('section').forEach(sec => sec.style.display = 'none');
```

```
  document.getElementById(id).style.display = 'block';
```

```
}
```

```
function handleRegister(e) {
```

```
  e.preventDefault();
```

```
  const user = {
```

```
    username: document.getElementById('regUsername').value,
```

```
    email: document.getElementById('regEmail').value,
```

```
password: document.getElementById('regPassword').value
};
localStorage.setItem('photographerUser', JSON.stringify(user));
alert('Реєстрація успішна! Тепер увійдіть.');
```

```
showSection('login');
}

function handleLogin(e) {
  e.preventDefault();
  const email = document.getElementById('loginEmail').value;
  const password = document.getElementById('loginPassword').value;
  const user = JSON.parse(localStorage.getItem('photographerUser'));

  if (user && user.email === email && user.password === password) {
    document.getElementById('mainNav').style.display = 'flex';
    document.getElementById('profileName').textContent = user.username;
    document.getElementById('profileEmail').textContent = user.email;
    showSection('profile');
  } else {
    alert('Невірний email або пароль.');
```

```
  }
}

function logout() {
  document.getElementById('mainNav').style.display = 'none';
  showSection('login');
}

function addSession() {
  const date = document.getElementById('sessionDate').value;
  const desc = document.getElementById('sessionDesc').value;
  if (date && desc) {
```

```

const text = `${date}: ${desc}`;
const li = document.createElement('li');
li.textContent = text;
document.getElementById('calendarList').appendChild(li);
}
}

document.getElementById('serviceForm').addEventListener('submit', function(e) {
  e.preventDefault();
  const name = document.getElementById('serviceName').value;
  const price = document.getElementById('servicePrice').value;
  const text = `${name} – ${price} грн`;
  const li = document.createElement('li');
  li.textContent = text;
  document.getElementById('servicesList').appendChild(li);
  e.target.reset();
});

document.getElementById('clientForm').addEventListener('submit', function(e) {
  e.preventDefault();
  const name = document.getElementById('clientName').value;
  const email = document.getElementById('clientEmail').value;
  const row = document.createElement('tr');
  row.innerHTML = `<td>${name}</td><td>${email}</td>`;
  document.getElementById('clientTable').appendChild(row);
  e.target.reset();
});

window.onload = () => showSection('register');
</script>
</body>
</html>
...

```