

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ	економіки та бізнес-освіти
Кафедра	Економіки та цифрового бізнесу
Спеціальність	122 «Комп'ютерні науки»
Форма навчання	Денна

КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА

Колєснікова Богдана Олеговича

(прізвище, ім'я, по батькові здобувача)

на тему	<u>Розробка інформаційної системи для автоматизації роботи та обліку оренди техніки фотостудії</u>
	<i>(повна назва теми)</i>
за матеріалами	
	<i>(повна назва бази дослідження)</i>

науковий керівник	<u>к.е.н., доцент</u>	<u>Соловйова В.В.</u>
	<i>(наук. ступінь, вчене звання)</i>	<i>(підпис) (прізвище, ініціали)</i>

Робота допущена до захисту в ЕК

Протокол засідання кафедри

від 12 червня 2026 р. № 15

Завідувач кафедри

(підпис)

к.е.н., доцент
наук. ступінь, вчене звання

Радько В.М.
прізвище, ініціали

ЗАТВЕРДЖЕНО

Наказ Міністерства освіти і науки, молоді та спорту України
29 березня 2012 року № 384

Форма № Н-9.01

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕКОНОМІКИ ТА БІЗНЕС-ОСВІТИ
(повне найменування вищого навчального закладу)

Кафедра економіки та цифрового бізнесу
Освітній ступінь бакалавр
Спеціальність 122 Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ **В.М. Радько**

“30” березня 2026 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ ЗДОБУВАЧУ

Колеснікова Богдана Олеговича

1. Тема роботи Розробка інформаційної системи для автоматизації роботи та обліку оренди техніки фотостудії

науковий керівник роботи _____ к.е.н., доцент, Соловйова В.В.

затверджені наказом вищого навчального закладу від «23» березня 2026 р. № 193-ст (д/ф)
№ 194-ст (з/ф)

2. Строк подання здобувачем роботи 29.05.2026р.

3. Зміст кваліфікаційної бакалаврської роботи, об'єкт, предмет та мета дослідження:

Розділ 1 - Стан проблемної області та аналіз існуючих інформаційних систем для автоматизації бізнес-процесів фотостудій

Розділ 2 - Інформаційне та математичне забезпечення інформаційної системи обліку клієнтів та техніки фотостудії

Розділ 3 - Програмне та технічне забезпечення веб-додатка для автоматизації роботи фотостудії

Об'єкт дослідження – процес управління, обліку та надання в оренду техніки і обладнання фотостудії.

Предмет дослідження – методи, моделі та програмні засоби розробки веб-орієнтованих інформаційних систем для автоматизації бізнес-процесів прокату техніки.

Мета кваліфікаційної бакалаврської роботи – розробка інформаційної системи (веб-додатка) для автоматизації процесів бронювання, обліку та контролю амортизації (зносу) техніки фотостудії з використанням сучасних веб-технологій.

4. Дата видачі завдання 03.04.2026р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної бакалаврської роботи	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	до 17.04.2026р.	
2	Підготовка розділу 2	до 08.05.2026р.	
3	Підготовка розділу 3	до 25.05.2026р.	
4	Реєстрація завершеної кваліфікаційної роботи	до 29.05.2026р.	
5	Отримання відгуку від наукового керівника	04.06.2026р.	
6	Отримання зовнішньої рецензії	05.06.2026р.	
7	Попередній захист кваліфікаційної роботи на кафедрі	08.06.2026р.	
8	Перевірка кваліфікаційної роботи на плагіат	12.06.2026р.	
9	Допуск кафедрою кваліфікаційної роботи до захисту	12.06.2026р.	
10	Підготовка студента до захисту в ЕК	до 19.06.2026р.	

Завдання підготував науковий керівник _____
(підпис)

Соловйова В.В
(прізвище та ініціали)

Завдання одержав здобувач _____
(підпис)

Колесніков Б.О
(прізвище та ініціали)

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ	6
ВСТУП	9
РОЗДІЛ 1. СТАН ПРОБЛЕМНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ ФОТОСТУДІЙ	12
1.1. Аналіз предметної області та бізнес-процесів оренди техніки у фотостудії	12
1.2. Огляд існуючих інформаційних систем автоматизації та обліку для фотобізнесу	15
1.3. Постановка задачі та обґрунтування вимог до розроблюваної інформаційної системи	19
Висновки до розділу 1	22
РОЗДІЛ 2. ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ КЛІЄНТІВ ТА ТЕХНІКИ ФОТОСТУДІЙ	23
2.1. Концептуальне проектування системи та побудова інформаційної моделі (UML-діаграми класів та прецедентів)	23
2.2. Алгоритмічне моделювання процесів бронювання та розрахунку вартості оренди	26
2.3. Проектування архітектури бази даних для обліку клієнтів та техніки (ER-діаграма)	29
Висновки до розділу 2	31
РОЗДІЛ 3. ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ВЕБ-ДОДАТКА ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ ФОТОСТУДІЙ	32
3.1. Обґрунтування вибору інструментарію та технологічного стека для реалізації інформаційної системи	32
3.2. Практична реалізація: розробка веб-інтерфейсу (Dashboard) для адміністратора та каталогу техніки	35
3.3. Опис функціоналу та тестування розробленого програмного забезпечення	39
Висновки до розділу 3	42
ВИСНОВКИ	43
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	45
ДОДАТКИ	47

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – база даних (сукупність взаємопов’язаних даних, що зберігаються в структурованому вигляді).

ІС – інформаційна система (сукупність програмних, технічних та організаційних засобів для збору, обробки, зберігання та передачі інформації).

ПЗ – програмне забезпечення (сукупність програм, процедур і документації, що забезпечують функціонування обчислювальної системи).

СУБД – система управління базами даних (програмний комплекс для створення, ведення та використання баз даних).

ТЗ – технічне завдання (документ, що визначає вимоги до розроблюваного програмного продукту).

AJAX – Asynchronous JavaScript and XML (технологія асинхронного обміну даними між браузером і сервером без перезавантаження сторінки).

API – Application Programming Interface (інтерфейс програмування застосунків, що визначає способи взаємодії між програмними компонентами).

CRUD – Create, Read, Update, Delete (базові операції над даними: створення, читання, оновлення та видалення).

CSS – Cascading Style Sheets (мова стилів, що використовується для опису зовнішнього вигляду веб-сторінок).

CSV – Comma-Separated Values (текстовий формат зберігання табличних даних, у якому значення розділені комами).

ER-діаграма – Entity-Relationship diagram (діаграма «сутність–зв’язок», що відображає структуру бази даних).

HTML – HyperText Markup Language (мова розмітки гіпертексту, що використовується для створення веб-сторінок).

HTTP – HyperText Transfer Protocol (протокол передачі гіпертексту, що забезпечує обмін даними в мережі Інтернет).

JSON – JavaScript Object Notation (текстовий формат обміну даними, заснований на синтаксисі JavaScript).

MVC – Model-View-Controller (архітектурний патерн, що розділяє застосунок на три взаємопов'язані компоненти: модель, уявлення та контролер).

RBAC – Role-Based Access Control (модель розмежування прав доступу на основі ролей користувачів).

REST – Representational State Transfer (архітектурний стиль для розробки розподілених систем, що базується на ресурсах та стандартних методах HTTP).

RESTful – архітектура, побудована за принципами REST (стиль проєктування веб-сервісів).

SQL – Structured Query Language (структурована мова запитів, призначена для роботи з реляційними базами даних).

UI – User Interface (користувацький інтерфейс, сукупність засобів взаємодії користувача з програмним забезпеченням).

UML – Unified Modeling Language (уніфікована мова моделювання, що використовується для візуалізації, проєктування та документування програмних систем).

UX – User Experience (користувацький досвід, сукупність вражень користувача від взаємодії з інтерфейсом).

WSGI – Web Server Gateway Interface (стандартний інтерфейс взаємодії між веб-сервером та веб-додатком на мові Python).

РЕФЕРАТ

Кваліфікаційна робота складається із: 46 с., 17 рис., 4 табл., 19 джерел.

Тема роботи: Розробка інформаційної системи для автоматизації роботи та обліку оренди техніки фотостудії.

Об'єкт дослідження - процес розробки інформаційної системи для автоматизації роботи та обліку оренди техніки фотостудії.

Предмет дослідження - розробка інформаційної системи для автоматизації роботи та обліку оренди техніки фотостудії.

Мета роботи - розробка клієнт-серверного веб-додатка для автоматизації процесів бронювання, обліку та контролю зносу техніки фотостудії.

У першому розділі проаналізовано предметну область та бізнес-процеси оренди техніки у фотостудії. Виявлено основні недоліки існуючої моделі організації процесу, зокрема відсутність автоматизації обліку, ризик виникнення овербукінгу та відсутність системного контролю за технічним станом обладнання. Здійснено огляд існуючих програмних рішень автоматизації оренди та проведено їх порівняльний аналіз. На основі отриманих результатів сформульовано функціональні та нефункціональні вимоги до розроблюваної інформаційної системи.

У другому розділі виконано концептуальне проектування інформаційної системи. Побудовано UML-діаграми прецедентів та класів, а також ER-діаграму реляційної бази даних. Розроблено алгоритми запобігання овербукінгу, розрахунку вартості оренди та контролю технічного зносу обладнання. Здійснено нормалізацію бази даних до третьої нормальної форми та спроектовано архітектуру системи з урахуванням вимог до надійності та безпеки.

У третьому розділі здійснено практичну реалізацію програмного забезпечення інформаційної системи. Розроблено веб-додаток на основі фреймворку Flask з використанням бази даних SQLite. Реалізовано клієнтську частину (вітрину для бронювання техніки) та адміністративну панель (дашборд з можливістю керування обладнанням і замовленнями). Забезпечено механізм асинхронної перевірки доступності техніки за допомогою AJAX-запитів,

генерацію договорів оренди та експорт звітів у форматі CSV. Проведено функціональне тестування системи.

Практичне значення роботи полягає у створенні готового до впровадження програмного продукту, який дозволяє автоматизувати процеси бронювання та обліку техніки фотостудії, зменшити вплив людського фактора та забезпечити формування необхідної звітної документації.

КЛЮЧОВІ СЛОВА: ІНФОРМАЦІЙНА СИСТЕМА, ВЕБ-ДОДАТОК, ОRENDA ФОТОТЕХНІКИ, ФОТОСТУДІЯ, FLASK, SQLITE, БАЗА ДАНИХ, АВТОМАТИЗАЦІЯ, БРОНЮВАННЯ.

ВСТУП

Актуальність теми. У сучасних умовах розвитку ринку послуг з організації фото- та відеозйомок діяльність фотостудій характеризується зростанням вимог до якості обслуговування клієнтів та ефективності використання матеріально-технічної бази. Професійне обладнання, що використовується у фотостудіях (дзеркальні та бездзеркальні камери, об'єктиви, системи освітлення, стабілізатори та інше), має високу вартість та обмежений технічний ресурс. У більшості невеликих і середніх фотостудій процеси обліку техніки, бронювання та контролю її технічного стану досі організовані за допомогою паперових журналів або електронних таблиць. Такий підхід призводить до низки проблем: високого ризику втрати або пошкодження облікових даних, виникнення овербукінгу (перехресного бронювання однієї одиниці техніки кількома клієнтами), відсутності системного контролю за амортизацією обладнання (зокрема, кількості спрацьовувань затвора фотоапаратів) та неможливості оперативного формування управлінської звітності. У зв'язку з цим актуальним завданням є розробка автоматизованої інформаційної системи, яка б дозволила оптимізувати процеси оренди техніки, підвищити точність обліку та зменшити вплив людського фактора на якість надання послуг.

Об'єкт дослідження - процес розробки інформаційної системи для автоматизації роботи та обліку оренди техніки фотостудії.

Предмет дослідження - розробка інформаційної системи для автоматизації роботи та обліку оренди техніки фотостудії.

Мета кваліфікаційної роботи — розробка клієнт-серверного веб-додатка для автоматизації процесів бронювання, обліку та контролю зносу техніки фотостудії з використанням сучасних веб-технологій.

Основні завдання кваліфікаційної роботи:

проаналізувати предметну область та існуючі бізнес-процеси оренди техніки у фотостудії;

здійснити огляд існуючих програмних рішень автоматизації оренди обладнання та обґрунтувати необхідність розробки спеціалізованої системи;
спроєктувати інформаційну модель та архітектуру бази даних системи;
розробити алгоритми захисту від овербукінгу та розрахунку вартості оренди;

реалізувати програмне забезпечення системи, включаючи серверну та клієнтську частини;

провести функціональне тестування розробленого програмного забезпечення та оцінити його відповідність поставленим вимогам.

Методи дослідження. У процесі виконання роботи застосовувалися методи системного аналізу для дослідження предметної області та бізнес-процесів, реляційне моделювання даних для проєктування бази даних, об'єктно-орієнтоване проєктування з використанням нотації UML для формалізації вимог та структури системи, а також методи функціонального тестування програмного забезпечення.

Наукова новизна роботи полягає в розробці спеціалізованої архітектури веб-орієнтованої інформаційної системи, адаптованої до потреб малого фотобізнесу, з інтеграцією математичної моделі контролю амортизації (зносу) обладнання безпосередньо в цикл бронювання та обліку техніки. На відміну від існуючих універсальних рішень, розроблювана система поєднує в собі функції бронювання, обліку зносу та формування документів в єдиному програмному продукті.

Практична значущість результатів роботи полягає у створенні готового до впровадження програмного продукту (MVP), здатного функціонувати на production-сервері Waitress. Розроблена система дозволяє зменшити ризики, пов'язані з людським фактором, автоматизувати процес документообігу (генерація договорів оренди, формування звітів у форматі CSV) та забезпечити прозорий облік технічного стану обладнання. Система може бути впроваджена у діяльність фотостудій без значних фінансових витрат на придбання та налаштування комерційного програмного забезпечення.

Структура роботи. Кваліфікаційна робота складається зі вступу, трьох розділів, висновків та списку використаних джерел. У першому розділі проведено аналіз предметної області та бізнес-процесів оренди техніки, здійснено огляд існуючих рішень та сформульовано вимоги до розроблюваної системи. У другому розділі висвітлено питання концептуального проєктування системи, включаючи побудову UML-діаграм та алгоритмічне моделювання основних процесів. Третій розділ присвячено практичній реалізації програмного забезпечення, опису архітектури та результатам тестування системи.

РОЗДІЛ 1

СТАН ПРОБЛЕМНОЇ ОБЛАСТІ ТА АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНИХ СИСТЕМ ДЛЯ АВТОМАТИЗАЦІЇ БІЗНЕС-ПРОЦЕСІВ ФОТОСТУДІЙ

1.1. Аналіз предметної області та бізнес-процесів оренди техніки у фотостудії

Сучасна фотостудія являє собою складний виробничий комплекс, у якому поєднуються творчі та технологічні процеси. Основним ресурсом такої студії є професійне обладнання, вартість якого може сягати десятків і сотень тисяч гривень. До такого обладнання належать дзеркальні та бездзеркальні камери, кінооб'єктиви, системи постійного та імпульсного освітлення, стабілізатори, записуючі пристрої та допоміжне приладдя. Висока вартість та технічна складність обладнання вимагають організації чіткого обліку, контролю його технічного стану та раціонального використання. Відсутність систематизованого підходу до управління парком техніки призводить до зростання експлуатаційних витрат, зниження якості надання послуг та втрати конкурентоспроможності студії на ринку.

У діяльності фотостудії процес оренди техніки є одним із ключових бізнес-процесів. Він безпосередньо впливає на завантаженість студії, рівень доходів та ступінь задоволеності клієнтів. У більшості невеликих і середніх фотостудій цей процес досі організований за традиційною моделлю, що базується на ручному обліку. Така модель характеризується низьким рівнем автоматизації та значною залежністю від людського фактора.

Існуючий бізнес-процес оренди техніки (модель AS-IS) розпочинається з моменту звернення клієнта до студії. Клієнт телефонує або пише в месенджер з проханням забронювати певну одиницю техніки на конкретні дати. Адміністратор студії відкриває паперовий журнал або таблицю в електронному вигляді та перевіряє наявність вільних дат. У разі відсутності перетинів з уже

оформленими бронюваннями адміністратор фіксує нове бронювання вручну, записуючи прізвище клієнта, контактні дані, назву техніки та період оренди. При видачі техніки клієнту складається акт приймання-передачі, в якому фіксується поточний стан обладнання. Після повернення техніки адміністратор візуально оглядає її на наявність пошкоджень, перевіряє комплектацію та вносить відповідні записи до журналу. У разі виявлення дефектів складається акт пошкодження.

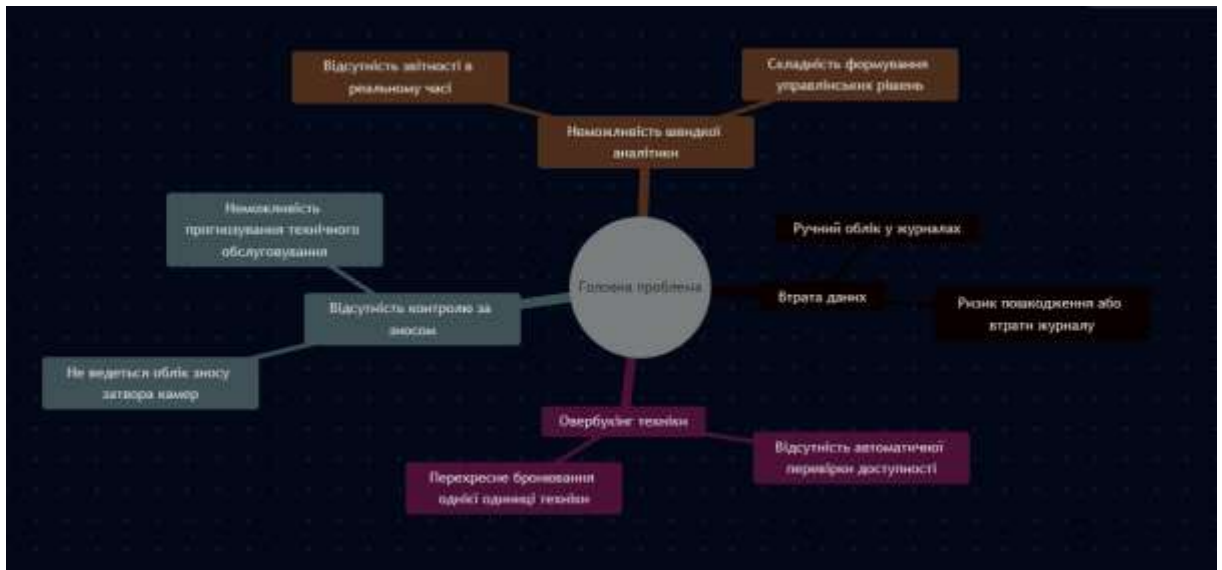


Рис 1.1. Модель бізнес-процесу оренди техніки «AS-IS» у нотації BPMN.

Існуюча модель організації процесу оренди техніки має низку суттєвих недоліків, які негативно впливають на ефективність роботи фотостудії. По-перше, ручний облік у паперових журналах або неструктурованих таблицях створює високий ризик втрати або пошкодження даних. У разі втрати журналу або пошкодження файлу вся історія бронювань може бути безповоротно втрачена. По-друге, відсутність автоматизованої перевірки доступності техніки призводить до виникнення овербукінгу — ситуації, коли одна одиниця техніки бронюється одночасно кількома клієнтами на перетинальні періоди. Такі випадки не лише створюють конфліктні ситуації з клієнтами, а й завдають фінансових збитків студії.

По-третє, у існуючій моделі практично відсутній системний контроль за технічним станом обладнання. Фототехніка, особливо камери, має обмежений ресурс роботи затвора. Після певної кількості спрацьовувань затвора зростає ймовірність його виходу з ладу. Однак у традиційній моделі облік таких показників, як кількість знімків або циклів роботи, ведеться нерегулярно або взагалі не ведеться. Це унеможливлює своєчасне планування технічного обслуговування та прогнозування потреби в заміні обладнання [4, с. 15]. По-четверте, ручний облік значно ускладнює отримання оперативної аналітичної інформації. Адміністратор студії не має можливості швидко сформувати звіт про завантаженість тієї чи іншої одиниці техніки, загальну виручку за певний період або структуру попиту клієнтів за категоріями обладнання.

Для візуалізації виявлених недоліків існуючої моделі організації процесу оренди техніки побудовано дерево проблем, яке наведено на рисунку 1.2.

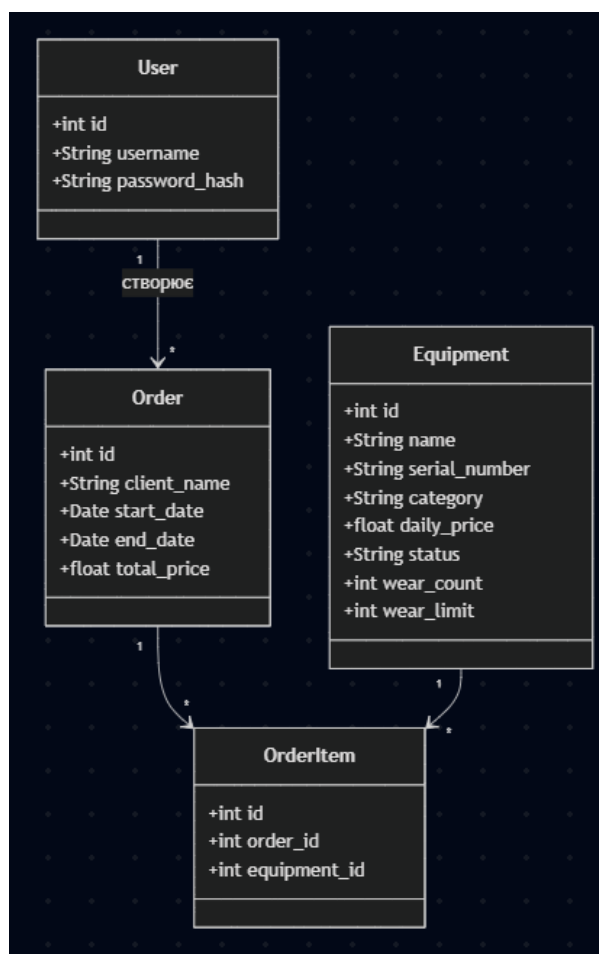


Рис 1.2. Дерево проблем існуючої моделі організації процесу оренди техніки у фотостудії

Проведений аналіз предметної області та бізнес-процесів оренди техніки у фотостудії дозволяє зробити висновок про необхідність переходу до моделі ТО-ВЕ, що передбачає впровадження автоматизованої інформаційної системи. Така система повинна забезпечити централізоване зберігання даних про обладнання та бронювання, автоматизовану перевірку доступності техніки, облік показників технічного зносу, а також формування оперативної звітності. Впровадження інформаційної системи дозволить суттєво підвищити ефективність управління парком техніки, зменшити ризики виникнення конфліктних ситуацій з клієнтами та створити передумови для прийняття обґрунтованих управлінських рішень на основі достовірних даних.

1.2. Огляд існуючих інформаційних систем автоматизації та обліку для фотобізнесу

Ринок програмного забезпечення для автоматизації бізнес-процесів у сфері оренди обладнання та надання послуг активно розвивається. Існує значна кількість як універсальних CRM-систем, так і спеціалізованих рішень, призначених для компаній, що займаються орендою техніки та обладнання. Проте більшість наявних рішень орієнтовані переважно на великі компанії або мають універсальний характер, що не завжди відповідає специфіці роботи невеликих фотостудій. Згідно з дослідженнями в галузі проектування інформаційних систем, при виборі програмного забезпечення для малого бізнесу важливо враховувати не лише функціональність, а й вартість впровадження, зручність використання та відповідність специфічним бізнес-процесам підприємства [11, с. 55].

Одним із популярних рішень на міжнародному ринку є система Boobookable. Це хмарна платформа, призначена для автоматизації процесів оренди обладнання. Система дозволяє вести облік техніки, формувати бронювання, керувати клієнтською базою та формувати звіти. До переваг Boobookable можна

віднести її хмарну архітектуру, що забезпечує доступ до системи з будь-якого пристрою, наявність мобільного додатку та інтеграцію з платіжними системами. Однак для невеликої фотостудії дане рішення має низку суттєвих недоліків. По-перше, вартість підписки є досить високою і розрахована в іноземній валюті, що створює додаткове фінансове навантаження для вітчизняних підприємств. По-друге, інтерфейс системи є перевантаженим великою кількістю функцій, які не потрібні для фотостудії. По-третє, у Booqable відсутній специфічний функціонал для обліку технічного зносу фототехніки, зокрема фіксації кількості спрацьовувань затвора камер. Це є критичним недоліком, оскільки саме облік амортизації обладнання є важливою складовою ефективного управління парком техніки у фотостудії.

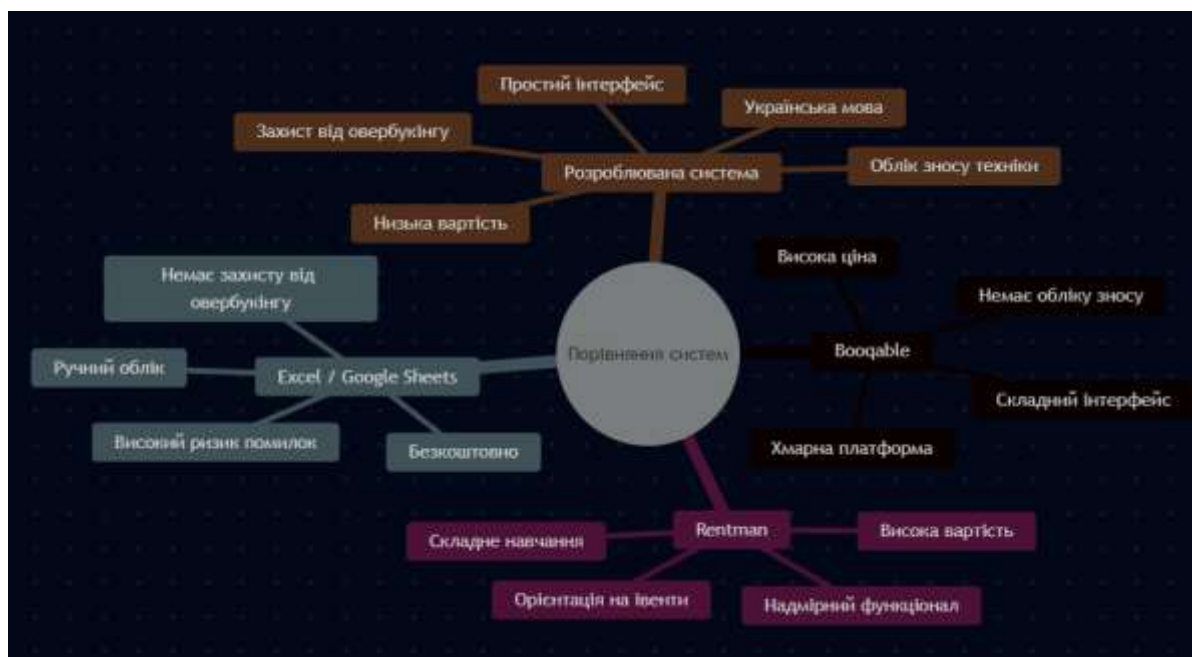


Рис 1.3. Інтерфейс панелі управління системи Booqable.

Ще одним поширеним рішенням є система Rentman. Дана платформа позиціонується як професійне рішення для компаній, що займаються орендою обладнання для івентів, зйомок та виробництва. Rentman має розвинуті модулі планування, управління ресурсами, формування комерційних пропозицій та інтеграції з бухгалтерськими системами. Система дозволяє вести детальний облік обладнання та формувати графіки завантаженості. Однак Rentman

орієнтований переважно на великі івент-агентства та компанії з великим штатом співробітників. Інтерфейс системи є складним і вимагає тривалого навчання персоналу. Крім того, вартість використання Rentman є високою, а функціонал надмірним для потреб локальної фотостудії. У системі також відсутній спеціалізований модуль обліку зносу фототехніки.

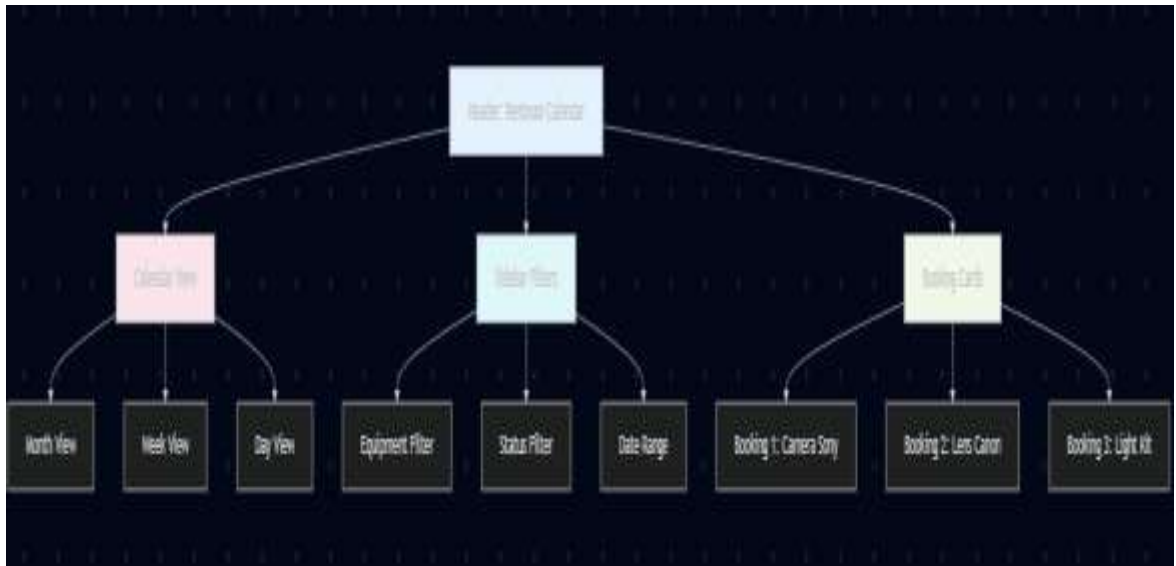


Рис 1.4. Інтерфейс календаря бронювань системи Rentman

Окремим типом рішень, які часто використовуються у фотостудіях, є традиційні табличні процесори, зокрема Microsoft Excel та Google Sheets. Багато невеликих студій ведуть облік техніки та бронювань саме в електронних таблицях. Такий підхід має низький поріг входу та не потребує додаткових фінансових витрат. Однак при збільшенні обсягів діяльності фотостудії використання таблиць призводить до серйозних проблем. Зростає ймовірність помилок при ручному внесенні даних, ускладнюється пошук інформації, а також виникають труднощі з одночасною роботою кількох співробітників. Крім того, електронні таблиці не забезпечують надійного захисту від овербукінгу та не дозволяють ефективно вести облік технічного зносу обладнання. При масштабуванні бізнесу така модель обліку часто призводить до колапсу управлінських процесів.

Для наочного порівняння існуючих рішень та розроблюваної системи було проведено аналіз за ключовими критеріями, що є важливими для фотостудії.

Проведений аналіз показує, що існуючі комерційні рішення, незважаючи на наявність широкого функціоналу, не повністю задовольняють потреби невеликої фотостудії через високу вартість, перевантаженість інтерфейсу та відсутність специфічного обліку технічного зносу фототехніки.

Таблиця 1.1

Порівняння існуючих рішень та розроблюваної системи за ключовими критеріями, важливими для фотостудії

Критерій порівняння	Booqable	Rentman	MS Excel / Google Sheets	Розроблювана система
Вартість використання	Висока (в USD)	Висока	Низька	Низька
Зручність інтерфейсу для малої студії	Середня	Низька	Висока (при малому обсязі)	Висока
Наявність контролю зносу техніки	Відсутня	Часткова	Відсутня	Присутня
Підтримка української мови	Відсутня	Відсутня	Присутня	Присутня
Захист від овербукінгу	Присутній	Присутній	Відсутній	Присутній
Можливість роботи без підключення до інтернету	Ні	Ні	Так	Частково
Складність впровадження	Середня	Висока	Низька	Низька
Можливість кастомізації під потреби фотостудії	Обмежена	Обмежена	Висока	Висока

Традиційні табличні рішення, у свою чергу, не забезпечують необхідного рівня автоматизації та надійності при зростанні обсягів роботи. Це обґрунтовує необхідність розробки спеціалізованої інформаційної системи, яка б поєднувала в собі простоту використання, низьку вартість впровадження та наявність функціоналу, адаптованого під специфіку оренди техніки у фотостудії.

1.3. Постановка задачі та обґрунтування вимог до розроблюваної інформаційної системи

Метою дипломної роботи є розробка інформаційної системи автоматизації роботи та обліку оренди техніки фотостудії, яка забезпечить підвищення ефективності бізнес-процесів, зменшення ризиків виникнення помилок та створення зручного інструменту для клієнтів і адміністративного персоналу.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести аналіз предметної області та існуючих бізнес-процесів оренди техніки у фотостудії;

- здійснити огляд існуючих рішень автоматизації та обґрунтувати необхідність розробки власної системи;

- спроєктувати архітектуру інформаційної системи, включаючи базу даних та функціональні модулі;

- розробити програмне забезпечення системи з використанням сучасних веб-технологій;

- реалізувати механізми захисту інформації та розмежування прав доступу;

- провести тестування розробленої системи та оцінити її відповідність поставленим вимогам.

На основі проведеного аналізу предметної області були сформульовані функціональні та нефункціональні вимоги до розроблюваної інформаційної системи. Функціональні вимоги системи поділяються на дві основні підсистеми: підсистему клієнта та підсистему адміністратора.

Вимоги до підсистеми клієнта:

- перегляд каталогу доступної техніки з можливістю фільтрації за категорією та ціною;

- перегляд детальної інформації про обрану одиницю техніки (назва, серійний номер, добова вартість, технічні характеристики);

формування бронювання з вибором техніки, дат оренди та зазначенням контактних даних;

автоматична перевірка доступності обраної техніки на вказані дати;

отримання підтвердження про успішне створення бронювання.

Вимоги до підсистеми адміністратора:

авторизація в системі з використанням захищеного механізму;

створення, перегляд, редагування та видалення записів про обладнання (CRUD-операції);

зміна статусу техніки («Доступно», «В оренді», «На технічному обслуговуванні»);

перегляд списку всіх бронювань з можливістю редагування та видалення;

формування та друк договору оренди техніки;

експорт даних про бронювання у форматі CSV;

перегляд статистичної інформації про завантаженість техніки та фінансові показники.

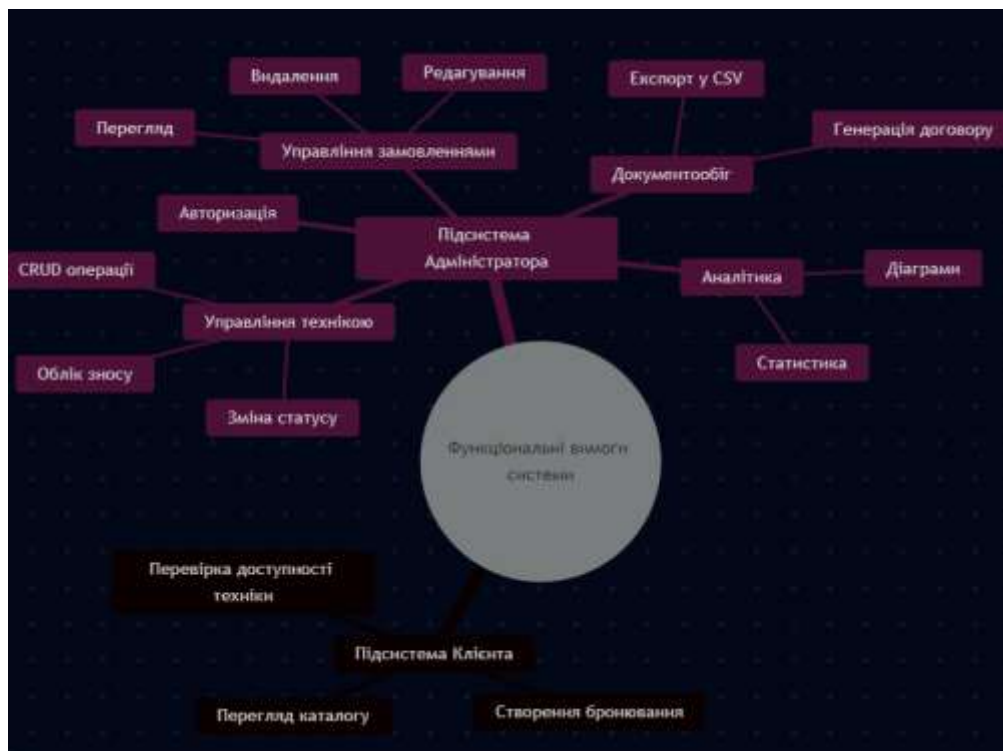


Рис 1.5. Ієрархічна структури функціональних вимог системи

Нефункціональні вимоги до системи сформульовані з урахуванням міжнародного стандарту якості програмного забезпечення ISO/IEC 25010 [17].

Система повинна забезпечувати прийнятний рівень продуктивності. Час відгуку сервера при обробці запитів не повинен перевищувати 2 секунд при стандартному навантаженні. Асинхронні запити (AJAX) для перевірки доступності техніки повинні виконуватися за час, що не перевищує 1 секунди.

Вимоги до надійності та безпеки передбачають використання механізму Role-Based Access Control (RBAC) для розмежування прав доступу між клієнтами та адміністратором. Усі паролі користувачів повинні зберігатися у хешованому вигляді з використанням криптографічно стійких алгоритмів. Система повинна забезпечувати захист від несанкціонованого доступу до адміністративної панелі.

Вимоги до ергономіки та юзабіліті передбачають наявність адаптивного дизайну, що забезпечує коректне відображення інтерфейсу на різних пристроях (персональні комп'ютери, планшети, смартфони). Інтерфейс системи повинен бути інтуїтивно зрозумілим та не вимагати тривалого навчання користувачів. Для реалізації адаптивного дизайну доцільно використовувати сучасні фронтенд-фреймворки.

Висновки до розділу 1

У першому розділі дипломної роботи було проведено аналіз предметної області та бізнес-процесів оренди техніки у фотостудії. Виявлено основні недоліки існуючої моделі організації процесу, зокрема відсутність автоматизації, ризик виникнення овербукінгу, відсутність системного контролю за технічним станом обладнання та неможливість оперативного отримання аналітичної інформації.

Проведений огляд існуючих програмних рішень показав, що комерційні системи автоматизації оренди (Booqable, Rentman) не повною мірою відповідають потребам невеликої фотостудії через високу вартість, перевантаженість функціоналом та відсутність специфічного обліку зносу

фототехніки. Використання традиційних табличних рішень також є неефективним при зростанні обсягів діяльності.

На основі проведеного аналізу було сформульовано мету та завдання дипломної роботи, а також визначено функціональні та нефункціональні вимоги до розроблюваної інформаційної системи. Отримані результати дозволяють перейти до етапу концептуального проєктування системи, який буде висвітлено у другому розділі роботи.

РОЗДІЛ 2

ІНФОРМАЦІЙНЕ ТА МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ОБЛІКУ КЛІЄНТІВ ТА ТЕХНІКИ ФОТОСТУДІЇ

2.1. Концептуальне проєктування системи та побудова інформаційної моделі (UML-діаграми класів та прецедентів)

Аналіз існуючої системи обліку оренди техніки фотостудії виявив низку суттєвих недоліків. Процеси бронювання та контролю за станом обладнання здійснювалися переважно в паперових журналах або неструктурованих електронних таблицях. Такий підхід не забезпечував оперативного доступу до актуальної інформації, створював ризики втрати даних та унеможлиблював ефективний контроль технічного зносу обладнання, зокрема показників зносу затворів камер. Крім того, відсутність механізмів розмежування прав доступу не дозволяла забезпечити прозорість і контрольованість процесів видачі та повернення техніки.

З метою усунення виявлених недоліків було здійснено перехід від моделі AS-IS до моделі TO-BE шляхом розробки автоматизованої інформаційної системи. Нова система забезпечує централізоване зберігання даних, автоматизовану перевірку доступності техніки, облік її технічного стану та розмежування прав доступу між клієнтами та адміністративним персоналом. Принципи проєктування інформаційних систем, викладені в роботі [11], стали основою для визначення функціональних вимог та архітектурних рішень майбутньої системи.

Для формалізації вимог до системи та визначення її функціональних меж було застосовано об'єктно-орієнтований підхід з використанням уніфікованої мови моделювання UML. На першому етапі концептуального проєктування було розроблено діаграму прецедентів, яка відображає основні сценарії взаємодії користувачів із системою.



Рис 2.1. UML-діаграма прецедентів інформаційної системи автоматизації оренди техніки фотостудії

У системі виділено два основні актори: «Клієнт» та «Адміністратор». Актор «Клієнт» має можливість переглядати каталог доступної техніки та формувати бронювання. Прецедент «Створення бронювання» включає перевірку доступності обраної техніки на вказаний період. Актор «Адміністратор» після успішної авторизації отримує доступ до функцій керування обладнанням. До його основних прецедентів належать: зміна статусу техніки, відстеження показників зносу, формування договорів оренди та перегляд статистичних звітів. Побудова діаграми прецедентів здійснювалася відповідно до рекомендацій, викладених у джерелах [3, с. 128] та [8, с. 67].

На другому етапі концептуального проектування було розроблено діаграму класів, яка відображає статичну структуру системи та взаємозв'язки між її основними сутностями.

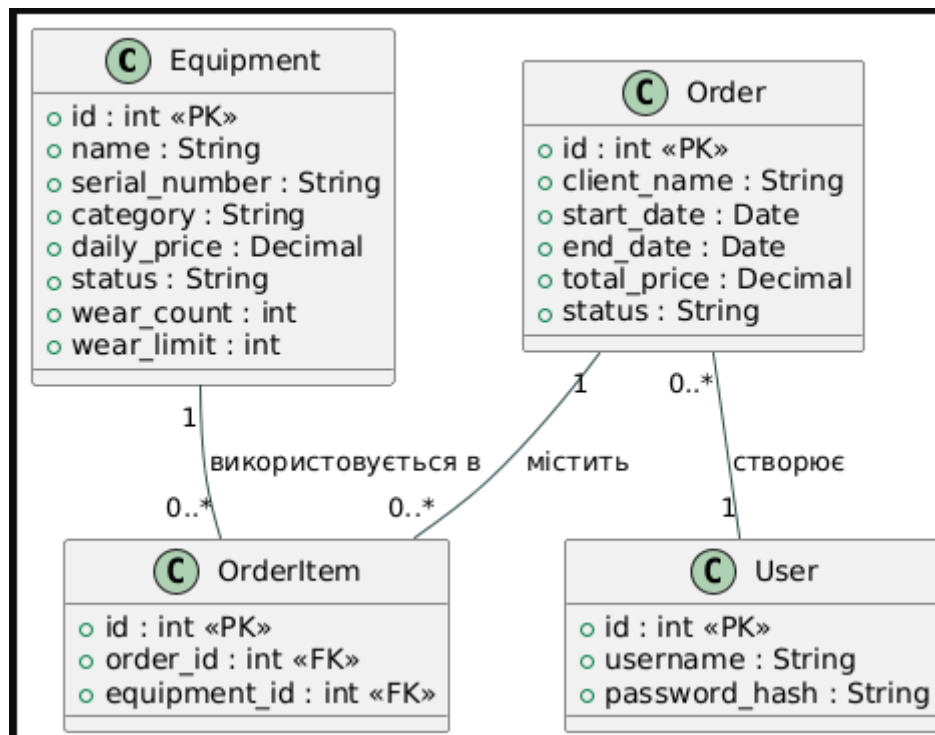


Рис 2.2. UML-діаграма класів інформаційної системи автоматизації оренди техніки фотостудії.

Центральною сутністю системи є клас Equipment, який описує одиницю техніки. Клас містить атрибути: id, name, serial_number, category, daily_price, status, wear_count та wear_limit. Атрибут status може набувати значень «available», «rented» та «maintenance». Атрибути wear_count та wear_limit забезпечують облік технічного зносу обладнання.

Клас Order відповідає за зберігання інформації про бронювання та містить атрибути: id, client_name, start_date, end_date та total_price. Для реалізації зв'язку «багато-до-багатьох» між замовленнями та обладнанням введено асоціативний клас OrderItem, який містить зовнішні ключі order_id та equipment_id.

Клас User описує облікові записи адміністраторів системи та містить атрибути: id, username та password_hash. Використання хешованого значення пароля забезпечує захист облікових даних. При побудові діаграми класів було враховано рекомендації щодо застосування шаблонів проєктування, викладені в роботі [14, с. 89], а також правила нотації UML, наведені в джерелі [15, с. 54] та [3, с. 215].

Побудовані діаграми прецедентів та класів забезпечують чітке визначення функціональних меж системи, основних акторів та структури даних. Отримані моделі стали основою для подальшого детального проєктування бази даних та реалізації програмного забезпечення інформаційної системи.

2.2. Алгоритмічне моделювання процесів бронювання та розрахунку вартості оренди

У розробленій інформаційній системі реалізовано низку алгоритмів, спрямованих на забезпечення коректності процесів бронювання, точності фінансових розрахунків та своєчасного контролю технічного стану обладнання.

Для виключення можливості одночасного бронювання однієї одиниці техніки різними клієнтами на перетинальні періоди в системі реалізовано алгоритм перевірки доступності. Нехай D_start та D_end — відповідно дата початку та дата закінчення нової заявки на бронювання. Для кожної існуючої активної броні тієї ж одиниці техніки відомі дати D_exist_start та D_exist_end . Бронювання вважається неможливим, якщо виконується умова перетину часових інтервалів:

$$(D_{start} \leq D_{exist_end}) \text{ AND } (D_{end} \geq D_{exist_start}) \quad (2.1)$$

Перевірка за формулою (2.1) виконується на сервері при кожному запиті на створення бронювання. Додатково реалізовано асинхронну перевірку доступності на клієнтській стороні. При зміні дат у формі бронювання система надсилає AJAX-запит до REST API (/api/check_dates). Такий підхід дозволяє попереджати користувача про неможливість бронювання ще до надсилання форми [12].

Алгоритм перевірки доступності складається з таких кроків:

Крок 1. Отримання параметрів нової заявки: дата початку, дата закінчення та ідентифікатор одиниці техніки.

Крок 2. Вибірка з бази даних усіх активних бронювань для вказаної техніки.

Крок 3. Послідовна перевірка умови (2.1) для кожного існуючого запису.

Крок 4. У разі виявлення перетину — формування повідомлення про impossibility бронювання та припинення процесу збереження замовлення.

Алгоритм розрахунку вартості оренди.

Вартість оренди обчислюється автоматично на основі добової вартості техніки та тривалості періоду оренди. Кількість днів оренди N визначається як різниця між датою закінчення та датою початку бронювання:

$$N = D_{end} - D_{start} \quad (2.2)$$

При цьому мінімальне значення N прирівнюється до 1. Підсумкова вартість оренди P_{total} розраховується за формулою:

$$P_{total} = P_{daily} \times N \quad (2.3)$$

де P_{daily} — добова вартість обраної одиниці техніки. Розрахунок виконується на рівні серверної бізнес-логіки безпосередньо перед збереженням замовлення в базу даних [16].

Алгоритм контролю технічного зносу обладнання.

Для забезпечення своєчасного технічного обслуговування в системі реалізовано механізм обліку зносу обладнання. Кожна одиниця техніки характеризується поточним показником зносу $W_{current}$ та граничним ресурсом W_{limit} .

Після повернення техніки з оренди або під час технічного обслуговування адміністратор вводить значення доданого зносу W_{add} . Новий показник зносу обчислюється за формулою:

$$W_{new} = W_{current} + W_{add} \quad (2.4)$$

У разі виконання умови

$$\frac{W_{new}}{W_{limit}} > 0.9 \quad (2.5)$$

система автоматично формує сигнал про необхідність проведення технічного обслуговування. Інформація про перевищення граничного зносу відображається в адміністративній панелі та може впливати на статус доступності техніки.

Алгоритм контролю зносу виконується при оновленні статусу обладнання адміністратором і складається з таких кроків:

Крок 1. Отримання поточного значення зносу $W_{current}$ та граничного ресурсу W_{limit} з бази даних.

Крок 2. Отримання значення доданого зносу W_{add} від адміністратора.

Крок 3. Обчислення нового значення зносу за формулою (2.4).

Крок 4. Перевірка умови (2.5).

Крок 5. У разі виконання умови — формування повідомлення про необхідність технічного обслуговування та оновлення статусу обладнання.

Реалізовані алгоритми базуються на фундаментальних підходах до аналізу інтервалів, числових розрахунків та порогового аналізу стану об'єктів [1; 7; 9]. Їх застосування забезпечує надійність процесів бронювання, точність фінансових розрахунків та своєчасне виявлення необхідності технічного обслуговування обладнання.

2.3. Проектування архітектури бази даних для обліку клієнтів та техніки (ER-діаграма)

При розробці інформаційної системи автоматизації оренди техніки фотостудії важливим етапом стало проектування бази даних, яка б забезпечувала надійне зберігання інформації, цілісність даних та ефективність виконання запитів. Вибір системи керування базами даних обумовлений специфікою експлуатації системи у невеликій фотостудії. Було обрано **SQLite** — вбудовану реляційну СУБД, яка не потребує встановлення окремого сервера та забезпечує повну підтримку транзакцій ACID. Такий підхід є оптимальним для

інформаційних систем малого та середнього бізнесу, оскільки значно спрощує процес розгортання та адміністрування [11, с. 156].

Процес проєктування бази даних здійснювався на основі реляційної моделі даних. Для забезпечення відсутності надмірності даних та аномалій оновлення було проведено нормалізацію відношень до третьої нормальної форми (3NF). Нормалізація до 3NF дозволила усунути транзитивні функціональні залежності між атрибутами та мінімізувати дублювання інформації [5, с. 244]. У результаті кожна таблиця бази даних містить лише дані, що безпосередньо стосуються її сутності, а зв'язки між таблицями реалізуються через зовнішні ключі.

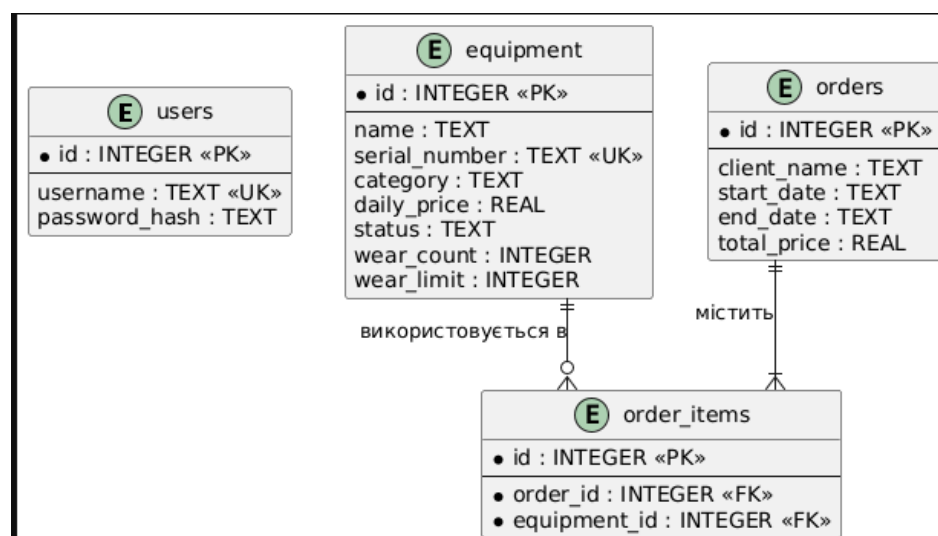


Рис 2.3. Фізична модель реляційної бази даних у вигляді ER-діаграми

Фізична модель реляційної бази даних у вигляді ER-діаграми зображено на рисунку 2.3.

База даних системи містить чотири основні таблиці: users, equipment, orders та order_items.

Таблиця users призначена для зберігання облікових даних адміністраторів системи. Вона містить первинний ключ id, унікальне поле username та поле password_hash, в якому зберігається хешоване значення пароля. Використання хешування забезпечує захист облікових даних відповідно до вимог інформаційної безпеки.

Таблиця `equipment` є центральною сутністю системи та зберігає інформацію про одиниці техніки. Вона містить первинний ключ `id`, унікальне поле `serial_number`, а також атрибути `name`, `category`, `daily_price`, `status`, `wear_count` та `wear_limit`. Поле `status` фіксує поточний стан техніки («available», «rented», «maintenance»), а поля `wear_count` та `wear_limit` забезпечують облік технічного зносу обладнання.

Таблиця `orders` призначена для зберігання інформації про бронювання. Вона містить первинний ключ `id`, поле `client_name`, а також атрибути `start_date`, `end_date` та `total_price`, які фіксують період оренди та підсумкову вартість.

Для реалізації зв'язку «багато-до-багатьох» між таблицями `orders` та `equipment` була створена таблиця-посередник `order_items`. Ця таблиця містить первинний ключ `id`, а також два зовнішні ключі: `order_id` та `equipment_id`. Використання таблиці `order_items` дозволяє одному замовленню містити декілька одиниць техніки, а одній одиниці техніки брати участь у кількох замовленнях. На рівні бази даних забезпечується реляційна цілісність шляхом встановлення зовнішніх ключів з каскадним видаленням, що гарантує автоматичне видалення пов'язаних записів при видаленні замовлення [13, с. 89].

Таким чином, спроектована архітектура бази даних відповідає вимогам третьої нормальної форми, забезпечує цілісність даних та ефективну підтримку бізнес-процесів оренди техніки. Використання зовнішніх ключів та механізмів реляційної цілісності дозволяє уникнути аномалій при вставці, оновленні та видаленні даних, що є важливою умовою надійної роботи інформаційної системи.

Висновки до розділу 2

У другому розділі було здійснено концептуальне проєктування інформаційної системи автоматизації роботи фотостудії та розроблено її інформаційне і математичне забезпечення. За допомогою уніфікованої мови моделювання (UML) побудовано діаграми прецедентів та класів. Це дозволило чітко формалізувати функціональні вимоги, визначити основні сутності системи

(обладнання, замовлення, деталі замовлень, користувачі) та розмежувати ролі і сценарії взаємодії для акторів «Клієнт» та «Адміністратор». Розроблено комплекс алгоритмів, що складають основу бізнес-логіки системи. Зокрема, математично формалізовано алгоритм запобігання овербукінгу шляхом перевірки перетину часових інтервалів оренди, реалізовано алгоритм автоматичного розрахунку підсумкової вартості замовлень, а також створено модель контролю технічного зносу обладнання, яка дозволяє автоматично відстежувати амортизацію та сигналізувати про необхідність технічного обслуговування. Здійснено проєктування архітектури реляційної бази даних. Обґрунтовано вибір СУБД SQLite як оптимального рішення для потреб малого бізнесу. Базу даних нормалізовано до третьої нормальної форми (3NF) та побудовано фізичну ER-діаграму, що гарантує реляційну цілісність даних, виключає їх надмірність та аномалії оновлення. Отримані в цьому розділі результати інформаційного та алгоритмічного моделювання утворюють надійний фундамент, необхідний для переходу до безпосередньої практичної реалізації програмного забезпечення та створення веб-додатка, що розглядається в наступному розділі роботи.

РОЗДІЛ 3

ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ВЕБ-ДОДАТКА ДЛЯ АВТОМАТИЗАЦІЇ РОБОТИ ФОТОСТУДІЇ

3.1. Обґрунтування вибору інструментарію та технологічного стека для реалізації інформаційної системи

При розробці інформаційної системи автоматизації роботи та обліку оренди техніки фотостудії одним із найважливіших етапів стало обґрунтування вибору технологічного стека. Правильний вибір інструментарію безпосередньо впливає на швидкість розробки, зручність подальшої підтримки системи, її масштабованість та відповідність вимогам малого бізнесу. У процесі аналізу було розглянуто кілька варіантів реалізації, кожен з яких оцінювався за критеріями продуктивності, простоти освоєння, наявності готових рішень та відповідності специфіці проєкту.

Аналіз наявних рішень показав, що для створення веб-системи, орієнтованої на невелику фотостудію, доцільно використовувати мову програмування Python. Порівняно з PHP, Python має значно вищу читабельність коду завдяки чіткій структурі та використанню відступів замість фігурних дужок. Це зменшує ймовірність помилок та спрощує підтримку системи в майбутньому. На відміну від Node.js, який також активно використовується для веб-розробки, Python забезпечує більш зрілу та стабільну екосистему для роботи з базами даних та обробки бізнес-логіки. Крім того, Python характеризується високою швидкістю розробки, що є важливим фактором при реалізації проєктів з обмеженим бюджетом та часом [10, с. 78].

Після вибору мови програмування було проведено порівняльний аналіз веб-фреймворків. Для реалізації системи було обрано мікрофреймворк Flask. На відміну від повноцінного фреймворку Django, Flask не нав'язує жорсткої структури проєкту та не містить надмірної кількості вбудованого функціоналу,

який не потрібен для даної системи. Це дозволяє розробнику самостійно обирати необхідні компоненти та формувати архітектуру відповідно до конкретних вимог. Flask забезпечує гнучкість у проектуванні маршрутів, обробці запитів та інтеграції з базами даних, що є важливою перевагою при створенні інформаційних систем середнього рівня складності [16, с. 45].

Таблиця 3.1

Порівняння веб-фреймворків для реалізації системи

Критерій порівняння	Flask	Django	Node.js (Express)
Швидкість розробки	Висока	Середня	Висока
Гнучкість архітектури	Висока	Низька (жорстка структура)	Висока
Обсяг вбудованого функціоналу	Низький	Високий	Низький
Підтримка баз даних	Гнучка (через ORM або SQLAlchemy)	Вбудований ORM (Django ORM)	Потребує додаткових бібліотек
Крива навчання	Низька	Середня	Середня
Підходить для малого бізнесу	Так	Частково	Так

Вибір Flask також обумовлений можливістю чіткої реалізації архітектурного патерну Model-View-Controller (MVC). У розробленій системі роль моделі (Model) виконує база даних SQLite разом із відповідними класами та функціями для роботи з даними. Роль представлення (View) реалізовано за допомогою шаблонізатора Jinja2, який дозволяє розділити логіку відображення даних та бізнес-логіку. Контролер (Controller) представлений маршрутами (routes) та функціями-обробниками запитів у Flask. Такий розподіл відповідальності відповідає принципам чистої архітектури, згідно з якими компоненти системи повинні мати чіткі межі відповідальності та бути незалежними один від одного [14, с. 112; 10, с. 95].

Окремим важливим аспектом стало обґрунтування вибору веб-сервера для запуску системи у виробничому середовищі. Вбудований тестовий сервер Flask

призначений виключно для розробки та не рекомендується для використання в реальних умовах експлуатації через недостатній рівень продуктивності та безпеки. Для запуску системи у продакшені було обрано WSGI-сервер Waitress. Даний сервер забезпечує стабільну обробку HTTP-запитів, підтримує багатопотоковість та має вищий рівень захищеності порівняно з тестовим сервером. Використання спеціалізованого WSGI-сервера є обов'язковою умовою при розгортанні Flask-додатків у реальному середовищі, оскільки це дозволяє уникнути проблем з продуктивністю та безпекою під час обробки одночасних запитів від користувачів [2, с. 267].

Фронтенд-частина системи реалізована з використанням стандартних веб-технологій. Для створення інтерфейсу застосовано мову розмітки HTML5 та каскадні таблиці стилів CSS3. Для забезпечення адаптивності інтерфейсу та прискорення розробки було використано фреймворк Bootstrap 5, який дозволяє швидко створювати сучасні та зручні у використанні інтерфейси. Взаємодія між клієнтською та серверною частинами здійснюється за допомогою нативного JavaScript та Fetch API. Такий підхід не потребує підключення додаткових бібліотек та забезпечує ефективну асинхронну взаємодію з REST API системи [12].

Таким чином, обраний технологічний стек, що включає мову Python, фреймворк Flask, архітектурний патерн MVC, WSGI-сервер Waitress та сучасні фронтенд-технології, є оптимальним рішенням для реалізації інформаційної системи автоматизації оренди техніки фотостудії. Даний стек забезпечує баланс між простотою розробки, надійністю та відповідністю вимогам малого бізнесу.

3.2. Практична реалізація: розробка веб-інтерфейсу (Dashboard) для адміністратора та каталогу техніки

Практична реалізація інформаційної системи виконана з використанням фреймворку Flask та шаблонізатора Jinja2. Архітектура системи побудована за

принципом розділення відповідальності, що відповідає принципам чистої архітектури [10, с. 112]. Усі маршрути системи розділені на клієнтську та адміністративну частини з чітким розмежуванням прав доступу.

Реалізація Role-Based Access Control (RBAC)

Для забезпечення контролю доступу до адміністративної панелі в системі реалізовано механізм Role-Based Access Control. Захист маршрутів адміністратора здійснюється за допомогою декоратора `@login_required`. Декоратор перевіряє наявність ключа 'logged_in' у сесії користувача. У разі відсутності авторизації відбувається перенаправлення на сторінку входу.

Нижче наведено реалізацію декоратора:

```
from functools import wraps
from flask import session, redirect, url_for

def login_required(f):
    @wraps(f)
    def decorated_function(*args, **kwargs):
        if 'logged_in' not in session:
            return redirect(url_for('login'))
        return f(*args, **kwargs)
    return decorated_function
```

Рис 3.1. Реалізація декоратора `login_required` для захисту маршрутів веб-додатку

Декоратор використовує механізм сесій Flask (`session`), який зберігає дані користувача на сервері. Після успішної авторизації в сесію записується значення `session['logged_in'] = True`. Це дозволяє контролювати доступ до захищених маршрутів без необхідності передачі токенів у кожному запиті [16, с. 78].

Механізм хешування паролів

З метою забезпечення безпеки облікових даних у системі реалізовано хешування паролів. Зберігання паролів у відкритому вигляді є неприпустимим, оскільки у разі компрометації бази даних зловмисник отримає доступ до всіх

облікових записів. Для хешування використано функції бібліотеки `werkzeug.security`:

```
from werkzeug.security import generate_password_hash, check_password_hash

hashed_pw = generate_password_hash('admin123')
# Перевірка пароля
if check_password_hash(user['password_hash'], entered_password):
    # авторизація успішна
```

Рис 3.2. Використання функцій `generate_password_hash` та `check_password_hash` для хешування та перевірки паролів

Функція `generate_password_hash()` автоматично застосовує криптографічно стійкий алгоритм хешування з додаванням солі. Функція `check_password_hash()` порівнює введений пароль з хешем, що зберігається в базі даних. Такий підхід значно підвищує рівень захищеності системи [16, с. 92].

Інтерфейс клієнтської частини (Вітрина)

Клієнтська частина системи реалізована у вигляді веб-вітрини. Для відображення каталогу техніки використовується шаблонізатор Jinja2. У шаблоні реалізовано цикл, який виводить картки доступного обладнання. При переході з картки техніки на сторінку бронювання через GET-параметр `eq_id` відбувається автоматичне заповнення відповідного поля у формі.

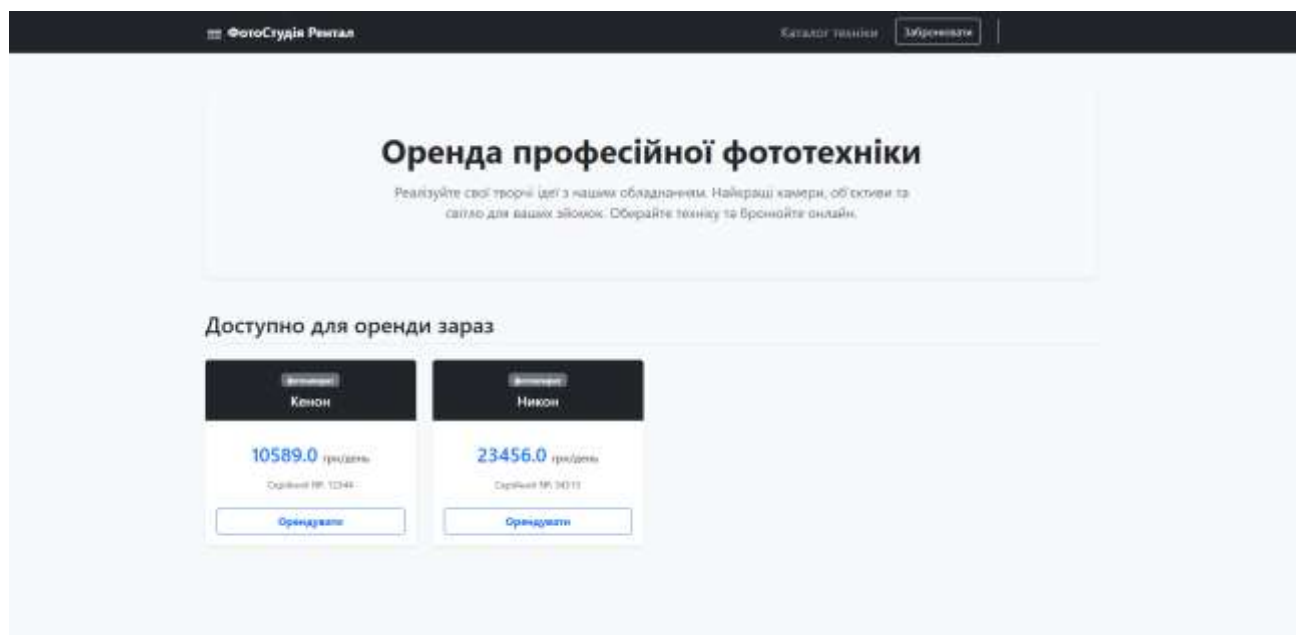


Рис 3.3. Головна сторінка каталогу техніки для клієнтів.

Така реалізація покращує користувацький досвід, оскільки клієнту не потрібно повторно обирати техніку у випадяючому списку.

Інтерфейс адміністративної панелі (Dashboard)

Адміністративна панель містить дашборд зі зведеною статистикою. Дані для дашборду отримуються з бази даних за допомогою агрегатних функцій SQLite:

```
stats = {
    'total_equipment': conn.execute('SELECT COUNT(*) FROM equipment').fetchone()[0],
    'available_equipment': conn.execute("SELECT COUNT(*) FROM equipment WHERE status = 'ava").fetchone()[0],
    'total_revenue': conn.execute('SELECT SUM(total_price) FROM orders').fetchone()[0] or 0
}
```

Рис. 3.4. Формування об'єкта статистичних даних для адміністративної панелі

Для візуалізації розподілу техніки за категоріями використовується бібліотека Chart.js. Дані з Python (списки) конвертуються у формат JSON і передаються у шаблон:

```
stats['chart_labels'] = [row['category'] for row in categories]
stats['chart_data'] = [row['count'] for row in categories]
```

Рис. 3.5. Підготовка даних для побудови діаграми розподілу техніки за категоріями

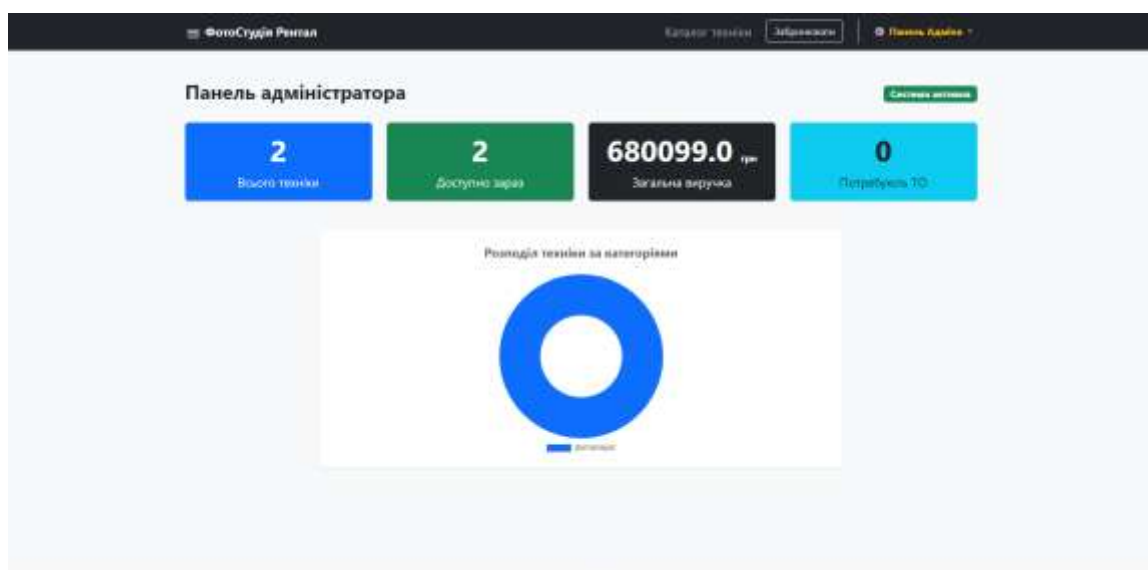


Рис 3.6. Дашборд адміністратора зі статистикою та діаграмою.

Така реалізація дозволяє адміністратору швидко оцінювати поточний стан парку техніки та приймати управлінські рішення.

Асинхронна перевірка доступності техніки

Для підвищення зручності роботи з формою бронювання реалізовано асинхронну перевірку доступності техніки на вказані дати. Перевірка виконується за допомогою JavaScript та Fetch API без перезавантаження сторінки:

```
fetch('/api/check_dates', {
  method: 'POST',
  headers: {'Content-Type': 'application/json'},
  body: JSON.stringify({
    equipment_id: equipmentId,
    start_date: startDate,
    end_date: endDate
  })
})
.then(response => response.json())
.then(data => {
  if (!data.available) {
    // показати повідомлення про зайнятість
  }
});
```

Рис. 3.7. AJAX-запит на перевірку доступності техніки у заданому періоді

Даний механізм реалізує взаємодію за принципом REST API та відповідає сучасним підходам до побудови клієнт-серверних веб-додатків [12, с. 134]. На рисунку 3.3 зображено процес асинхронної перевірки доступності техніки.

Реалізований інтерфейс забезпечує зручну роботу як для клієнтів, так і для адміністратора системи. Використання сучасних веб-технологій та чітке розділення логіки дозволило створити функціональну та підтримувану інформаційну систему.

3.3. Опис функціоналу та тестування розробленого програмного забезпечення

Розроблена інформаційна система забезпечує не лише базові функції бронювання та обліку техніки, а й додаткові можливості формування документів та експорту даних. Функціонал системи реалізовано з урахуванням вимог до якості програмного забезпечення, викладених у міжнародних стандартах [17; 18].

Реалізація генерації договорів оренди

Однією з важливих функцій адміністративної панелі є можливість формування договору оренди техніки. Для генерації документа обрано підхід, заснований на використанні HTML-шаблону з подальшим друком через браузер. У шаблоні contract.html застосовано CSS-правила @media print, які дозволяють адаптувати відображення сторінки спеціально для друку на аркуші формату А4.

Такий підхід має низку переваг порівняно з генерацією PDF-файлу на стороні сервера. По-перше, він не створює додаткового навантаження на сервер, оскільки формування документа відбувається на клієнтській стороні. По-друге, забезпечується висока гнучкість оформлення документа без необхідності використання додаткових бібліотек для генерації PDF. При натисканні кнопки «Друкувати» браузер автоматично застосовує стилі друку та відкриває стандартне діалогове вікно друку операційної системи.

Система передбачає можливість експорту інформації про всі бронювання у форматі CSV для подальшої обробки в табличних процесорах. Функціонал експорту реалізовано на рівні серверної частини. При зверненні до маршруту /export/orders виконується складний SQL-запит з використанням конструкцій JOIN та GROUP_CONCAT для отримання інформації про замовлення та переліку заброньованої техніки в одному рядку.

Ф. Друкать / Вернуть PDF
Печатать

ДОГОВІР ОРЕНДИ ОБЛАДНАННЯ №2

Фотостудія "Рентал" | Дата формування: 10.06.2025

1. Сторони

Орендодавець: Фотостудія "Рентал"

Орендар: Богдан Колесніков

2. Предмет оренди

Назва техніки	Серійний номер	Період оренди	Вартість
Ніксон	34575	2025-06-10 — 2025-06-25	351840.0 грн

3. Умови

3.1. Орендар зобов'язується повернути техніку в справному стані та у визначений термін.

3.2. У разі пошкодження обладнання Орендар відповідно несе відповідальність за ремонт.

Від Орендодавця

Від Орендаря

Рис 3.8. Зображено сформований договір оренди техніки, готовий до друку.

Експорт даних у форматі CSV

Отримані дані записуються в об'єкт `io.StringIO`, після чого формується HTTP-відповідь з відповідними заголовками. Для коректного відображення кирилиці у програмі Microsoft Excel використано кодування `utf-8-sig` (BOM). Це дозволяє уникнути проблем з відображенням українських символів при відкритті файлу.

ID Замовл	Клієнт	Техніка	Початок	Кінець	Сума (грн)
1	Богдан Кс	Кенон	11.06.2026	12.07.2026	328259.0
2	Богдан Кс	Ніксон	10.06.2026	25.06.2026	351840.0

Рис 3.9. Приклад сформованого CSV-звіту, відкритого в Microsoft Excel.

Тестування розробленого програмного забезпечення

Тестування інформаційної системи проводилося відповідно до вимог стандартів якості програмного забезпечення [17; 18] та методичних рекомендацій ДУЕТ [19]. Основним методом тестування обрано Black-box testing (тестування

«чорної скриньки»)), за якого перевіряється відповідність поведінки системи заданим вимогам без аналізу її внутрішньої структури коду.

Тестування проводилося за попередньо розробленими сценаріями (Test Cases), які охоплюють основні функціональні модулі системи. У таблиці 3.2 наведено ключові сценарії тестування.

Таблиця 3.2

Сценарії тестування модулів системи

№	Назва тесту	Вхідні дані	Очікуваний результат	Фактичний результат	Статус
1	Успішна авторизація адміністратора	Логін: admin, Пароль: admin123	Перехід на сторінку дашборду	Перехід на сторінку дашборду	Пройдено
2	Створення бронювання з коректними датами	Вибір техніки, коректні дати, ім'я клієнта	Бронювання успішно створено, техніка недоступна для повторного бронювання	Бронювання створено	Пройдено
3	Спроба бронювання на зайнятій даті	Вибір техніки, дати, що перетинаються з існуючим бронюванням	Блокування кнопки бронювання + повідомлення про зайнятість	Повідомлення про неможливість бронювання	Пройдено
4	Спроба доступу до адмін-панелі без авторизації	Перехід за адресою /admin без авторизації	Перенаправлення на сторінку входу /login	Перенаправлення на /login	Пройдено
5	Експорт звіту у форматі CSV	Натиснення кнопки «Завантажити звіт»	Завантаження файлу .csv з коректним кодуванням	Файл успішно завантажено	Пройдено

Проведене тестування підтвердило відповідність системи функціональним вимогам. Усі позитивні сценарії виконуються коректно, а негативні сценарії обробляються системою з формуванням відповідних повідомлень користувачу. Особливу увагу було приділено перевірці механізмів захисту від овербукінгу та контролю доступу, які є критичними для надійної роботи системи.

Висновки до розділу 3

У третьому розділі роботи було обґрунтовано вибір технологічного стека та описано практичну реалізацію інформаційної системи автоматизації оренди техніки фотостудії. Використання фреймворку Flask у поєднанні з архітектурним патерном MVC дозволило створити гнучку та підтримувану систему. Реалізація механізмів авторизації, хешування паролів та розмежування прав доступу забезпечила необхідний рівень інформаційної безпеки.

Особливу увагу приділено розробці додаткового функціоналу, такого як генерація договорів оренди та експорт даних у форматі CSV. Застосування сучасних веб-технологій (Jinja2, Fetch API, Chart.js) забезпечило створення зручного та функціонального інтерфейсу як для клієнтів, так і для адміністратора системи.

Проведене функціональне тестування системи за методом Black-box testing підтвердило відповідність розробленого програмного забезпечення поставленим вимогам. Система успішно функціонує на WSGI-сервері Waitress у виробничому середовищі та готова до впровадження в діяльність фотостудії.

ВИСНОВКИ

У кваліфікаційній роботі бакалавра вирішено актуальне науково-практичне завдання — розроблено та реалізовано інформаційну систему (веб-додаток) для автоматизації процесів бронювання, обліку та контролю амортизації техніки фотостудії. У результаті виконання досліджень та практичної розробки поставлену мету досягнуто в повному обсязі, а всі завдання успішно виконано.

За підсумками проведеної роботи можна зробити наступні висновки:

Проаналізовано предметну область та специфіку бізнес-процесів. Встановлено, що використання традиційної моделі обліку оренди (паперові журнали, електронні таблиці) у фотостудіях призводить до високого ризику втрати даних, виникнення перехресних бронювань (овербукінгів) та відсутності системного контролю за технічним зносом обладнання.

Доведено необхідність розробки власного програмного рішення. Огляд наявних комерційних інформаційних систем (Booqable, Rentman) показав, що вони не повною мірою задовольняють потреби локальних фотостудій через надлишкову функціональність, високу вартість ліцензування та відсутність спеціалізованих модулів для обліку специфічної амортизації (наприклад, пробігу затворів камер).

Виконано концептуальне проєктування інформаційної системи. За допомогою уніфікованої мови моделювання (UML) розроблено діаграми прецедентів та класів, які чітко розмежували ролі «Клієнт» і «Адміністратор» та визначили статичну структуру сутностей.

Розроблено алгоритмічне та математичне забезпечення. Створено математичні моделі перевірки доступності техніки для запобігання овербукінгу, алгоритми автоматичного розрахунку вартості оренди та порогову модель контролю технічного зносу обладнання, яка автоматично сигналізує про необхідність технічного обслуговування.

Спроектовано реляційну базу даних. Обґрунтовано використання СУБД SQLite для малого бізнесу. Базу даних приведено до третьої нормальної форми (3NF), розроблено ER-діаграму, яка забезпечує реляційну цілісність даних та виключає аномалії оновлення завдяки використанню зовнішніх ключів.

Здійснено практичну реалізацію програмного забезпечення. Веб-додаток створено з використанням мови Python та мікрофреймворку Flask за архітектурним патерном MVC. Реалізовано асинхронну взаємодію (AJAX) для перевірки дат без перезавантаження сторінки. Для забезпечення інформаційної безпеки впроваджено модель Role-Based Access Control (RBAC) та криптографічне хешування паролів. Автоматизовано супутні бізнес-процеси. У систему інтегровано функціонал генерації договорів оренди, адаптованих для миттєвого друку через браузер, та модуль експорту статистичних даних у форматі CSV для подальшого бухгалтерського чи управлінського аналізу.

Проведено функціональне тестування. Тестування за методом «чорної скриньки» (Black-box testing) підтвердило стабільність роботи системи, коректну обробку позитивних і негативних сценаріїв (зокрема, блокування спроб забронювати зайняту техніку). Розгортання додатку на WSGI-сервері Waitress довело його готовність до експлуатації у виробничому (production) середовищі.

Практичне значення одержаних результатів полягає у створенні готового до впровадження програмного продукту (MVP), який дозволяє малим та середнім фотостудіям оптимізувати процес управління парком техніки. Впровадження розробленого веб-додатка мінімізує вплив людського фактора, підвищує якість обслуговування клієнтів та надає адміністративному персоналу зручний інструмент для ведення прозорого обліку й аналітики.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ахо А. В., Хопкрофт Дж. Е., Ульман Дж. Д. Структури даних і алгоритми : пер. з англ. / за ред. Ю. В. Капітонової. Київ : Наукова думка, 2018. 448 с.
2. Басс Л., Клементс П., Казман Р. Архітектура програмного забезпечення на практиці : пер. з англ. Київ : Діалектика, 2022. 624 с.
3. Буч Г., Рамбо Дж., Джекобсон А. Мова UML. Посібник користувача : пер. з англ. 2-ге вид. Москва : ДМК Пресс, 2019. 496 с.
4. Вовк В. М., Семенов С. Г. Моделі та методи прогнозування технічного стану складних технічних систем : монографія. Харків : НТУ «ХПІ», 2020. 248 с.
5. Дейт К. Дж. Вступ до систем баз даних : пер. з англ. 8-ме вид. Київ : Діалектика, 2020. 1328 с.
6. Кнут Д. Е. Мистецтво програмування. Том 1 : Основні алгоритми : пер. з англ. 3-тє вид. Москва : Вільямс, 2019. 720 с.
7. Кормен Т., Лейзерсон Ч., Рівест Р., Штайн К. Вступ до алгоритмів : пер. з англ. 3-тє вид. Київ : К.І.С., 2021. 1320 с.
8. Ларман К. Застосування UML 2.0 і шаблонів проєктування : пер. з англ. 3-тє вид. Москва : Вільямс, 2018. 736 с.
9. Левітін А. В. Алгоритми: вступ до розробки та аналізу : пер. з англ. 3-тє вид. Київ : Діалектика, 2020. 720 с.
10. Мартін Р. Чиста архітектура. Мистецтво розробки програмного забезпечення : пер. з англ. Київ : Фабула, 2020. 352 с.
11. Полатайко О. М., Гавриленко О. В. Проєктування інформаційних систем : навч. посіб. Київ : КПІ ім. Ігоря Сікорського, 2021. 312 с.

12. Річардсон Л., Амундсен М. RESTful Web API : пер. з англ. Київ : Діалектика, 2019. 448 с.
13. Сильбершатц А., Корт Г., Судершан С. Системи баз даних: концепції, проєктування та реалізація : пер. з англ. 7-ме вид. Київ : Діалектика, 2021. 1088 с.
14. Фаулер М. Шаблони корпоративних додатків : пер. з англ. Москва : Вільямс, 2018. 544 с.
15. Фаулер М. UML у стислому викладі : пер. з англ. 3-тє вид. Москва : Лорі, 2020. 192 с.
16. Грінберг М. Розробка веб-додатків з використанням Flask : пер. з англ. Київ : Діалектика, 2020. 312 с.
17. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models. Geneva : ISO, 2011. 34 p.
18. ДСТУ ISO/IEC 12207:2017. Інженерія систем і програмного забезпечення. Процеси життєвого циклу програмного забезпечення. Київ : ДП «УкрНДНЦ», 2017. 152 с.
19. Методичні рекомендації щодо виконання та захисту кваліфікаційних робіт здобувачів вищої освіти ступеня бакалавра за спеціальністю 122 «Комп'ютерні науки». Дніпро : ДУЕТ, 2023. 48 с.

ЗГОДА здобувача вищої освіти

Державного університету економіки і технологій
про перевірку кваліфікаційної роботи на прояви
академічного плагіату
та розміщення в Репозитарії Університету

Я, _____ **Колеснікова Богдана Олегович,**

підтримую політику Державного університету економіки і технологій
з академічної доброчесності і відкритого доступу.

Засвідчую, що кваліфікаційна бакалаврська робота

**«Розробка інформаційної системи для автоматизації роботи та обліку оренди
техніки фотостудії»**

виконана самостійно та не містить академічного плагіату. Я не надавав і не
одержував недозволену допомогу під час підготовки цієї роботи. Робота містить
результати власних досліджень. Використання ідей, результатів і текстів інших
авторів мають посилання на відповідне джерело.

Із чинним Положенням про запобігання та виявлення академічного
плагіату в роботах здобувачів вищої освіти Державного університету економіки і
технологій ознайомена. Чітко усвідомлюю, що в разі виявлення у
кваліфікаційній роботі порушення норм академічної доброчесності робота не
допускається до захисту або оцінюється незадовільно.

Також я поінформований, що відповідно до «Положення про Репозитарій
(електронну базу даних) Державного університету економіки і технологій»
зазначена робота буде розміщена в Електронному архіві Університету
(Репозитарії ДУЕТ). З умовами такого розміщення ознайомена.

15.06.26 р



Колесников Б.О.