

Вінницький національний технічний університет
(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації
(повне найменування інституту, факультету)

Кафедра комп'ютерних систем управління
(повна назва кафедри)

Бакалаврська кваліфікаційна робота

на тему:

«Автоматизована система управління орендою робочих місць у коворкінга»

Виконав: студент 4-го курсу,
групи _____

спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології
(шифр і назва спеціальності)



Вікторія БРАТКОВА
(ім'я та прізвище)

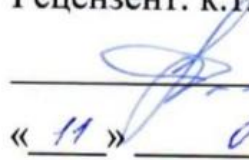
Керівник: д.т.н. проф. каф. КСУ



Марія ЮХИМЧУК
(ім'я та прізвище)

« 10 » 06 2025 р.

Рецензент: к.т.н доцент каф. АІТ



Володимир ГАРМАШ
(ім'я та прізвище)

« 11 » 06 2025 р.

Допущено до захисту
Завідувач кафедри КСУ
д.т.н., проф.

 В'ячеслав КОВТУН
(ім'я та прізвище)

« 12 » 06 2025 р.

Вінниця ВНТУ - 2025 рік

ВІННИЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління
Рівень вищої освіти I-й (бакалаврський)
Галузь знань – 15 Автоматизація та приладобудування
Спеціальність 151 Автоматизація та комп'ютерно-інтегровані технології
Освітньо-професійна програма Інтелектуальні комп'ютерні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

д.т.н., проф.

 В'ячеслав КОВТУН

«24» 05 2025 року

**ЗАВДАННЯ
НА БАКАЛАВРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Братковій Вікторії Юріївні

(прізвище, ім'я, по батькові)

1. Тема бакалаврської кваліфікаційної роботи «Автоматизована система управління орендою робочих місць у коворкінгах»

керівник бакалаврської кваліфікаційної роботи Юхимчук Марія Сергіївна,
професор технічних наук

затверджені наказом вищого навчального закладу від «20» 03 2025 року № 97

2. Строк подання студентом бакалаврської кваліфікаційної роботи 10.06.2025 року

3. Вихідні дані до бакалаврської кваліфікаційної роботи:

1. Костін В.В. Автоматизовані системи управління підприємствами: навч. посіб. – К.: НАУ, 2014. – 278 с.
2. Козлов В.Ю. Хмарні технології в управлінні підприємством. – К.: Ліра-К, 2019. – 312 с.
3. Spurlock J. Bootstrap: Responsive Web Development. – O'Reilly Media, 2013. – 140 p.
4. Zhang Y., Deng R.H. Security and Privacy in Smart Office Environments. – IEEE Internet of Things Journal, 2020. – Vol. 7(5). – P. 4202–4212.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) вступ, дослідження об'єкта автоматизації, дослідження та аналіз роботи систем управління послугами в сфері коворкінгів, дослідження функціональних можливостей сучасних програмних рішень, проектування автоматизованої систем управління орендою робочих місць у коворкінгах, розробка автоматизованої системи управління орендою робочих місць у коворкінгах

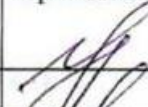
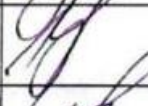
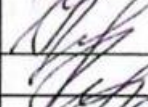
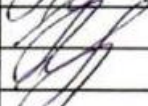
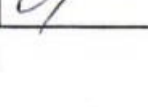
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) реляційна схема бази даних – 1 шт, UML-діаграми – 5 шт, графічні матеріали(вікна інтерфейсу) – 14 шт.

1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада Консультанта	Підпис, дата	
		завдання видав	виконання прийняв

2. Дата видачі завдання: «26» 05 2025 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів Роботи	Строк виконання етапів роботи	Примітка
1	Дослідження актуальності поставленої задачі	10.05.2025 р.	
2	Дослідження та аналіз роботи систем управління послугами в сфері коворкінгів	14.05.2025 р.	
3	Дослідження функціональних можливостей сучасних програмних рішень	19.05.2025 р.	
4	Проектування автоматизованої системи управління орендою робочих місць у коворкінгах	20.05.2025 р.	
5	Розробка автоматизованої системи управління орендою робочих місць у коворкінгах	21.05.2025 р.	
6	Апробація результатів дослідження	29.05.2025 р.	
7	Публікації		
8	Оформлення пояснювальної записки, графічного матеріалу і презентації	07.06.2025 р.	
9	Графічні матеріали	08.06.2025 р.	
10	Захист БКР	17.06.2025 р.	

Студент

(підпис)

Вікторія БРАТКОВА

Керівник роботи

(підпис)

Марія ЮХИМЧУК

АНОТАЦІЯ

Бакалаврська робота складається з 118 сторінок формату А4, на яких представлено 40 рисунків, 2 таблиці, список використаних джерел містить 55 найменувань.

Мета роботи – розробка автоматизованої системи управління орендою робочих місць у коворкінгах, яка б забезпечувала ефективне та зручне керування процесами бронювання, оплати та контролю доступу, а також сприяла підвищенню якості обслуговування клієнтів і оптимізації внутрішніх операцій.

В теоретичній частині роботи досліджено сучасний стан систем управління у сфері коворкінгів, визначено їхні основні компоненти, технологічний рівень та перспективи розвитку, а також проаналізовано існуючі виклики і обмеження. В практичній частині здійснено проектування та розробку системи, включаючи структуру бази даних, UML-діаграми та вибір технологій для реалізації. Проведено тестування системи та усунення виявлених недоліків, що підтверджує її працездатність і відповідність сучасним вимогам галузі.

Ключові слова: автоматизована система, управління орендою, коворкінг, бронювання, база даних, UML-діаграми, тестування.

ABSTRACT

The Bachelor's thesis consists of 118 pages of A4 format, including 40 illustrations, tables, and a list of references containing 55 sources.

The aim of the work is to develop an automated management system for renting workspaces in coworking spaces that ensures efficient and convenient control of booking, payment, and access management processes, as well as contributes to improving customer service quality and optimizing internal operations.

In the theoretical part, the current state of management systems in the coworking industry was studied, their main components, technological level, and development prospects were defined, and existing challenges and limitations were analyzed. In the practical part, the system was designed and developed, including the database structure, UML diagrams, and the choice of technologies for implementation. The system was tested, and identified deficiencies were eliminated, confirming its operability and compliance with modern industry requirements.

Keywords: automated system, rental management, coworking, booking, database, UML diagrams, testing

Зміст

ВСТУП.....	4
1 ДОСЛІДЖЕННЯ ОБ’ЄКТА АВТОМАТИЗАЦІЇ ТА АНАЛІЗ РОБОТИ СИСТЕМ УПРАВЛІННЯ ПОСЛУГАМИ В СФЕРІ КОВОРКІНГІВ	8
1.1 Аналіз основних компонентів систем управління послугами в сфері коворкінгів	10
1.2 Дослідження технологічного рівня систем управління орендою робочих місць у коворкінгах	15
1.3 Аналіз обмежень і викликів сучасних систем управління орендою у сфері коворкінгів	17
1.4 Перспективи розвитку та інноваційні тенденції.....	20
1.5 Уточнена постановка задачі.....	22
2 ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СУЧАСНИХ ПРОГРАМНИХ РІШЕНЬ.....	24
2.1 Nexodus.....	25
2.2 OfficeRnD	29
2.3 Coworkify.....	33
2.4 Cobot	36
2.5 Satellite Deskworks.....	39
3 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ОРЕНДОЮ РОБОЧИХ МІСЦЬ У КОВОРКІНГАХ	44
3.1 Уточнена постановка задачі та визначення ключових функцій системи.....	44
3.2 Розробка структури бази даних	44
3.3 Розробка UML-діаграм	47
3.3.1 Опис UML-діаграми класів	48
3.3.2 Опис UML-діаграми варіантів використання	50
3.3.3 Опис UML-діаграми активностей	52

3.4 Вибір архітектури системи.....	55
------------------------------------	----

4 РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ОРЕНДОЮ РОБОЧИХ МІСЦЬ У КОВОРКІНГАХ	58
--	-----------

4.1 Вибір технологій для реалізації системи	58
---	----

4.2 Реалізація основних компонентів системи	61
---	----

4.3 Тестування роботи автоматизованої системи управління	65
--	----

4.3.1 Модуль користувача	65
--------------------------------	----

4.3.2 Модуль адміністратора.....	71
----------------------------------	----

ВИСНОВКИ	74
-----------------------	-----------

Список використаних джерел.....	76
--	-----------

Додатки(обов'язковий)	83
------------------------------------	-----------

Додаток А(обовязковий) Протокол перевірки на наявність запозичень.....	84
---	-----------

Додаток Б(обовязковий) Технічне завдання.....	85
--	-----------

Додаток В(довідковий) Лістинг	88
--	-----------

Додаток Г(обовязковий) Ілюстративна частина.....	94
---	-----------

ВСТУП

Актуальність. В умовах стрімкого зростання популярності гнучких робочих просторів та розширення мережі коворкінгів зростає потреба у впровадженні сучасних автоматизованих систем управління. Щорічно кількість таких просторів збільшується, що ускладнює їх ефективне адміністрування без застосування сучасних інформаційних технологій. Відсутність єдиного інтегрованого рішення призводить до низької оперативності обробки запитів, зниження рівня сервісу та підвищення ймовірності помилок при управлінні бронюваннями, платежами та доступом до робочих місць [1, 2].

На ринку існує значна кількість платформ, що пропонують базові функції бронювання та контролю доступу, але їх можливості часто обмежені, а інтеграція з іншими системами – недостатньо гнучка. Це ускладнює масштабування бізнесу та адаптацію під індивідуальні потреби конкретного коворкінгу. Водночас, конкуренція зростає, і для підвищення конкурентоспроможності необхідно мати сучасну платформу, що автоматизує всі ключові процеси: від бронювання і оплати до управління користувачами та формування аналітичних звітів.

Автоматизовані системи управління орендою робочих місць дозволяють централізовано керувати всіма аспектами роботи коворкінгу, скорочуючи час обробки запитів, зменшуючи людський фактор і підвищуючи точність операцій. Вони забезпечують зручний інтерфейс для клієнтів, що дозволяє швидко бронювати робочі місця, оплачувати послуги та отримувати підтвердження у реальному часі. Для адміністрації такі системи відкривають доступ до аналітики, що дозволяє оцінювати завантаженість, популярність послуг, сезонні коливання та інші важливі показники. Це сприяє більш точному плануванню ресурсів і формуванню маркетингових стратегій [3].

Крім того, сучасні технології, такі як мобільні додатки, системи автоматичного контролю доступу через QR-коди або RFID, інтеграція з платформами оплати та

аналітичними інструментами, відкривають нові можливості для підвищення рівня автоматизації. Впровадження штучного інтелекту та машинного навчання дозволяє прогнозувати попит, оптимізувати використання робочих місць та персоналізувати пропозиції для клієнтів. Це забезпечує не лише підвищення ефективності управління, але й створює конкурентну перевагу у динамічно змінюваній галузі.

Отже, актуальність дослідження зумовлена необхідністю створення сучасної, масштабованої та гнучкої автоматизованої системи управління орендою робочих місць у коворкінгах, здатної задовольнити зростаючі потреби галузі, підвищити її ефективність та конкурентоспроможність.

Мета і завдання дослідження. Метою дипломної роботи є розробка автоматизованої системи управління орендою робочих місць у коворкінгах, яка б забезпечувала ефективне та зручне керування процесами бронювання, оплати та контролю доступу, а також сприяла підвищенню якості обслуговування клієнтів і оптимізації внутрішніх операцій.

Для досягнення поставленої мети були сформульовані та вирішені наступні задачі:

- 1) проведено дослідження сучасного стану ринку автоматизованих систем управління у сфері коворкінгів, визначено основні тенденції та проблеми;
- 2) здійснено аналіз існуючих програмних рішень, що використовуються в галузі, з метою визначення їх функціональних можливостей, сильних і слабких сторін;
- 3) досліджено механізми управління і контролю за процесами оренди, включаючи системи аналітики, звітності та автоматичного моніторингу доступу;
- 4) визначено ключові функціональні вимоги до розроблюваної системи;
- 5) здійснено проектування архітектури системи, включаючи розробку UML-діаграм та структури бази даних;
- 6) виконано розробку системи із застосуванням сучасних технологій, а також проведено її тестування та оцінку функціональності.

Об'єктом дослідження є процеси управління орендою робочих місць у коворкінгах, включаючи бронювання, оплату, контроль доступу та аналіз діяльності.

Предметом дослідження є методи, підходи та технічні засоби розробки автоматизованої системи управління, що забезпечують автоматизацію ключових процесів та підвищують ефективність роботи коворкінгів.

Методи дослідження. В процесі виконання дослідження застосовувалися методи системного аналізу для вивчення процесів управління орендою робочих місць у коворкінгах, методи аналізу та проектування баз даних для створення ефективної структури зберігання інформації, а також сучасні підходи до розробки програмного забезпечення, зокрема Agile-методології (Scrum, Kanban) для гнучкого управління проектом. Для реалізації функціоналу системи використовувалися мови програмування JavaScript, HTML, CSS для фронтенд-розробки, а також мови Python або Node.js для серверної частини, з застосуванням RESTful API для інтеграції компонентів.

Новизна одержаних результатів полягає у наступному:

1. Створено автоматизовану систему управління орендою робочих місць у коворкінгах, яка враховує всі ключові процеси – від бронювання до контролю доступу та формування аналітичних звітів. Враховано особливості сучасних коворкінгів, що дозволяє підвищити ефективність управління та рівень обслуговування клієнтів.

2. Розроблено ефективну модель бази даних, яка відповідає сучасним вимогам та потребам галузі, забезпечує швидкий пошук, зручне бронювання та обробку транзакцій. Створена структура бази даних дозволяє масштабувати систему та інтегрувати додаткові функціональні можливості.

3. Впроваджено інтуїтивний та зручний інтерфейс користувача, що забезпечує легкий доступ до всіх функцій системи для адміністраторів та клієнтів, сприяє підвищенню рівня задоволеності та зменшенню часу на навчання нових користувачів.

Практичне значення одержаних результатів полягає у створенні високоефективної автоматизованої системи, яка дозволяє оптимізувати процеси бронювання та управління орендою робочих місць, підвищує продуктивність роботи адміністрації, зменшує людські помилки та забезпечує прозорість операцій. Це сприяє підвищенню конкурентоспроможності коворкінгів, розширенню клієнтської бази та покращенню якості наданих послуг.

1 ДОСЛІДЖЕННЯ ОБ'ЄКТА АВТОМАТИЗАЦІЇ ТА АНАЛІЗ РОБОТИ СИСТЕМ УПРАВЛІННЯ ПОСЛУГАМИ В СФЕРІ КОВОРКІНГІВ

Темою бакалаврської кваліфікаційної роботи є створення автоматизованої системи управління орендою робочих місць у коворкінгах, що покликане підвищити ефективність адміністрування, оптимізувати процеси бронювання та доступу, а також забезпечити прозорість і зручність взаємодії між користувачами та адміністрацією. В рамках дослідження особливу увагу приділяється аналізу сучасного стану галузі, визначенню ключових вимог до автоматизованих систем, а також оцінці технологічних тенденцій, що впливають на її розвиток.

Аналіз сучасних технологічних рішень показує, що для створення такої системи необхідно застосовувати сучасні підходи до розробки, зокрема використання веб-технологій, мобільних платформ, систем автоматичного контролю доступу та інтеграції з платіжними сервісами. Важливим аспектом є також можливість розширення функціоналу та адаптація до змін ринкових умов, що визначає необхідність застосування гнучких архітектурних рішень та сучасних методів управління проектами [4].

Загалом, дослідження об'єкта автоматизації дозволяє сформулювати основні вимоги до системи, визначити її функціональні модулі та технологічний стек, а також окреслити перспективи розвитку у контексті швидкозмінних технологічних тенденцій і зростаючих потреб галузі.

Місце задачі, що розв'язується в роботі, у загальному комплексі задач автоматизації: автоматизація процесів управління орендою робочих місць у коворкінгах є ключовим компонентом сучасних систем управління у сфері гнучких робочих просторів. Вона сприяє підвищенню швидкості обробки запитів, зменшенню людських помилок і підвищенню надійності операцій, що є особливо важливим у контексті високої конкуренції та динамічних ринкових умов. Впровадження автоматизованих процесів дозволяє оптимізувати управління

бронюваннями, контролем доступу, оплатами та аналітикою, забезпечуючи зручність для користувачів і ефективність для адміністраторів.

Автоматизація допомагає скоротити час, витрати та ресурси, необхідні для ручного виконання операцій, зменшує ризик людських помилок і забезпечує рівномірне навантаження на персонал. Водночас, вона дозволяє швидко масштабувати бізнес, додавати нові функції та інтегруватися з іншими системами, що є важливим у контексті розширення мережі коворкінгів. Особливу роль відіграє можливість обробки великих обсягів даних у реальному часі, що дозволяє отримувати цінну аналітичну інформацію для стратегічного планування і прийняття рішень [5].

Область застосування розробки: розроблена автоматизована система орієнтована на управління послугами у сфері гнучких робочих просторів, зокрема, у коворкінгах, що пропонують оренду робочих місць на короткостроковій основі. Основними користувачами системи є адміністратори коворкінгів, менеджери з управління нерухомістю, а також самі орендарі – фрілансери, малий бізнес і компанії, що шукають зручний спосіб бронювання робочих місць через онлайн-платформу.

Вимоги до програмно-технічного забезпечення, що розробляється у рамках бакалаврської роботи, визначаються сучасними стандартами та технічними можливостями для створення ефективної системи управління орендою робочих місць у коворкінгах. Основною базою даних обрана PostgreSQL через її високий рівень продуктивності, масштабованості та гнучкості у роботі з великими обсягами даних, що є критично важливим для обробки інформації про користувачів, бронювання та статистику.

Веб-сервер – Apache, який забезпечує стабільний хостинг та обслуговування веб-додатку, підтримуючи безпечне та швидке з'єднання з клієнтськими пристроями. Серверна частина системи реалізована на мові Python з використанням фреймворку Django, що дозволяє швидко розробляти надійний бекенд із високим

рівнем безпеки, а також забезпечує гнучкість у розширенні функціоналу та інтеграції з іншими сервісами.

Фронтенд-інтерфейс створений за допомогою HTML, CSS і JavaScript, з інтеграцією фреймворку Bootstrap для швидкої розробки адаптивного та зручного інтерфейсу користувача. Це забезпечує інтуїтивно зрозумілу навігацію, швидкий доступ до функцій системи та зручність роботи для адміністраторів і клієнтів.

Функціональні можливості системи включають автоматизований пошук і бронювання робочих місць, управління базою клієнтів та їх бронюваннями, моніторинг стану бронювань з можливістю швидкого реагування на зміни, а також представлення повної інформації про послуги – включаючи відгуки користувачів, фотографії, ціни та доступність.

Необхідним результатом готовності розробки є перевірка функціональності системи, що здійснюється через проведення тестування процесів пошуку, бронювання і надання послуг відповідно до визначених вимог. Це включає тестування коректності роботи інтерфейсу, швидкості обробки запитів, безпеки даних і стабільності системи під навантаженням.

1.1 Аналіз основних компонентів систем управління послугами в сфері коворкінгів

Історія автоматизованих систем управління у сфері коворкінгів бере початок із перших рішень, що з'явилися на ринку наприкінці 2000-х років. Спочатку це були прості програмні продукти, що забезпечували базові функції бронювання та обліку клієнтів, здебільшого у вигляді локальних додатків або веб-сайтів із статичним контентом. З поширенням інтернет-технологій і мобільних пристроїв зросла потреба у більш динамічних і гнучких рішеннях, що дозволяли б керувати орендою робочих місць у реальному часі.

Протягом останніх десяти років технологічний розвиток значно прискорився. З'явилися комплексні платформи, що інтегрують функції бронювання, оплати, управління користувачами, контролю доступу та аналітики. Вони базуються на

сучасних архітектурах із застосуванням хмарних сервісів, що забезпечують масштабованість і високу доступність. Інноваційним кроком стало впровадження мобільних додатків, що дозволяють клієнтам керувати бронюваннями через смартфони, а також інтеграція з системами автоматичного контролю доступу (RFID, QR-коди, біометрія).

Сучасний стан галузі характеризується високою конкуренцією та швидким темпом технологічних змін. На ринку представлені як глобальні платформи, так і локальні рішення, що адаптовані до специфіки конкретних регіонів і клієнтських потреб. Водночас, зростає увага до аналітики даних, прогнозування попиту та автоматизації управлінських процесів за допомогою штучного інтелекту та машинного навчання [6].

Основні компоненти включають модулі бронювання, управління користувачами, платіжні системи, контроль доступу, а також аналітичні та звітні інструменти. Взаємодія між цими компонентами забезпечує цілісність і зручність роботи системи, дозволяючи швидко реагувати на запити клієнтів, автоматизувати рутинні операції та підтримувати високий рівень сервісу [7, 8].

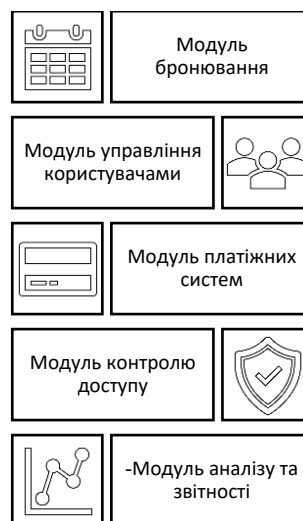


Рисунок 1.1 – Основні компоненти систем управління послугами в сфері коворкінгів

Детальний опис кожного компонента дозволить розкрити їхню роль, функціональні можливості та інтеграційні зв'язки у системі управління орендою робочих місць:

- Модуль бронювання

Модуль бронювання є центральною складовою системи управління орендою робочих місць у коворкінгах. Він забезпечує можливість користувачам швидко та зручно резервувати робочі місця через веб-інтерфейс або мобільний додаток. Основні функції цього модуля включають пошук доступних місць за обраним часом і датою, вибір конкретного робочого місця, підтвердження бронювання та автоматичне оновлення статусу доступності.

Для підвищення зручності в реалізації зазвичай використовується календарний інтерфейс, що дозволяє користувачам бачити доступність у реальному часі, а також фільтри за типами місць, ціною або додатковими послугами. Модуль бронювання також інтегрується з платіжними системами для автоматичної обробки оплат та формування квитанцій.

Завдяки цьому компоненту система здатна автоматизувати процес резервування, зменшити час очікування та людський фактор, а також забезпечити актуальну інформацію про доступність робочих місць у будь-який час.

- Модуль управління користувачами

Модуль управління користувачами відповідає за обробку інформації про клієнтів, їхні профілі та рівень доступу до системи. Він забезпечує реєстрацію нових користувачів, управління їхніми особистими даними, а також налаштування прав і ролей для різних категорій користувачів – клієнтів, адміністраторів, менеджерів.

Основні функції цього модуля включають створення та редагування профілів, автентифікацію та авторизацію, а також управління історією бронювань і платежів. Для зручності користувачів реалізовується можливість входу через соціальні мережі або електронну пошту з двофакторною аутентифікацією для підвищення безпеки.

Цей компонент забезпечує цілісність даних про користувачів, дозволяє швидко знаходити та редагувати інформацію, а також контролювати рівень доступу до різних функцій системи. В результаті система здатна підтримувати високий рівень безпеки та персоналізації обслуговування, що сприяє підвищенню задоволеності клієнтів і зниженню ризиків несанкціонованого доступу.

- Модуль платіжної системи

Модуль платіжної системи відповідає за обробку фінансових операцій, пов'язаних із бронюванням і орендою робочих місць. Він забезпечує інтеграцію з популярними платіжними шлюзами та сервісами, що дозволяє користувачам здійснювати оплату онлайн у зручний для них спосіб – через кредитні/дебетові карти, електронні гаманці або мобільні платформи.

Основні функції цього модуля включають автоматичне формування рахунків, обробку платежів у реальному часі, зберігання історії транзакцій та генерацію квитанцій. Також можлива підтримка різних валют і налаштування тарифів залежно від типу послуги або часу оренди.

Модуль платіжної системи інтегрується з модулем бронювання для автоматичного підтвердження оплати і оновлення статусу броні. Це дозволяє зменшити людський фактор, підвищити швидкість обробки операцій і забезпечити високий рівень безпеки за рахунок використання сучасних протоколів шифрування та відповідності стандартам PCI DSS (Payment Card Industry Data Security Standard).

Завдяки цьому компоненту система здатна забезпечити безперебійний і безпечний фінансовий супровід усіх операцій, що підвищує довіру клієнтів і сприяє зростанню доходів [9].

- Модуль контролю доступу

Основні функції цього модуля включають генерацію унікальних кодів або карт для кожного бронювання, автоматичне відкриття дверей або воріт при підтвердженні броні, а також ведення журналу входів і виходів для кожного

користувача. В системі реалізовано можливість ручного або автоматичного блокування доступу у разі порушень правил або закінчення орендного періоду.

Цей компонент забезпечує високий рівень безпеки, запобігає несанкціонованому доступу, а також дозволяє адміністратору оперативно контролювати ситуацію у приміщеннях. Водночас, інтеграція з мобільними додатками або системами оплати дозволяє автоматично відкривати доступ для користувачів після підтвердження оплати або бронювання, що підвищує зручність і швидкість обслуговування.

- Модуль аналізу та звітності

Модуль аналізу та звітності є важливою складовою системи управління орендою робочих місць, яка забезпечує збір, обробку і представлення даних для прийняття управлінських рішень. Він інтегрує інформацію з усіх інших модулів системи – бронювання, платежів, доступу та управління користувачами – і формує детальні звіти, діаграми та аналітичні показники.

Основні функції цього модуля включають автоматичне створення звітів про завантаженість робочих місць, фінансові показники, активність користувачів, популярність послуг і сезонні тренди. Також реалізовано можливість налаштовувати кастомізовані звіти за різними параметрами, періодами та сегментами клієнтів.

Модуль підтримує інтеграцію з системами BI (Business Intelligence), що дозволяє проводити глибокий аналіз великих обсягів даних, прогнозувати попит і оптимізувати управлінські рішення.

Завдяки цьому компоненту керівництво отримує цінну інформацію для стратегічного розвитку, а менеджери – інструменти для оперативного реагування на зміни у попиті та поведінці клієнтів [10, 11].

1.2 Дослідження технологічного рівня систем управління орендою робочих місць у коворкінгах

На сучасному ринку автоматизованих систем управління орендою робочих місць у коворкінгах домінують платформенні рішення, що базуються на сучасних веб-технологіях і архітектурних підходах, здатних забезпечити високий рівень масштабованості, гнучкості та безпеки. Вивчення існуючих систем показує, що більшість із них використовують клієнт-серверну архітектуру, що дозволяє розподіляти навантаження між клієнтською та серверною частинами, забезпечуючи стабільну роботу навіть при високій кількості користувачів.

У ролі серверної платформи найчастіше застосовуються фреймворки на основі мови Python, зокрема Django або Flask, що забезпечують швидкий розвиток, безпеку та підтримку REST API для обміну даними з клієнтською частиною. Вони працюють на серверних платформах Linux, що гарантує високу стабільність та масштабованість. Для зберігання даних використовуються реляційні бази даних.

Клієнтська частина системи, як правило, реалізується на сучасних фронтенд-фреймворках, таких як React або Vue.js, що дозволяє створювати динамічний і адаптивний інтерфейс користувача. Вони забезпечують швидке реагування системи, зручну навігацію та інтерактивність, що особливо важливо для мобільних користувачів [12].

Ще одним важливим аспектом архітектурного рішення є використання хмарних сервісів – Amazon Web Services (AWS), Google Cloud Platform або Microsoft Azure. Це дозволяє легко масштабувати інфраструктуру відповідно до зростання кількості користувачів і обсягів даних, а також забезпечує високий рівень доступності та безпеки.

Для підвищення надійності системи багато рішень застосовують контейнеризацію (Docker) і оркестрацію (Kubernetes), що дозволяє швидко розгортати, масштабувати та оновлювати компоненти системи без простоїв. Таке

рішення особливо актуальне для великих коворкінг-мереж або мережевих платформ, що обслуговують сотні або тисячі клієнтів одночасно.

Загалом, архітектурні рішення сучасних систем управління орендою у сфері коворкінгів базуються на модульних, масштабованих і безпечних компонентах, що забезпечують гнучкість у розвитку і інтеграцію з іншими системами.

Значна кількість систем використовують стандартні RESTful API для інтеграції з зовнішніми сервісами, такими як CRM-системи (наприклад, Salesforce або HubSpot), платіжні шлюзи (Stripe, PayPal, LiqPay), системи контролю доступу, а також системи електронної пошти та SMS-розсилок для комунікації з клієнтами. Це дозволяє автоматизувати процеси оповіщення, підтвердження бронювань, обробку платежів та контроль входу/виходу без необхідності ручного втручання.

Ще одним важливим аспектом є використання платформ хмарних сервісів для масштабування інфраструктури. Це дозволяє системам швидко збільшувати або зменшувати обсяги оброблюваних даних і кількість користувачів без зниження продуктивності. Контейнеризація та оркестрація дозволяють автоматично розгортати нові версії системи, масштабувати окремі компоненти та забезпечувати безперебійну роботу навіть при високих навантаженнях [13].

Щодо рівня автоматизації, сучасні системи прагнуть максимально зменшити участь людини у рутинних операціях. Це досягається за рахунок автоматичного оновлення статусів бронювань, автоматичного виставлення рахунків і платежів, автоматичного відкриття та закриття доступу, а також формування аналітичних звітів без необхідності ручного втручання. Впровадження штучного інтелекту і машинного навчання дозволяє системам прогнозувати попит, оптимізувати ціноутворення і навіть автоматично пропонувати клієнтам додаткові послуги.

Масштабованість системи забезпечується не лише за рахунок хмарних платформ та контейнерних рішень, але й за рахунок модульної архітектури, що була описана в попередньому підрозділі [14]. Це дозволяє додавати нові функціональні модулі, інтегруватися з новими сервісами і розширювати можливості без

необхідності перепроєктування всього рішення. Такі підходи особливо актуальні для мережових коворкінг-операторів, що прагнуть швидко реагувати на зміни ринку та масштабувати бізнес.

1.3 Аналіз обмежень і викликів сучасних систем управління орендою у сфері коворкінгів

Незважаючи на значний прогрес і широке впровадження сучасних технологій, існуючі системи управління орендою у сфері коворкінгів мають низку обмежень і проблем, що стримують їхній потенціал і знижують рівень задоволеності користувачів, таких як: недостатня гнучкість налаштувань, низька інтеграція з новими технологіями, проблеми безпеки і захисту даних, а також низький рівень аналітики і прогнозування.

- Недостатня гнучкість налаштувань

Дана проблема є однією з найпоширеніших проблем сучасних систем управління, оскільки багато платформ мають заздалегідь визначені шаблони та обмежені можливості кастомізації, що ускладнює адаптацію системи до специфіки конкретного коворкінгу або розширення функціоналу відповідно до нових вимог.

Наприклад, у багатьох системах відсутня можливість гнучко налаштовувати тарифні плани, додавати нові типи послуг або змінювати правила бронювання без залучення розробників або технічної підтримки. Це призводить до того, що адміністратори змушені працювати у рамках обмеженого функціоналу, що не враховує особливості їхнього бізнесу, сезонних коливань або нових сервісів. В результаті виникає ситуація, коли система перестає бути універсальним інструментом і лише стримує розвиток бізнесу на оренді.

Крім того, низька гнучкість у налаштуваннях впливає на можливості інтеграції з іншими системами або модифікації інтерфейсу під потреби користувачів. Наприклад, не всі системи дозволяють легко додавати нові мовні версії, змінювати

вигляд інтерфейсу або додавати індивідуальні функції без глибокого втручання у код. Це ускладнює адаптацію системи до змінних умов ринку і знижує її конкурентоспроможність.

Ще одним аспектом є обмежена можливість налаштування автоматичних правил, наприклад, для формування цінових акцій, знижок або спеціальних пропозицій [15 -17]. У багатьох платформах ці функції є або дуже обмеженими, або відсутні взагалі. Це ускладнює маркетингову діяльність і персоналізацію пропозицій для різних груп клієнтів.

- Низька інтеграція з новими технологіями

Ще однією значущою проблемою сучасних систем є недостатня інтеграція з новими технологіями, що стримує їхній розвиток і знижує рівень автоматизації процесів. У світі, де мобільні додатки, інтернет речей (IoT), штучний інтелект та машинне навчання стають стандартом для багатьох галузей, системи управління орендою повинні активно використовувати ці інновації для підвищення ефективності та зручності.

На жаль, багато платформ мають обмежену підтримку сучасних технологій. Наприклад, мобільні додатки часто створюються як окремі рішення, а не як частина єдиної системи, що ускладнює їхню інтеграцію та синхронізацію даних. Це призводить до дублювання функціоналу, розпорошення даних і ускладнює управління бронюваннями та оплатами через різні пристрої.

Ще одна проблема – низька підтримка IoT-інтеграцій. У сучасних коворкінгах широко використовуються системи автоматичного контролю доступу, датчики присутності, системи освітлення та кондиціонування, що можуть бути підключені до систем управління для автоматичного реагування на події. Однак багато платформ не мають відкритих API або підтримки стандартних протоколів для інтеграції з цими пристроями, що ускладнює створення єдиної автоматизованої екосистеми [18].

Багато систем не підтримують інтеграцію з популярними платформами для комунікації (наприклад, Slack, Telegram), системами управління завданнями або CRM, що ускладнює автоматизацію внутрішніх процесів і знижує оперативність реагування на запити клієнтів.

- Проблеми безпеки і захисту даних

Багато систем недостатньо враховують сучасні стандарти безпеки при обробці конфіденційної інформації, такої як особисті дані користувачів, платіжні реквізити або дані систем контролю доступу. Це може бути зумовлено застарілими протоколами шифрування, відсутністю двофакторної автентифікації або необов'язковим застосуванням сучасних стандартів безпеки або захисту персональних даних.

Ще одним викликом є можливість кібератак, зокрема, SQL-ін'єкцій, DDoS-атак або зломів через вразливості у сторонніх компонентах системи. У багатьох випадках системи мають недостатній рівень захисту від таких загроз, що може призвести до втрати або пошкодження даних.

Крім того, багато платформ не мають належних механізмів моніторингу безпеки або систем оповіщення про підозрілі дії. Це ускладнює швидке реагування на інциденти і підвищує ризики компрометації системи.

Ще одна проблема – це недостатній рівень контролю доступу, що може призводити до несанкціонованого доступу до конфіденційної інформації або управління системою. Відсутність розмежування прав доступу для різних ролей (адміністраторів, клієнтів, технічних фахівців) створює додаткові ризики.

- Низький рівень аналітики і прогнозування

Ще одним суттєвим викликом сучасних систем є недостатній рівень аналітичних можливостей і прогнозних механізмів. Багато платформ здатні збирати та зберігати дані, але не мають інструментів для їхнього глибокого аналізу, що обмежує здатність керівництва приймати обґрунтовані рішення.

Зазвичай, системи пропонують стандартні звіти про завантаженість, фінансові показники або активність користувачів, але ці дані часто не мають інтерактивних інструментів для візуалізації або глибокого аналізу. Відсутність можливості налаштовувати аналітичні дашборди або проводити прогнозування за допомогою машинного навчання обмежує здатність бізнесу реагувати на зміни у попиту, сезонні коливання або поведінку клієнтів.

Крім того, багато систем не здатні автоматично виявляти тренди або аномалії, що ускладнює вчасне реагування на потенційні проблеми або можливості для зростання [19, 20]. Це призводить до того, що управлінські рішення базуються здебільшого на інтуїції або ручному аналізі, що є менш ефективним і більш схильним до помилок.

Ще один аспект – відсутність прогнозних моделей ціноутворення або автоматичних рекомендацій щодо оптимізації використання ресурсів. Це означає, що системи не допомагають у довгостроковому плануванні або стратегічному розвитку бізнесу, що є критичним у конкурентних умовах сучасного ринку.

Проведений аналіз показує, що сучасні системи управління орендою у сфері коворкінгів мають суттєві обмеження у кількох ключових напрямках. Недостатня гнучкість налаштувань, низька інтеграція з новими технологіями, проблеми безпеки і захисту даних, а також низький рівень аналітики і прогнозування – це виклики, що потребують уваги і постійного вдосконалення.

1.4 Перспективи розвитку та інноваційні тенденції

Розвиток технологій та змінювані потреби ринку формують нові можливості для вдосконалення систем управління орендою у сфері коворкінгів. Враховуючи швидкий темп цифрової трансформації та зростаючу конкуренцію, сучасні рішення мають рухатися у напрямках, що забезпечують ще більшу автоматизацію, гнучкість і інноваційність.

Одним із головних напрямків, що визначають майбутнє галузі, є інтеграція штучного інтелекту та машинного навчання. Вже сьогодні ці технології використовуються для аналізу великих обсягів даних, прогнозування попиту і автоматичного налаштування цінових стратегій. В майбутньому їх застосування стане ще ширшим, дозволяючи системам автоматично адаптуватися до змін у поведінці клієнтів, сезонних коливань і ринкових трендів. Це відкриває можливості для створення динамічного ціноутворення, яке враховує не лише поточний попит, але й прогнозу аналітику, що базується на історичних даних і зовнішніх факторах. Такий підхід дозволить коворкінгам максимально ефективно використовувати свої ресурси і підвищити конкурентоспроможність [21].

Ще одним важливим напрямком є розвиток мобільних і IoT-технологій. Мобільні додатки вже зараз є невід'ємною частиною систем управління, але їхня функціональність буде значно розширюватися. У майбутньому вони зможуть не лише забезпечувати бронювання та оплату, а й інтегруватися з системами автоматичного контролю доступу, датчиками присутності, системами освітлення та кондиціонування. Це дозволить створювати повністю автоматизовані та саморегулюючіся середовища, де всі процеси відбуватимуться без участі людини. Така інтеграція дозволить зменшити операційні витрати, підвищити безпеку і забезпечити ще більший комфорт для користувачів.

Розвиток технологій блокчейн відкриває нові можливості для підвищення прозорості та безпеки транзакцій у системах управління орендою. Впровадження децентралізованих реєстрів дозволить автоматизувати процеси укладання договорів, підтвердження оплат і контролю за виконанням умов оренди без залучення посередників. Це не лише підвищить рівень безпеки, але й знизить операційні витрати, а також забезпечить прозорість і довіру між сторонами.

Ще одним важливим трендом є розвиток систем аналітики та прогнозування, що базуються на великих даних (Big Data). Вони дозволяють не лише отримувати звіти про поточний стан справ, а й прогнозувати майбутні тенденції, виявляти слабкі

місця і можливості для зростання. Це особливо актуально для мережеских коворкінгів, які прагнуть оптимізувати використання ресурсів і відповідати високим вимогам клієнтів. В майбутньому системи будуть використовувати більш глибокий аналіз даних, застосовувати алгоритми машинного навчання для автоматичного виявлення закономірностей і формування рекомендацій щодо подальших дій.

Значну роль у майбутньому відіграватиме розвиток віртуальної та доповненої реальності (VR/AR). Технології VR і AR вже сьогодні знаходять застосування у створенні віртуальних турів, презентацій і демонстрацій об'єктів, що дозволяє потенційним клієнтам ознайомитися з робочими місцями і інфраструктурою ще до фактичного відвідування. У майбутньому ці технології стануть ще більш інтегрованими у системи управління, дозволяючи клієнтам за допомогою VR/AR проводити попередні бронювання, віртуально оглядати приміщення, а також створювати персоналізовані робочі простори.

Ще одним перспективним напрямком є застосування робототехніки та автоматизованих систем обслуговування [22-24]. В майбутньому роботи можуть виконувати функції технічного обслуговування, доставки, контролю безпеки і навіть комунікації з клієнтами. Це дозволить знизити операційні витрати і підвищити рівень сервісу, а також зробити управління більш гнучким і безпечним.

1.5 Уточнена постановка задачі

У ході проведеного дослідження було здійснено всебічний аналіз сучасного стану систем управління орендою у сфері коворкінгів, що дозволив визначити їхні основні компоненти, технологічний рівень та перспективи розвитку. Детально розглянуто функціональні модулі таких систем, зокрема модулі бронювання, управління користувачами, платіжні системи, контроль доступу та аналітичні інструменти. Виявлено, що сучасні рішення базуються на модульній, масштабованій архітектурі, що забезпечує інтеграцію з різноманітними зовнішніми сервісами та

платформами, а також активно застосовують новітні технології – від мобільних додатків і IoT до штучного інтелекту та блокчейну.

Разом із тим, було досліджено актуальні обмеження і виклики, з якими стикаються сучасні системи. Зокрема, відзначено недостатню гнучкість налаштувань, що ускладнює адаптацію під специфічні потреби окремих бізнесів і швидке реагування на зміни ринку.

Крім того, досліджено перспективи розвитку галузі, зокрема застосування штучного інтелекту, IoT, блокчейну, VR/AR та робототехніки. Виявлено, що ці технології мають потенціал суттєво змінити існуючий ландшафт автоматизованих систем, зробити їх більш гнучкими, безпечними і розумними, що відкриває широкі можливості для створення інноваційних сервісів і підвищення конкурентоспроможності.

Отже, проведений аналіз дозволив сформулювати чітке уявлення про сучасний рівень розвитку систем управління орендою у сфері коворкінгів, їхні сильні сторони та існуючі проблеми. Водночас, для подальшої роботи необхідно перейти до більш практичного етапу – розгляду конкретних існуючих систем, що представлені на ринку. Це дасть змогу порівняти отримані висновки з реальними рішеннями, виявити їхні переваги і недоліки, а також сформулювати чіткі вимоги до розробки власної автоматизованої системи управління орендою робочих місць у коворкінгах.

2 ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ СУЧАСНИХ ПРОГРАМНИХ РІШЕНЬ

Аналіз функціональних можливостей сучасних програмних рішень, що використовуються для управління орендою робочих місць у коворкінгах, є критичним етапом в процесі проектування та впровадження власної інформаційної системи. Дослідження наявних рішень дозволяє здійснити техніко-функціональну оцінку конкурентного середовища, виокремити типові підходи до автоматизації бізнес-процесів, а також виявити недоліки, які можуть бути усунені в майбутньому продукті [25, 26].

Порівняльний аналіз забезпечує такі переваги:

- Виявлення архітектурних та функціональних обмежень: Аналізуючи наявні продукти, можна виявити слабкі сторони, зокрема у масштабованості, гнучкості налаштувань або інтерфейсній взаємодії, що дозволяє уникнути їх повторення.
- Ідентифікація ключових функцій: Вивчення ринку сприяє виявленню найбільш затребуваних серед користувачів функцій, таких як динамічне управління робочими зонами, інтеграція з календарями, інструменти обліку часу та платежів, статистичний моніторинг тощо.
- Аналіз конкурентних переваг: Дослідження диференційних ознак дозволяє визначити потенційні напрями підвищення конкурентоспроможності власного продукту, зокрема через розширення спектра інтеграцій або впровадження модулів на базі штучного інтелекту.
- Оцінка технологічної бази: Аналіз платформ, мов програмування, архітектур (наприклад, мікросервісної), моделей розгортання (SaaS/PaaS/on-premise), а також використання API та SDK дозволяє визначити оптимальні підходи для реалізації системи.

- Формування користувацького досвіду (UX): Вивчення інтерфейсних рішень дозволяє оцінити ергономіку, логіку навігації, мобільність та адаптивність систем.

У рамках дослідження було обрано п'ять сучасних рішень, які активно застосовуються у сфері управління коворкінгами: Nexodus, OfficeRnD, Coworkify, Cobot та Satellite Deskworks. Вибір здійснено з урахуванням таких критеріїв, як популярність у середовищі операторів коворкінгів, функціональне наповнення, підтримка API та технологічна гнучкість. Всі зазначені сервіси реалізовані за моделлю хмарного програмного забезпечення (SaaS) та мають масштабовану архітектуру, що дозволяє обслуговувати як малі, так і великі простори.

2.1 Nexodus

Nexodus є одним з найбільш функціонально насичених і технологічно просунутих рішень для автоматизованого управління просторами спільної роботи. Заснована у 2012 році платформа набула значного поширення серед операторів коворкінгів, бізнес-центрів, гнучких офісів і інноваційних хабів у понад 90 країнах світу. Система реалізована за моделлю хмарного програмного забезпечення (SaaS) та підтримує мультиорендний режим із можливістю адміністрування декількох локацій в єдиному інтерфейсі [27].

Nexodus охоплює повний цикл автоматизації бізнес-процесів у коворкінг-просторі, починаючи з попереднього бронювання робочого місця або конференц-зали, і завершуючи аналітикою щодо завантаженості, доходів та активності користувачів. Однією з ключових переваг платформи є модульний підхід до архітектури, що дозволяє підключати або відключати окремі функціональні блоки залежно від потреб бізнесу. Зокрема, Nexodus надає такі модулі:

- Space Management – управління фізичним простором через візуальні мапи поверхів, інтерактивне бронювання місць, налаштування типів ресурсів (робочі столи, кімнати для зустрічей, обладнання).
- CRM-система – повноцінна система управління взаємодією з клієнтами, що включає сегментацію користувачів, трекінг активності, історію комунікацій та автоматизацію маркетингових кампаній.
- Billing & Invoicing – автоматизоване виставлення рахунків, підтримка різних платіжних моделей (щомісячна підписка, погодинна оплата, pay-as-you-go), генерація інвойсів, інтеграція з платіжними системами (Stripe, PayPal, GoCardless, Mollie).
- Community Engagement Tools – інструменти побудови спільноти: внутрішні новини, форуми, календар подій, система бронювання участі у заходах.
- Analytics & Reporting – розширена система звітності, яка дозволяє аналізувати фінансові показники, відвідуваність, використання простору, тривалість бронювань, продуктивність маркетингових кампаній.
- Help Desk & Support – інструменти технічної підтримки, створення запитів, облік інцидентів та SLA.
- Access Control & IoT Integration – підтримка інтеграції з пристроями контролю доступу (Salto KS, Kisi), сенсорами присутності, а також системами освітлення та клімат-контролю.

Крім того, Nexodus активно підтримує інтеграцію з зовнішніми сервісами, серед яких Google Workspace, Microsoft 365, Slack, Zapier, Xero, QuickBooks, а також RESTful API, що дозволяє розширювати функціональність за допомогою власних модулів. Також надається SDK для розробників.

Окрему увагу розробники приділили мобільній взаємодії. Система має два спеціалізовані мобільні застосунки: один для адміністраторів простору (Nexodus Admin Panel), інший – для користувачів (Passport by Nexodus). Через мобільний інтерфейс користувачі можуть бронювати місця, оплачувати послуги, отримувати

сповіщення про події та новини, а також взаємодіяти з іншими учасниками спільноти. На рисунках 3.1 – 3.5 представлено деякі функціональні можливості мобільного застосунку Passport by Nexudus.

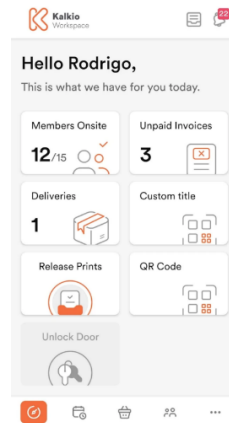


Рисунок 2.1 – Відображення головного меню вибраного воркспейсу користувача в Passport by Nexudus

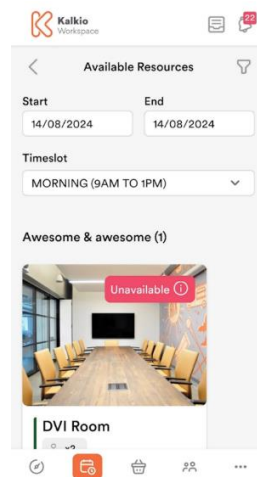


Рисунок 2.2 – Перегляд доступних приміщень у вибраному воркспейсі в Passport by Nexudus



Рисунок 2.3 – Перегляд планування приміщень у вибраному воркспейсі в Passport by Nexodus

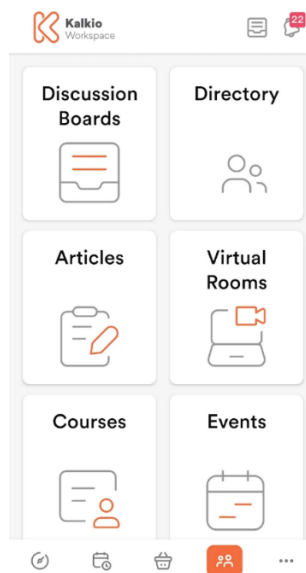


Рисунок 2.4 – Відображення подій, що відбуваються у вибраному воркспейсі в Passport by Nexodus

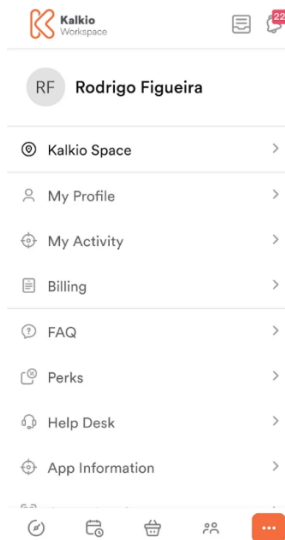


Рисунок 2.5 – Відображення головного меню для користувача в Passport by Nexodus

З точки зору безпеки, Nexodus відповідає сучасним стандартам: реалізовано багаторівневу автентифікацію, шифрування даних при передачі (SSL/TLS), резервне копіювання, обмеження прав доступу та журналювання дій користувачів. Система також підтримує відповідність вимогам GDPR щодо обробки персональних даних.

Важливим аспектом платформи є можливість масштабування: Nexodus придатна для впровадження як у невеликих локальних просторах, так і у міжнародних мережах з сотнями локацій. При цьому зберігається централізоване управління політиками, тарифами та брендуванням.

2.2 OfficeRnD

OfficeRnD є спеціалізованою хмарною платформою, орієнтованою на автоматизацію операційного управління коворкінгами, бізнес-центрами та гнучкими офісними просторами [28]. Заснована у 2015 році, компанія швидко зайняла провідні позиції на світовому ринку завдяки комплексному підходу до цифровізації повсякденних процесів у сфері спільної роботи. Станом на 2025 рік OfficeRnD використовується у понад 60 країнах, обслуговуючи тисячі локацій та десятки тисяч користувачів.

Однією з основних концептуальних особливостей OfficeRnD є орієнтація на користувацьке середовище, що виявляється у широкому спектрі інструментів для підтримки взаємодії користувачів, спрощення комунікації, створення спільнот та забезпечення прозорості адміністративних процесів.

Функціональні можливості системи включають:

- Управління простором (Space & Resource Management) – забезпечується точне планування й адміністрування ресурсів: робочих місць, переговорних кімнат, подієвих зон, паркувальних місць. Інтерфейс дає змогу інтерактивно керувати доступністю та конфігурацією ресурсів, налаштовуючи їх відповідно до бізнес-логіки конкретної організації.
- Фінансовий менеджмент (Billing & Payments) – платформа автоматизує процеси генерації рахунків, обліку платежів, розрахунку вартості послуг, підтримує повторювану виставку інвойсів, інтеграцію з платіжними шлюзами (Stripe, GoCardless, Authorize.Net). Також реалізовано облік ПДВ, знижок та промо-кодів.
- Управління клієнтами (CRM) – система дозволяє структурувати інформацію про користувачів, трекати життєвий цикл клієнтів, сегментувати базу та інтегрувати маркетингові дії з інструментами аналітики.
- Аналітика та звітність (Analytics & Dashboards) – інструменти моніторингу завантаженості, фінансових потоків, конверсій, використання послуг. Передбачено кастомізацію метрик, візуалізацію даних у вигляді діаграм та графіків, а також експорт до Excel/CSV.
- Користувацький портал та мобільні застосунки (Member Experience) – надається персоналізований веб-портал і мобільні застосунки для Android/iOS, через які користувачі можуть бронювати ресурси, переглядати рахунки, брати участь у заходах, спілкуватися з іншими членами спільноти.

Це підвищує рівень самостійності користувачів і зменшує навантаження на адміністрацію.

- Інтеграції з IT-інфраструктурою – підтримуються сторонні сервіси, зокрема Google Calendar, Outlook, HubSpot, Mailchimp, QuickBooks, Xero, Kisi (контроль доступу), а також API для інтеграції з власними системами.

Система OfficeRnD вирізняється інтуїтивно зрозумілим інтерфейсом, високою швидкістю реагування та масштабованістю, що робить її придатною для використання як у невеликих просторах на кілька десятків місць, так і в мережових структурах з розгалуженою інфраструктурою. Для адміністраторів передбачено централізовану консоль керування, де можна регулювати політики доступу, тарифи, правила бронювання тощо.

На рисунках 2.6 – 2.9 представлено деякі функціональні можливості платформи OfficeRnD.

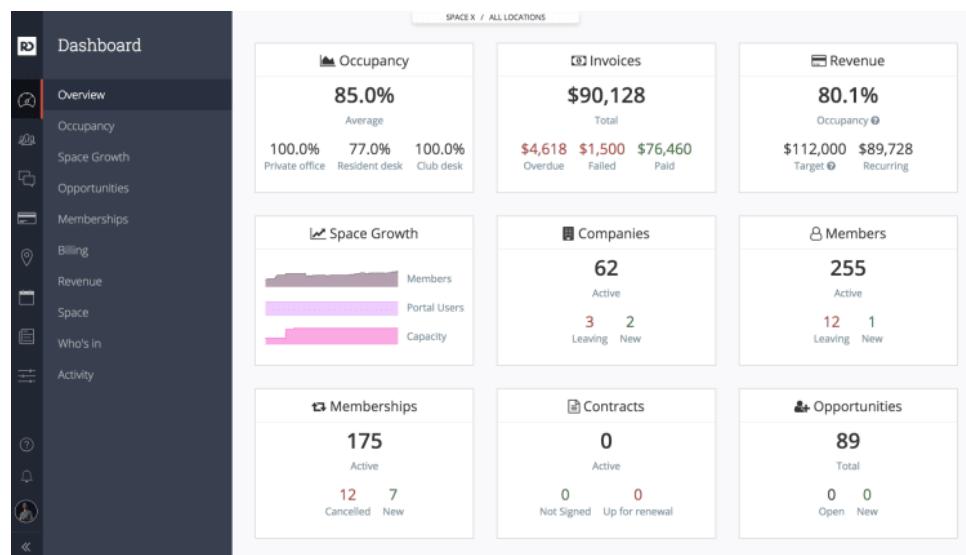


Рисунок 2.6 – Головне меню користувача в OfficeRnD



Рисунок 2.7 – Функція управління простором в OfficeRnD

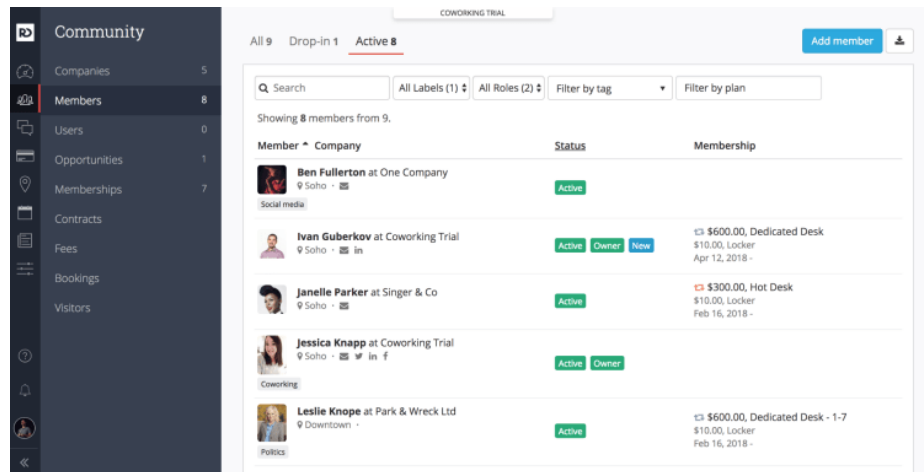


Рисунок 2.8 – Відображення меню управління коворкінгом в OfficeRnD

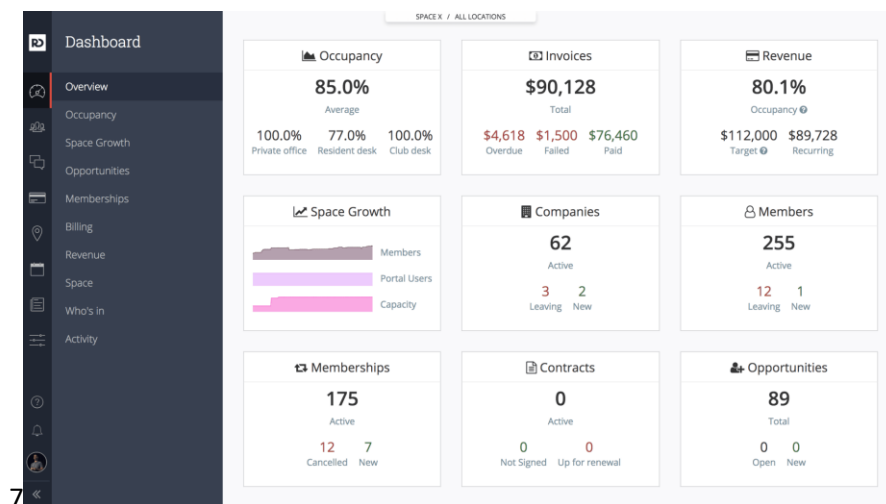


Рисунок 2.9 – Відображення аналітичних дашбордів в OfficeRnD

OfficeRnD також забезпечує відповідність вимогам міжнародного законодавства щодо захисту персональних даних, зокрема GDPR, SOC 2 та ISO/IEC 27001. Реалізовані механізми шифрування даних, розмежування прав доступу, багатофакторна автентифікація та аудит дій користувачів.

На відміну від деяких конкурентів, OfficeRnD активно розвиває інфраструктуру підтримки клієнтів, включаючи навчальні матеріали, консультаційні сесії, базу знань та спеціалізовані онбординг-програми для нових користувачів. Це сприяє швидшій адаптації та ефективнішому впровадженню рішення.

Таким чином, OfficeRnD можна охарактеризувати як високоефективну, гнучку та масштабовану систему, здатну забезпечити повноцінну цифрову трансформацію управління гнучкими робочими просторами. Її використання дозволяє суттєво знизити операційні витрати, покращити клієнтський досвід і підвищити продуктивність адміністрування.

2.3 Coworkify

Coworkify є одним із хмарних програмних рішень, орієнтованих на адміністрування коворкінг-просторів, бізнес-інкубаторів, спільних офісів та інших гнучких форматів комерційної нерухомості [29]. Основною метою платформи є спрощення повсякденних операцій, автоматизація взаємодії з клієнтами та підвищення ефективності управління простором.

На відміну від масштабніших платформ, Coworkify фокусується на мінімалістичному, але функціонально завершеному підході. Її архітектура орієнтована на базові, проте ключові аспекти операційного менеджменту, які забезпечують оптимальне співвідношення ціни, простоти використання та функціональної повноти.

Основні функціональні можливості Coworkify включають:

- Управління резерваціями (Bookings & Scheduling) – система забезпечує простий механізм для бронювання робочих місць, кімнат для переговорів та інших ресурсів.
- Управління користувачами (Member Management) – адміністратори мають змогу створювати профілі користувачів, надавати їм доступ до різних рівнів послуг, контролювати їхню активність та комунікувати безпосередньо через внутрішню панель.
- Звітність та аналітика (Reports & Insights) – передбачені базові звіти щодо використання ресурсів, прибутків, активності клієнтів та статистики бронювань.
- Онлайн-портал для клієнтів (Client Self-Service Portal) – користувачі можуть самостійно керувати своїми бронюваннями, переглядати рахунки, редагувати персональні дані та змінювати тарифні плани без необхідності звернення до адміністратора.

Coworkify є мультиплатформною системою з адаптивним веб-інтерфейсом, який дозволяє працювати на ПК, планшетах і смартфонах без необхідності встановлення окремих додатків. Це зменшує вартість технічного обслуговування та полегшує доступ до системи для користувачів з різним рівнем цифрової грамотності.

На рисунках 2.10 – 2.11 представлено деяку функціональні можливості Coworkify.

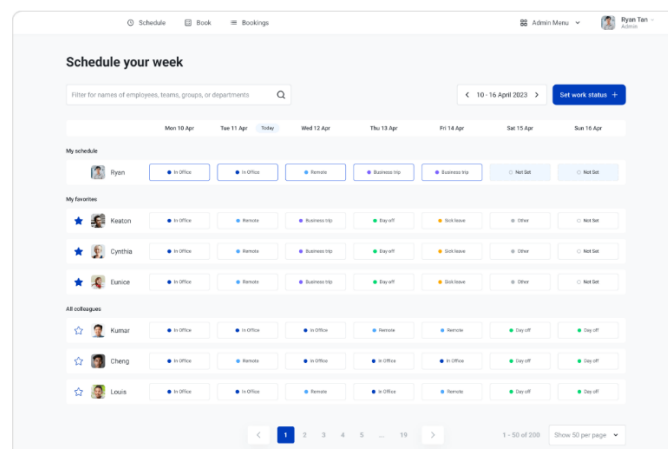


Рисунок 2.10 – Відображення меню поточних бронювань в Coworkify



Рисунок 2.11 – Відображення аналітичних дашбордів в Coworkify
Серед ключових переваг Coworkify:

- Швидкість впровадження – система не вимагає складної конфігурації або участі технічних спеціалістів, що дозволяє розпочати роботу практично «з коробки».
- Легка масштабованість – рішення легко адаптується до змін у структурі простору, кількості клієнтів або тарифній політиці.
- Мінімальне навчання персоналу – інтерфейс платформи інтуїтивно зрозумілий, що дозволяє скоротити час на онбординг та мінімізує кількість помилок в операційній діяльності.

Однак у порівнянні з більш просунутими платформами, такими як Nexodus або OfficeRnD, Coworkify має обмежену кількість інтеграцій з сторонніми сервісами, не підтримує розширену CRM-функціональність та менш придатна для великих мережових операторів.

Незважаючи на це, Coworkify є економічно доцільним рішенням для невеликих коворкінгів, які прагнуть мінімізувати витрати на ІТ-інфраструктуру, зберігаючи при цьому контроль над основними бізнес-процесами. 7

2.4 Cobot

Cobot є одним із провідних SaaS-рішень для управління коворкінг-просторами, що функціонує на ринку з 2010 року та обслуговує понад 1000 коворкінгів у більш ніж 80 країнах. Система розроблена з метою автоматизації ключових бізнес-процесів, зменшення адміністративного навантаження та забезпечення масштабованості для коворкінгів різних типів – від невеликих спільнот до розгалужених мереж із численними локаціями [30].

На відміну від мінімалістичних систем, таких як Coworkify, платформа Cobot пропонує гнучку архітектуру з широким спектром інтеграцій, що дозволяє адаптувати її до специфіки кожного простору.

Ключові функціональні можливості системи включають:

- Модуль управління користувачами (Membership Management) – забезпечує централізовану базу даних користувачів з можливістю призначення ролей, налаштування тарифних планів, відстеження активності та ведення історії взаємодії.
- Бронювання ресурсів (Resource Booking System) – дозволяє створювати кастомізовані правила для бронювання кімнат, столів, техніки або спеціалізованих зон. Інтеграція з календарями (Google Calendar, Outlook) забезпечує синхронізацію подій у реальному часі.
- Автоматизований білінг та платежі (Automated Billing & Payments) – підтримується генерація рахунків-фактур, облік послуг на основі передплати або погодинного використання, автоматичне списання коштів через Stripe, GoCardless або PayPal.
- Звіти та аналітика (Analytics & Reporting) – передбачено глибоку аналітику за типами використання, доходами, завантаженістю ресурсів, рівнем відтоку клієнтів тощо. Дані можуть експортуватися у форматах CSV або оброблятися через інтегровані панелі моніторингу.

- Сторонні інтеграції (Third-party Integrations) – підтримуються інтеграції з Slack, Mailchimp, Zapier, Kisi (для контролю доступу), Херо та іншими системами, що дозволяє розширити функціонал платформи без додаткової розробки.

Cobot також забезпечує багатомовну підтримку, адаптуючи інтерфейс відповідно до мовних уподобань користувача, що робить платформу придатною для міжнародного використання. Крім того, передбачено модуль для управління кількома локаціями одночасно, що є критично важливим для операторів мережеских просторів.

На рисунках 2.12 – 2.15 представлено деякі функціональні можливості платформи Cobot.

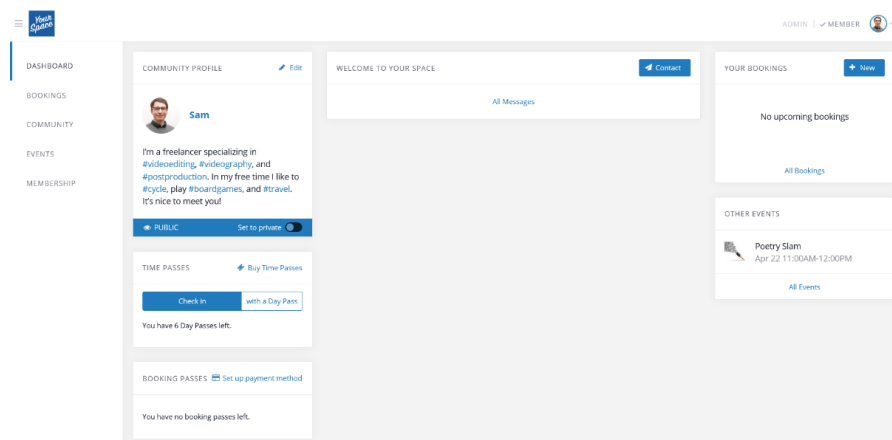


Рисунок 2.12 – Відображення профілю користувача в Cobot

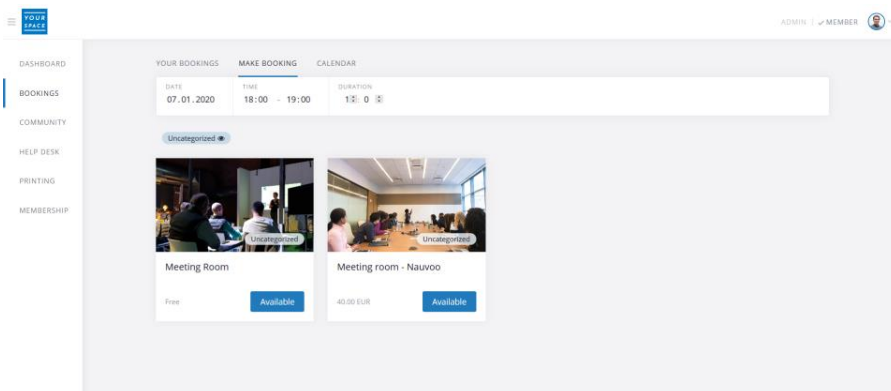


Рисунок 2.13 – Відображення доступних приміщень в Cobot

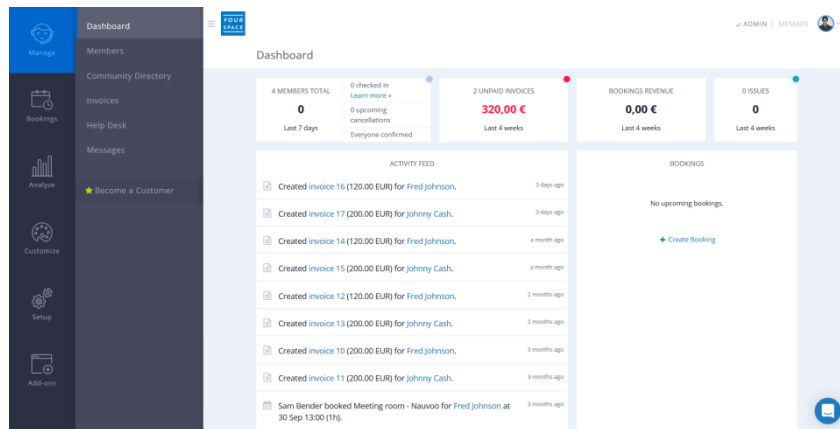


Рисунок 2.14 – Відображення надходжень за оплату оренди в Cobot

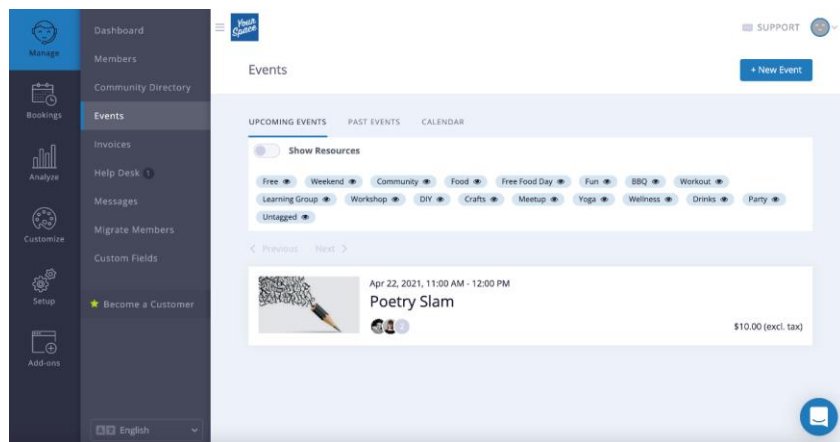


Рисунок 2.15 – Відображення списку подій в Cobot

Суттєвими перевагами Cobot є:

- Масштабованість та гнучкість – підходить як для невеликих коворкінгів, так і для великих бізнес-центрів.
- Прозора модель ціноутворення – залежність вартості від кількості активних користувачів дозволяє бізнесу ефективно планувати витрати.
- API-доступ – відкритий інтерфейс прикладного програмування дозволяє реалізувати кастомні сценарії інтеграції з внутрішніми або сторонніми системами.

- Активна спільнота користувачів та підтримка – компанія підтримує знання-базу, проводить регулярні оновлення та розвиває екосистему через відгуки користувачів.

Водночас, слід зауважити, що у порівнянні з такими рішеннями, як OfficeRnD або Nexodus, Cobot надає дещо обмежені можливості в частині глибокої CRM-аналітики або управління подіями. Проте ці обмеження компенсуються стабільністю роботи, простотою адаптації та орієнтацією на функціональну достатність для більшості бізнес-завдань.

Таким чином, Cobot є збалансованим рішенням для операторів коворкінг-просторів, які прагнуть автоматизувати ключові бізнес-процеси без суттєвих витрат на впровадження складних ERP-систем, зберігаючи при цьому гнучкість і контроль над внутрішніми операціями.

2.5 Satellite Deskworks

Satellite Deskworks є хмарним програмним забезпеченням, орієнтованим на автоматизацію операційного управління коворкінгами, бізнес-центрами, гнучкими офісами та спільними робочими просторами. Система розроблена компанією Satellite Centers у США, яка має багаторічний досвід створення та адміністрування коворкінг-просторів. Таким чином, програмний продукт базується на практичному розумінні специфіки управління такими об'єктами [31].

Ключові функціональні можливості Deskworks включають:

- Інтелектуальний облік часу перебування (Time Tracking & Access Monitoring): платформа автоматично відстежує фізичне перебування користувачів за допомогою Wi-Fi, сенсорів руху або інтеграцій з системами контролю доступу, що дозволяє точно обліковувати використання ресурсів.
- Резервування ресурсів: реалізовано гнучкий механізм бронювання кімнат, конференц-залів або інших об'єктів через онлайн-портал. Система дозволяє

уникати дублювання записів і передбачає індивідуальні правила доступу для різних категорій користувачів.

- Портал самообслуговування: користувачі можуть самостійно переглядати свою активність, змінювати підписки, бронювати кімнати, оплачувати рахунки та комунікувати з адміністрацією.
- Мобільна сумісність: хоча окремого мобільного застосунку немає, веб-інтерфейс Deskworks оптимізований для використання зі смартфонів і планшетів.

На рисунку 2.16 – 2.19 представлено деякі функціональні можливості Satellite Deskworks.

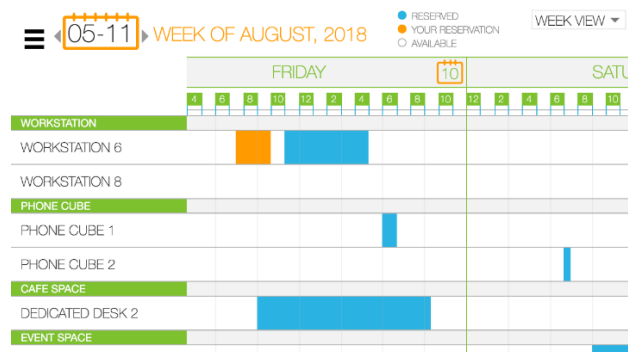


Рисунок 2.16 – Відображення вільних для бронювання робочих місць та приміщень в Satellite Deskworks

PRIVATE OFFICE, PER HOUR

Pass Balance: 20.0

Date	Duration of reservation (hours/passes)	Duration of usage (hours/passes)	Billing (amount/passes)
Aug 10, 2018	2.5 / 0.0	- / -	\$0.00 / 0
Aug 2, 2018	12.0 / 0.0	6.5 / 6.5	\$0.00 / 0

Reservation Log

From	To	Duration (hours)	Duration (passes)	Amount
02:00 ^{AM}	02:00 ^{PM}	12	12.0	\$216.00

Usage Log

From	To	Duration (hours)	Duration (passes)	Amount
09:00 ^{AM}	11:00 ^{AM}	2.0	2.0	\$36.00
02:00 ^{PM}	06:30 ^{PM}	4.5	4.5	\$81.00

Jul 26, 2018	2.0 / 2.0	0.03 / 0.0	\$36.00 / 2.0
Jul 24, 2018	- / 0.0	7.08 / 7.25	\$130.50 / 7.25

Рисунок 2.17 – Підсумування вартості робочого місця в Satellite Deskworks

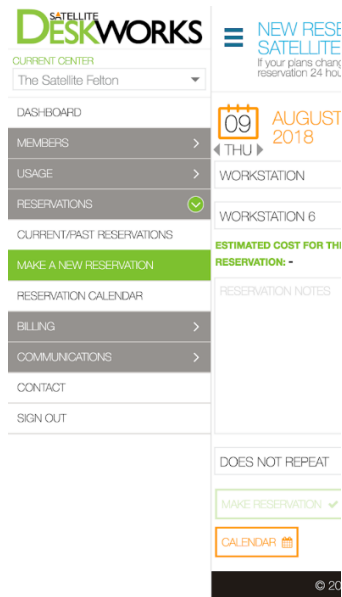


Рисунок 2.18 – Відображення меню користувача в Satellite Deskworks

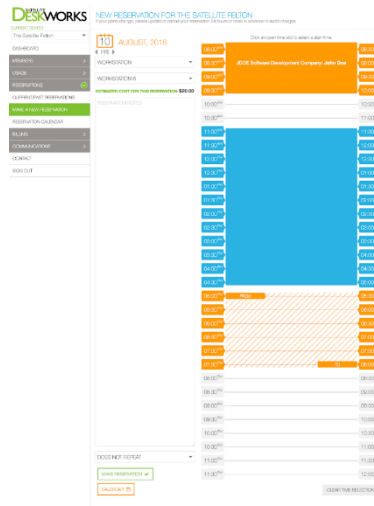


Рисунок 2.19 – Відображення завантаженості робочого місця в Satellite Deskworks

Особливою перевагою Satellite Deskworks є інтелектуальна система обліку перебування, яка дозволяє мінімізувати ручне введення даних і підвищує точність розрахунків за фактичне використання. Це є критично важливим у випадках, коли оплата здійснюється за реально проведений час, а не за фіксованим тарифом.

Satellite Deskworks є функціонально збалансованим інструментом, орієнтованим на практичні потреби адміністраторів коворкінгів.

Отже, у процесі дослідження було проаналізовано п'ять сучасних програмних продуктів, що орієнтовані на управління коворкінг-просторами, кожен із яких має свої особливості, функціональну спеціалізацію та переваги для конкретних сценаріїв використання. Системи Nexodus, OfficeRnD, Coworkify, Cobot і Satellite Deskworks демонструють різні підходи до організації управління простором, обслуговування клієнтів, обліку ресурсів та автоматизації адміністративних процесів. Nexodus вирізняється найширшим спектром можливостей, орієнтованим на великі простори з високим рівнем інтеграції, тоді як OfficeRnD пропонує зручну платформу з сильним акцентом на UX та звітність. Coworkify та Cobot підходять для малих і середніх коворкінгів, які не потребують складної конфігурації, натомість прагнуть до простоти у щоденних операціях. Satellite Deskworks пропонує гнучкий облік використання ресурсів у поєднанні з ефективною логікою білінгу, що робить його конкурентоспроможним рішенням у своєму сегменті. Порівняння основних характеристик розглянутих систем управління представлено в таблиці 2.1.

Таблиця 2.1 – Порівняння основних характеристик розглянутих систем управління

Характеристика	Nexodus	OfficeRnD	Coworkify	Cobot	Satellite Deskworks
Орієнтація	Середні та великі простори	Малі та середні простори	Малі коворкінги	Малі коворкінги	Гнучкі офіси, бізнес-центри
CRM-функції	Розширені	Розвинені	Базові	Базові	Обмежені
Бронювання ресурсів	Так	Так	Так	Так	Так
Інтеграції	Велика кількість (Slack, Zapier, QuickBooks, і т.д.)	Сильні (Google, Microsoft 365, Slack, тощо)	Мінімальні	Середній рівень	Обмежені (основні платіжні шлюзи)

Таблиця 2.1 – Продовження.

Автоматичний білінг	Так	Так	Так	Так	Так
Система доступу/обліку перебування	Частково	Частково	Ні	Ні	Так (Wi-Fi, сенсори)
Звітування та аналітика	Потужна	Висока деталізація	Базова	Базова	Середній рівень
Підтримка багатьох локацій	Так	Так	Ні	Обмежено	Так
Простота у використанні	Середня	Висока	Висока	Висока	Середня
Мобільний застосунок	Так	Так	Ні	Так	Ні (адаптивний веб-інтерфейс)

Підсумовуючи проведене дослідження функціональних можливостей, можна зробити висновок, що наявні програмні рішення мають широкий спектр функцій, здатних задовольнити потреби як невеликих локальних просторів, так і великих мережевих операторів коворкінгів. Однак жодна з платформ не є універсальною, кожна система орієнтована на свою цільову аудиторію та вирішує лише частину задач, які можуть виникати в рамках експлуатації сучасного коворкінгу.

3 ПРОЕКТУВАННЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ОРЕНДОЮ РОБОЧИХ МІСЦЬ У КОВОРКІНГАХ

3.1 Уточнена постановка задачі та визначення ключових функцій системи

На основі функціонального аналізу аналогічних рішень і результатів дослідження, сформульовано уточнену постановку задачі. Метою є створення автоматизованої системи, яка дозволяє адміністратору керувати ресурсами коворкінгу, а користувачам – здійснювати швидкий пошук, бронювання, оцінювання й перегляд доступних об'єктів оренди.

Основні функції системи згруповано за функціональними модулями:

- модуль автентифікації: реєстрація, авторизація, керування профілем користувача.
- модуль керування ресурсами: додавання, редагування та деактивація об'єктів оренди (переговорних кімнат, робочих місць).
- модуль візуалізації доступності (інтерфейс календаря).
- модуль оцінювання та відгуків: система рейтингу ресурсів і коментарів.
- адміністративна панель: аналітика, перегляд статистики, контроль активності користувачів.

Функціональність має бути реалізована як у фронтенд-інтерфейсі, так і через взаємодію з серверною частиною, що обробляє дані з бази даних, забезпечує авторизацію, логіку бронювання та фільтрацію доступних ресурсів у реальному часі.

3.2 Розробка структури бази даних

База даних виступає центральним елементом, що забезпечує збереження, доступ, цілісність і узгодженість інформації між усіма модулями системи. Для ефективного проектування структури бази даних було використано методологію побудови ER-діаграми (діаграми “сутність-зв'язок”), яка дозволяє візуалізувати основні сутності системи, їх атрибути, первинні ключі та логічні зв'язки між

таблицями [32-34]. Це дає змогу уникнути надмірного дублювання даних, забезпечити логічну нормалізацію таблиць і полегшити подальшу реалізацію на рівні реляційної СУБД PostgreSQL.

У системі управління орендою робочих місць у коворкінгах виділено основні сутності, які безпосередньо формують інформаційний каркас: Користувачі, Ресурси, Бронювання, Відгуки, Оцінки, Зображення, Повідомлення. Всі об'єкти пов'язані між собою логічними зв'язками один-до-багатьох або багато-до-одного. ER-діаграма автоматизованої системи управління орендою робочих місць у коворкінгах представлена на рисунку 3.1.

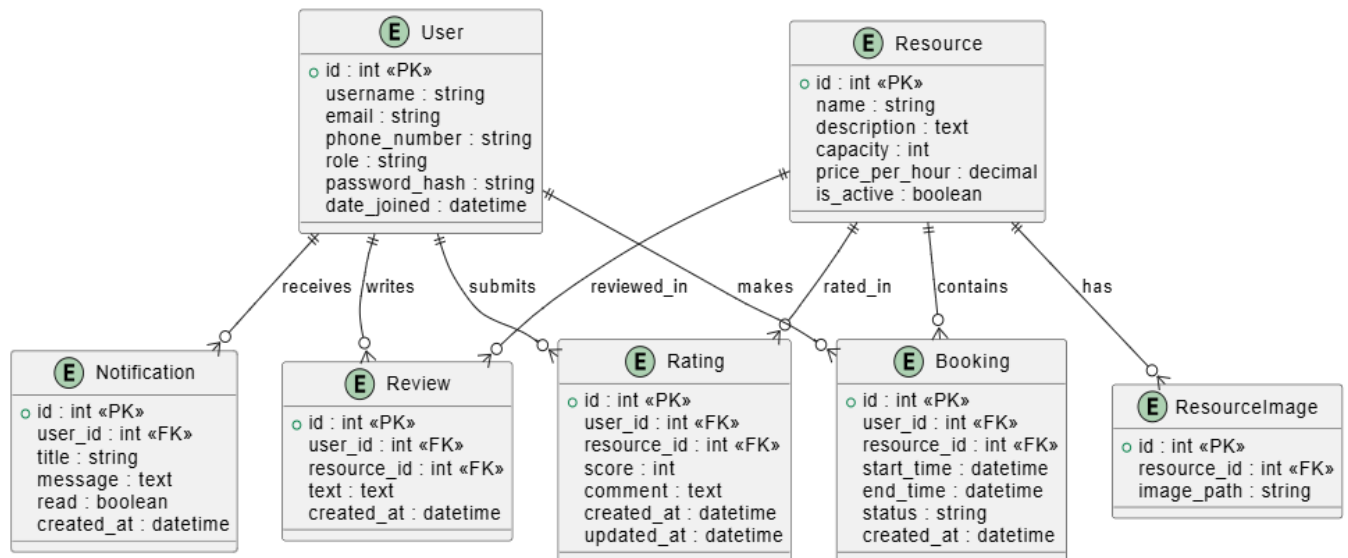


Рисунок 3.1 – ER-діаграма автоматизованої системи управління орендою робочих місць у коворкінгах

Детальний опис сутностей та їх атрибутів описано в таблиці 3.1.

Таблиця 3.1 – Опис сутностей ER-діаграми та атрибутів сутностей

№	Сутність	Атрибути	Призначення
1	User	id (PK), username, email, phone_number, role, password_hash, date_joined	Зберігає дані облікових записів користувачів
2	Resource	id (PK), name, description, capacity, price_per_hour, is_active	Представляє об'єкти оренди (робочі місця, переговорні кімнати)
3	Booking	id (PK), user_id (FK), resource_id (FK), start_time, end_time, status, created_at	Містить інформацію про бронювання ресурсів користувачами
4	Review	id (PK), user_id (FK), resource_id (FK), text, created_at	Зберігає текстові відгуки користувачів про ресурси
5	Rating	id (PK), user_id (FK), resource_id (FK), score, comment, created_at, updated_at	Зберігає кількісну оцінку ресурсу з можливим коментарем
6	Notification	id (PK), user_id (FK), title, message, read, created_at	Містить повідомлення системи для конкретного користувача
7	ResourceImage	id (PK), resource_id (FK), image_path	Містить шляхи до зображень, прикріплених до ресурсу

Логічна структура бази даних системи управління орендою робочих місць побудована з дотриманням принципів нормалізації та модульного розподілу. Центральною сутністю є таблиця User, що містить облікові дані користувачів, включаючи дату реєстрації. З цією таблицею пов'язані сутності Booking, Review, Rating та Notification, які реалізують персоналізовану взаємодію з системою – здійснення бронювання, залишення оцінок та відгуків, а також отримання повідомлень.

Сутність Resource репрезентує робочі простори, доступні для оренди. З кожним ресурсом можуть бути пов'язані кілька бронювань, відгуків, оцінок і зображень. Це дозволяє реалізувати повноцінну візуалізацію об'єкта в інтерфейсі користувача та підтримати логіку зворотного зв'язку.

Таблиця Booking виконує роль транзакційного журналу, фіксуючи, хто, коли і який ресурс орендує, із зазначенням статусу (наприклад, активне, скасоване) та дати створення запису. Це дозволяє вести хронологію подій і здійснювати аналітику на рівні часу.

Review та Rating утворюють підсистему користувацького оцінювання. Вони забезпечують не лише можливість залишати якісні й кількісні оцінки, але й переглядати історію змін оцінювання, завдяки наявності поля updated_at. Таким чином формується репутаційна модель ресурсів.

Сутність Notification призначена для обміну системними повідомленнями, що мають заголовок, основний текст, позначку про прочитання та часову мітку. Це дозволяє організувати ефективну інформаційну взаємодію з користувачем у важливих моментах – наприклад, при підтвердженні або скасуванні бронювання.

Нарешті, таблиця ResourceImage дозволяє кожному ресурсу мати декілька зображень, що критично важливо для формування привабливого користувацького інтерфейсу з візуальними елементами.

Запропонована структура охоплює всі необхідні аспекти зберігання та обробки даних у системі й є готовою до реалізації в середовищі реляційної СУБД PostgreSQL.

3.3 Розробка UML-діаграм

UML-діаграми є стандартним інструментом для моделювання структури та логіки програмних систем. Їх використання дозволяє чітко визначити зв'язки між об'єктами, сценарії взаємодії користувачів із системою та послідовність виконання основних процесів.

На етапі проєктування UML-діаграми допомагають уникнути логічних помилок і уточнити вимоги до системи [35].

На етапі тестування UML-діаграми служать важливою основою для формування сценаріїв тестування, побудови чек-листів і валідації очікуваної поведінки системи у відповідь на дії користувачів. Зокрема, діаграми варіантів використання (Use Case) дозволяють перевірити покриття функціональних вимог, а діаграми активностей – виявити граничні стани та можливі помилки в логіці переходів.

У період технічної підтримки та подальшого розширення системи UML-діаграми є джерелом актуальної структурованої інформації, що дозволяє швидко орієнтуватися у складній архітектурі, прогнозувати вплив змін, уникати помилок при інтеграції нових функцій та забезпечувати сталий розвиток системи без порушення вже реалізованої логіки [36-37].

3.3.1 Опис UML-діаграми класів

UML-діаграма класів є ключовим засобом візуального моделювання структури об'єктів інформаційної системи. Вона дозволяє представити класи, їхні атрибути, методи та зв'язки між класами, що суттєво полегшує реалізацію системи, її тестування та супровід. Цей тип діаграми дає змогу проаналізувати архітектуру даних ще до етапу програмування, забезпечити уніфікацію моделей у коді, уникнути логічних помилок у структурі бази даних і спростити розуміння взаємодії між компонентами [38, 39].

У контексті розробленої автоматизованої системи управління орендою робочих місць у коворкінгах, діаграма класів відображає статичну структуру основних логічних сутностей системи. До таких належать класи User, Resource, Booking, Review, Rating, Notification та ResourceImage. Кожен із них має чітко визначений набір атрибутів, а також базові методи для створення, оновлення, перегляду та видалення об'єктів.

Розроблена UML-діаграма класів автоматизованої системи управління орендою робочих місць у коворкінгах представлена на рисунку 3.2.

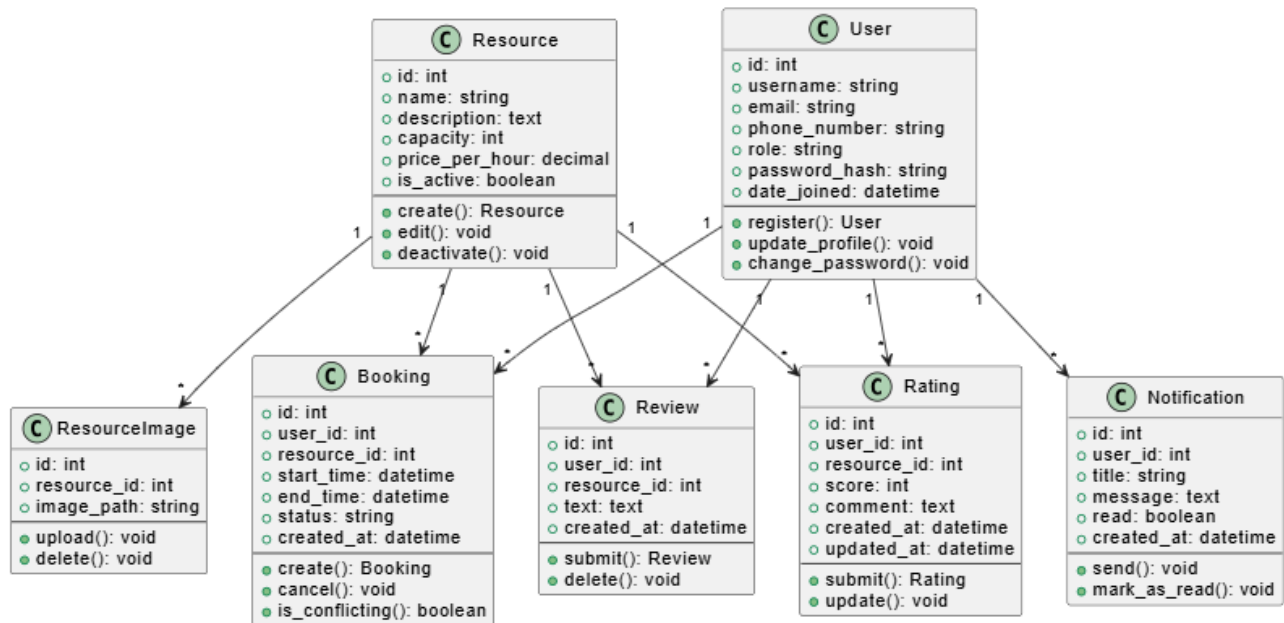


Рисунок 3.2 - UML-діаграма класів автоматизованої системи управління орендою робочих місць у коворкінгах

Клас User містить дані про зареєстрованого користувача, включно з контактною інформацією та роллю. Його методи охоплюють створення облікового запису, оновлення профілю та зміну пароля. Клас Resource відповідає за опис об'єктів оренди – кімнат або робочих місць. Його методи включають додавання, редагування, активацію/деактивацію ресурсів. Клас Booking реалізує логіку створення та обробки запитів на бронювання з перевіркою конфліктів.

Класи Review і Rating відображають механізми зворотного зв'язку, дозволяючи залишати текстові коментарі та кількісні оцінки. Notification відповідає за надсилання повідомлень користувачам, а ResourceImage – за прикріплення зображень до ресурсів.

3.3.2 Опис UML-діаграми варіантів використання

UML-діаграма варіантів використання дозволяє описати зовнішню поведінку системи через взаємодію з користувачами. Вона візуалізує акторів, які ініціюють функціональність, та варіанти використання (сценарії), що відображають реакцію системи. Дана діаграма є зручною для визначення функціональних вимог, планування тестування та комунікації між учасниками розробки [40, 41].

У системі управління орендою робочих місць у коворкінгах ідентифіковано двох основних акторів: Користувач та Адміністратор. Кожен із них взаємодіє з системою в межах свого функціонального рівня доступу. Деякі варіанти використання є спільними (наприклад, перегляд ресурсу або історії бронювань), інші – специфічними для ролі.

На рисунку 3.3 представлено UML-діаграму варіантів використання автоматизованої системи управління орендою робочих місць у коворкінгах з акторами Користувач та Адміністратор.

Варіанти використання для Користувача:

- Авторизуватися – початковий сценарій входу до системи.
 - Підваріанти: «Зареєструватися», «Відновити пароль».
- Переглянути список ресурсів – отримати доступ до актуального переліку доступних об’єктів.
 - Підваріанти: «Застосувати фільтри», «Переглянути зображення», «Переглянути деталі ресурсу».
- Створити бронювання – вибрати часовий інтервал і підтвердити бронювання.
 - Підваріанти: «Перевірити доступність», «Підтвердити бронювання».
- Переглянути власні бронювання – відобразити історію та активні записи.
 - Підваріанти: «Скасувати бронювання».

- Залишити оцінку ресурсу – дати кількісну оцінку.
- Залишити відгук – додати текстовий коментар.
- Отримувати повідомлення – перегляд сповіщень у системі.

Редагувати профіль – змінити контактні дані, пароль.

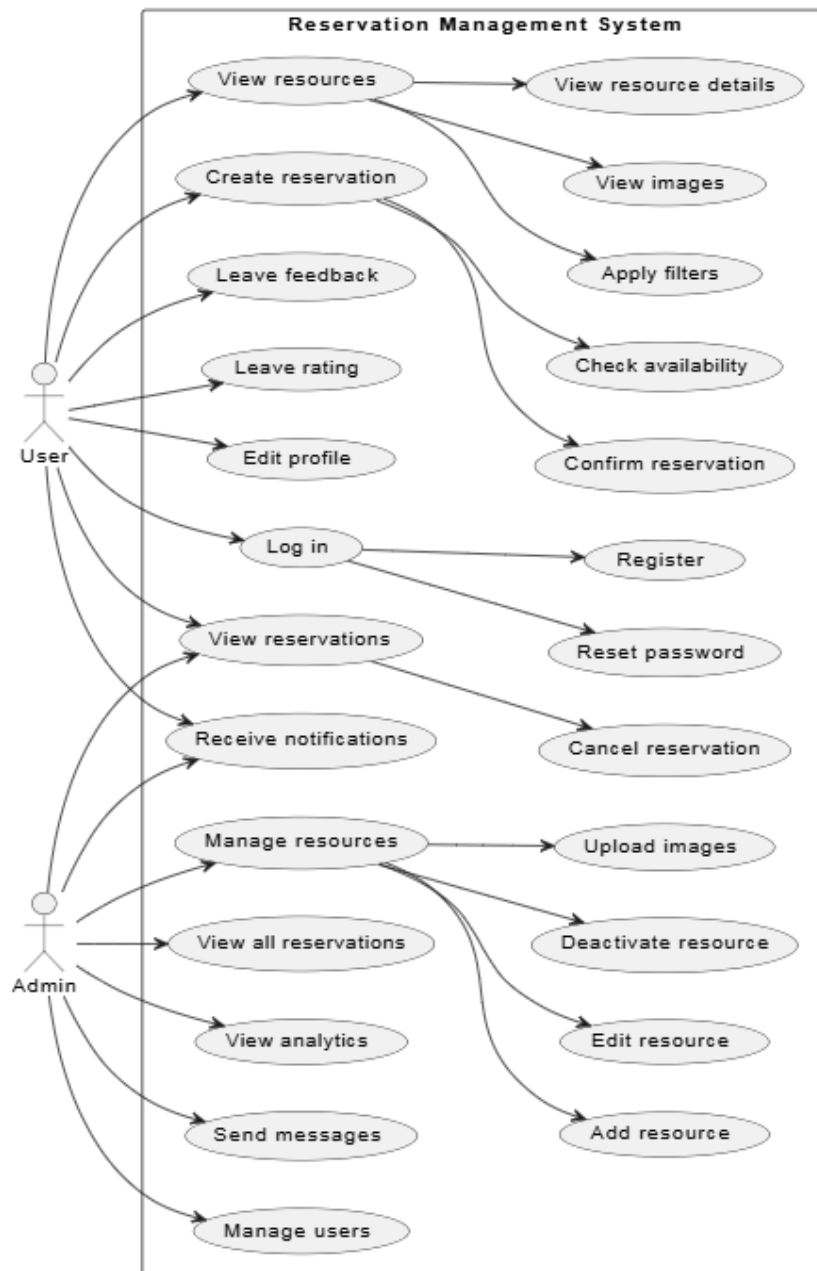


Рисунок 3.3 – UML-діаграма варіантів використання автоматизованої системи управління орендою робочих місць у коворкінгах з акторами Користувач та Адміністратор

Варіанти використання для Адміністратора:

- Керувати ресурсами – повний контроль над додаванням, редагуванням, видаленням об'єктів.
- Підваріанти: «Додати ресурс», «Редагувати ресурс», «Деактивувати ресурс», «Завантажити зображення».
- Переглянути всі бронювання – доступ до бронювань усіх користувачів.
- Надсилати повідомлення – інформування користувачів через систему.
- Переглядати аналітику – статистика бронювань, рейтинги ресурсів.
- Керувати користувачами: призначення ролей, блокування акаунтів.

Варіанти «Переглянути бронювання» та «Отримати повідомлення» мають обох акторів і можуть бути спільними сценаріями використання.

3.3.3 Опис UML-діаграми активностей

UML-діаграма активностей використовується для моделювання алгоритмів і бізнес-процесів у вигляді послідовностей дій. Вона дозволяє візуально описати логіку виконання певного процесу, включаючи гілки умов, паралельні дії та кінцеві стани. Така діаграма є корисною для виявлення неочевидних переходів або помилок у логіці, а також для документування внутрішньої поведінки системи в межах конкретної функції [42-44].

У межах автоматизованої системи управління коворкінгом, однією з ключових функцій є створення бронювання. Саме ця функція обрана для побудови UML-діаграми активностей, що зображена на рисунку 4.4, оскільки вона поєднує взаємодію користувача з системним інтерфейсом, логіку перевірки доступності ресурсу, взаємодію з базою даних та зворотний зв'язок через повідомлення.

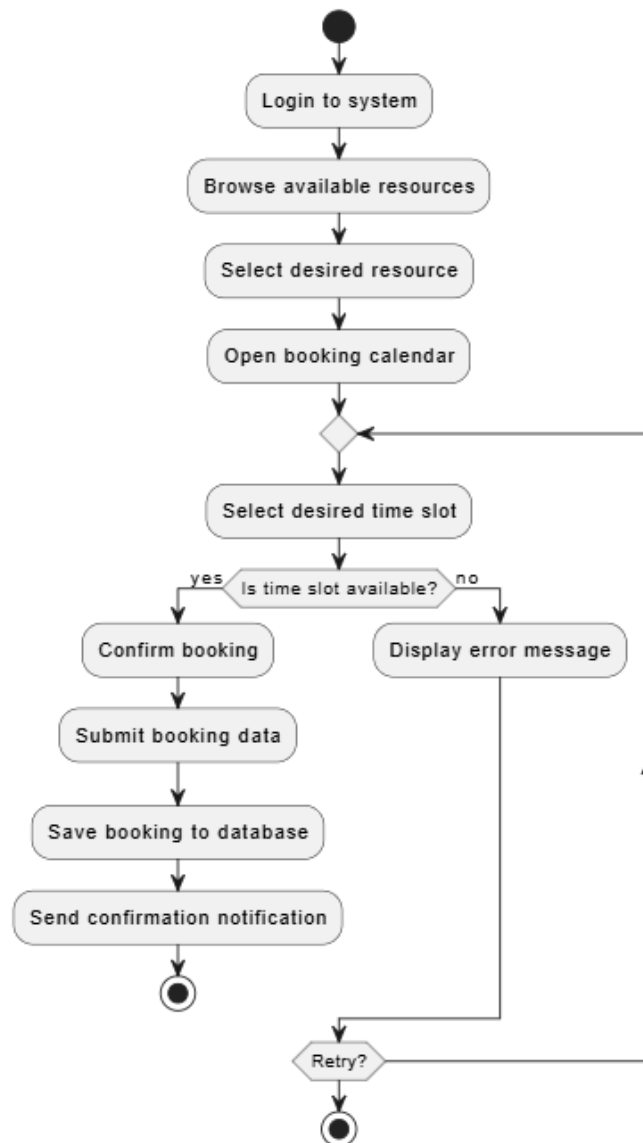


Рисунок 3.4 – Опис функції створення бронювання у вигляді UML-діаграми активностей

Дана діаграма активностей охоплює наступні кроки: авторизацію користувача, перегляд ресурсів, перевірку доступності у календарі, вибір часу, підтвердження умов, створення бронювання в базі та надсилання повідомлення. Якщо ресурс недоступний, користувач отримує відповідне повідомлення і може повернутися до вибору часу.

Ще однією функцією в системі є можливість залишення відгуків. Ця функція доступна лише для користувачів, які вже раніше здійснили бронювання певного ресурсу. Це дозволяє гарантувати, що відгуки надходять від реальних відвідувачів. Діаграма активностей, що зображено на рисунку 4.5, демонструє повну послідовність дій – від авторизації до публікації коментаря.

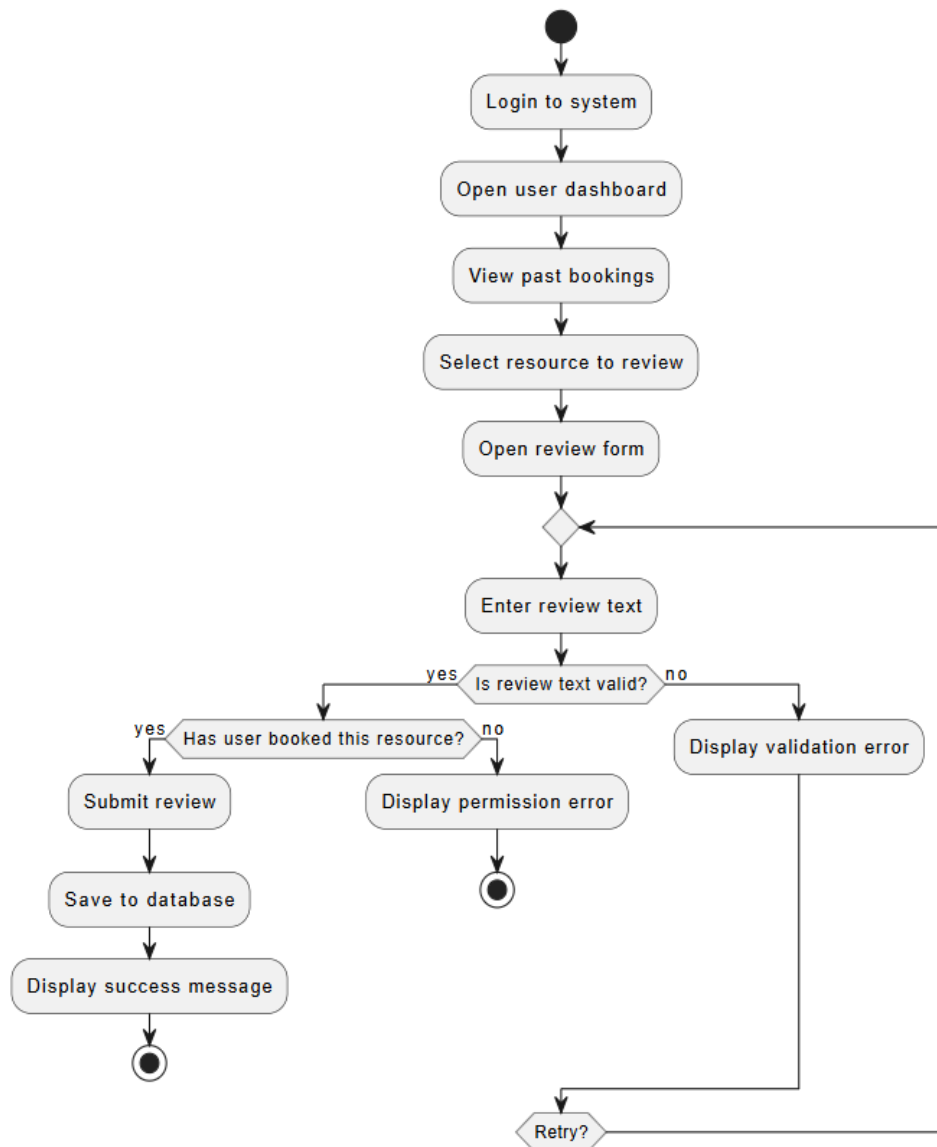


Рисунок 3.5 – Опис функції залишення відгуків у вигляді UML-діаграми активностей

У процесі користувач спочатку обирає ресурс із переліку відвіданих, відкриває форму відгуку, заповнює текст, після чого система виконує перевірку на

заповненість полів та наявність дозволу на залишення відгуку. У разі успішної перевірки коментар зберігається до бази, а користувач отримує підтвердження. У випадку помилки система повідомляє про це і пропонує повторну спробу.

3.4 Вибір архітектури системи

Для реалізації автоматизованої системи управління орендою робочих місць у коворкінгах обрано трирівневу клієнт-серверну архітектуру, яка передбачає розділення системи на рівні представлення (frontend), логіки (backend) та зберігання даних (база даних). Такий підхід є оптимальним для сучасних веборієнтованих сервісів з динамічним контентом, багатокористувацьким доступом та необхідністю гнучкого масштабування.

На рівні клієнта реалізовано графічний інтерфейс, який забезпечує зручну навігацію системою, перегляд ресурсів, керування бронюваннями, а також взаємодію з іншими модулями через веббраузер. Інтерфейс функціонує незалежно від логіки обробки даних і отримує всю інформацію з бекенду через RESTful API, який виконує роль посередника між клієнтом і серверною частиною [45].

Серверна логіка реалізована у вигляді окремих функціональних модулів, які відповідають за обробку запитів від клієнтів, авторизацію, перевірку прав доступу, керування ресурсами, бронюванням, повідомленнями та іншими сутностями. Використання архітектурного патерну MVC (Model–View–Controller), що підтримується фреймворком Django, дозволяє чітко відокремити бізнес-логіку від представлення та моделі даних. Усі зовнішні запити обробляються контролерами, які взаємодіють із моделями та повертають результат у вигляді структурованих відповідей (зокрема, у форматі JSON).

База даних, розміщена окремо від прикладної логіки, зберігає всю інформацію, пов'язану з користувачами, бронюваннями, ресурсами, відгуками та іншими об'єктами. Використання СУБД PostgreSQL дозволяє забезпечити цілісність,

транзакційність і масштабованість, що є критично важливим для зберігання взаємопов'язаних об'єктів у режимі багатокористувацької роботи.

На рисунку 3.6 представлено UML-діаграму компонентів автоматизованої системи управління орендою робочих місць у коворкінгах.

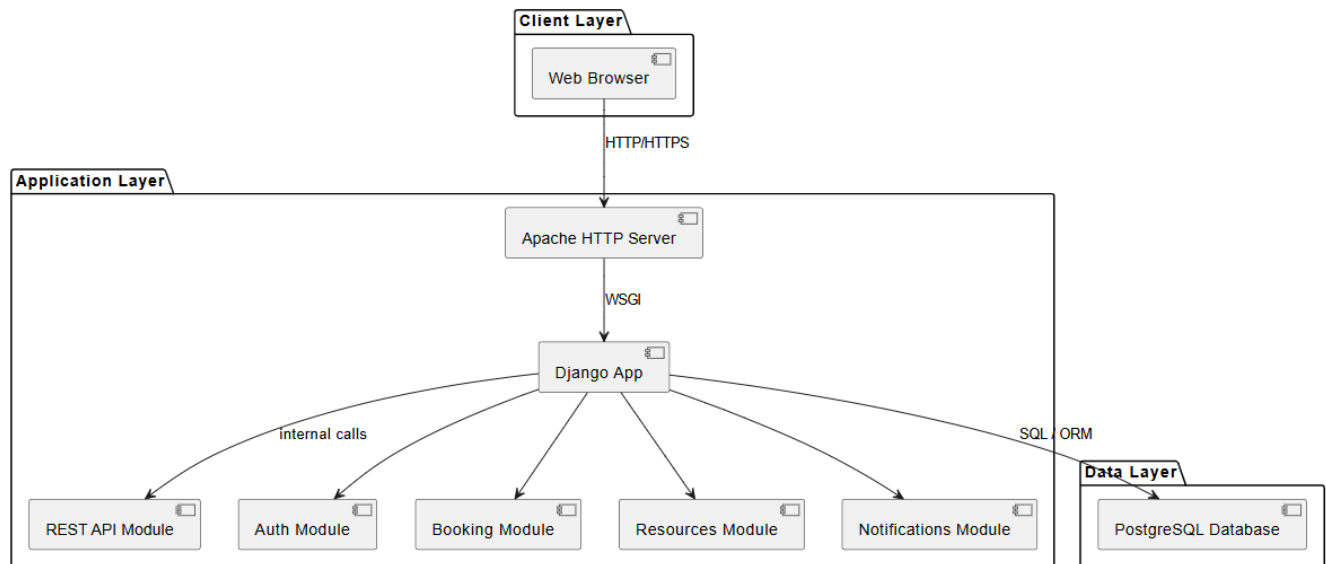


Рисунок 3.6 – UML-діаграма компонентів автоматизованої системи управління орендою робочих місць у коворкінгах

Завдяки модульній структурі серверної частини, система легко розширюється новими функціональними блоками. Кожен модуль (наприклад, auth, booking, resources, notifications) ізольований від інших на рівні внутрішньої реалізації, але взаємодіє через API або загальні сервіси. Це спрощує внесення змін, повторне використання коду та паралельну розробку окремих компонентів [46, 47].

Застосована архітектура дозволяє:

- розділити відповідальність між клієнтом і сервером;
- зменшити навантаження на сервер за рахунок асинхронної взаємодії;
- забезпечити незалежну розробку фронтенду і бекенду;
- підключати зовнішні інтерфейси через існуючий API;
- масштабувати систему горизонтально – шляхом розділення навантаження між кількома серверами або мікросервісами.

У перспективі, обрана архітектура дозволяє інтегрувати нові підсистеми, такі як онлайн-платежі, автоматичні нагадування, синхронізація з Google Calendar тощо, без потреби в глобальній перебудові існуючої логіки. Таким чином, обране архітектурне рішення забезпечує необхідний рівень стабільності, гнучкості та адаптивності до змін, що повністю відповідає функціональним і технічним вимогам розроблюваної системи.

4 РЕАЛІЗАЦІЯ АВТОМАТИЗОВАНОЇ СИСТЕМИ УПРАВЛІННЯ ОРЕНДОЮ РОБОЧИХ МІСЦЬ У КОВОРКІНГАХ

4.1 Вибір технологій для реалізації системи

Вибір технологій для реалізації автоматизованої системи управління орендою робочих місць у коворкінгах є одним з ключових етапів проєктування програмного забезпечення. Коректно підібрана сукупність інструментів, мов програмування, фреймворків та баз даних забезпечує не лише ефективність розробки, а й стабільність, масштабованість та продуктивність функціонування системи у реальному середовищі. Враховуючи цільове призначення системи, вимоги до продуктивності, надійності, безпеки, а також адаптивності інтерфейсу для різних категорій користувачів, було обґрунтовано застосування певного технологічного стеку.

У якості мови програмування для реалізації серверної логіки обрано Python. Це обумовлено не лише популярністю даної мови серед розробників веб-додатків, а й її гнучкістю, великою кількістю доступних бібліотек, високим рівнем читаємості коду та активною підтримкою спільноти. Python має потужну підтримку засобів роботи з базами даних, а також дозволяє швидко реалізовувати бізнес-логіку будь-якої складності. Для реалізації веб-додатку обрано фреймворк Django, який забезпечує комплексну архітектуру розробки за принципом Model–View–Template та сприяє логічному поділу коду на функціональні компоненти та пришвидшує процес розробки [48].

Однією з переваг Django є вбудована ORM (Object-Relational Mapping), що дозволяє працювати з базами даних через об'єктно-орієнтовані моделі. Це дає змогу ефективно реалізувати взаємозв'язки між користувачами, їх бронюваннями, описами робочих місць, цінами, а також системою повідомлень та відгуків. Django також підтримує модулі аутентифікації, адміністрування, кешування, маршрутизації

запитів і має широкий набір засобів безпеки, таких як захист від SQL-ін'єкцій, міжсайтового скриптингу (XSS) та підробки міжсайтових запитів (CSRF).

Для розміщення серверної частини системи обрано Apache HTTP Server, який є одним із найпоширеніших рішень для хостингу веб-додатків. Apache забезпечує високу стабільність, підтримує багато одночасних з'єднань, сумісний з різними операційними системами та має розвинену систему конфігурації. Крім того, він добре інтегрується з Django через використання WSGI (Web Server Gateway Interface), що дозволяє забезпечити надійне обслуговування динамічних запитів до серверної логіки системи.

У якості системи управління базами даних (СУБД) використано PostgreSQL, яка зарекомендувала себе як одна з найпотужніших реляційних баз даних з відкритим вихідним кодом. PostgreSQL забезпечує високу продуктивність, підтримує складні SQL-запити, транзакції, індексацію, перевірку цілісності даних та реплікацію. Це дозволяє ефективно працювати з великими обсягами інформації, зокрема такими як історія бронювань, журнал дій користувачів, аналітичні дані та мультимедійні матеріали, що додаються до опису об'єктів (фотографії, документи тощо). Крім того, PostgreSQL дозволяє масштабувати базу в умовах зростання навантаження без втрати стабільності [49-51].

Для реалізації інтерфейсу користувача (фронтенду) застосовано комбінацію HTML5, CSS3 та JavaScript, що забезпечує стандартну, кросбраузерну і доступну основу для створення сучасних веб-додатків. Для прискорення розробки та підтримки адаптивного дизайну було використано фреймворк Bootstrap, який дозволяє створювати інтерфейси з урахуванням принципів мобільної та адаптивної верстки. Це забезпечує коректне відображення елементів системи як на десктопних, так і на мобільних пристроях, зберігаючи зручність доступу до основних функцій для різних категорій користувачів – адміністраторів, резидентів, гостей.

У контексті інтерактивної взаємодії між клієнтом і сервером використовується мова JavaScript у поєднанні з AJAX для асинхронного обміну даними без

перезавантаження сторінки. Це дозволяє динамічно відображати доступні ресурси, змінювати календар доступності, підтверджувати бронювання або скасування в реальному часі. Також передбачена можливість використання бібліотек типу jQuery для реалізації інтерфейсних елементів (модальних вікон, фільтрів, календарів тощо), що сприяє зручності користування.

Інфраструктура системи дозволяє зберігати критичну інформацію у захищеному вигляді. Аутентифікація користувачів реалізується з використанням стандартних механізмів Django: система хешування паролів, підтримка двофакторної автентифікації, журналювання спроб входу та захист від підбору паролю. Передбачено реалізацію прав доступу до ресурсів відповідно до ролі користувача – адміністратор, зареєстрований клієнт, гість тощо. Усі комунікації з сервером відбуваються через захищене HTTPS-з'єднання, що забезпечує шифрування переданих даних [52].

Крім основної функціональності (створення та перегляд бронювань, перегляд доступності, керування акаунтом), передбачається реалізація додаткових функцій, зокрема: статистичний модуль для адміністратора, що дозволяє аналізувати популярність об'єктів, середню тривалість оренди, фінансові показники; система зворотного зв'язку та оцінювання, яка надає користувачам можливість залишати відгуки про локації; підсистема повідомлень, що надсилає сповіщення про підтвердження або зміну статусу бронювання через e-mail або в особистому кабінеті.

Таким чином, описаний технологічний стек забезпечує реалізацію повноцінної, безпечної та масштабованої системи управління орендою робочих місць у коворкінгах. Обрані рішення відповідають сучасним галузевим практикам та технічним вимогам до подібних систем, дозволяючи забезпечити ефективну інтеграцію з іншими сервісами, зручний інтерфейс для користувачів та адміністрації, а також можливість подальшого розширення функціоналу відповідно до потреб бізнесу.

4.2 Реалізація основних компонентів системи

Розроблена система управління орендою робочих місць у коворкінгах реалізується як багаторівневий веб-додаток, що складається з клієнтської частини (інтерфейсу користувача), серверної логіки (бекенду) та бази даних для зберігання інформації. Такий підхід дозволяє розділити відповідальність між компонентами, забезпечити гнучкість та масштабованість системи. Основними функціональними модулями є: модуль автентифікації, модуль керування бронюванням, модуль адміністрування об'єктів, модуль обробки аналітики та інтерфейсні компоненти користувача.

Ключова частина серверної логіки реалізована мовою Python із застосуванням фреймворку Django. Django забезпечує логічне структурування проєкту у вигляді окремих додатків (apps), які об'єднують моделі, контролери та шаблони, пов'язані з окремими функціональними блоками. Комунікація між користувачем і сервером відбувається за допомогою HTTP-запитів, а дані передаються у форматі JSON у разі роботи з API або HTML – у разі рендерингу сторінок [53].

Модуль автентифікації користувачів забезпечує створення облікового запису, вхід до системи, зміну паролю та скидання паролю через email. Django надає вбудовану модель користувача User, яка розширюється при потребі, наприклад, додаванням полів organization, phone_number, role.

```
# models.py
from django.contrib.auth.models import AbstractUser

class CustomUser(AbstractUser):
    phone_number = models.CharField(max_length=15)
    role = models.CharField(max_length=20, choices=[('admin', 'Admin'), ('user', 'User')])
```

Реєстрація нового користувача здійснюється через POST-запит до відповідного маршруту, який обробляє форму реєстрації, зберігає нового користувача та автоматично виконує його вхід у систему.

```
# views.py
from django.contrib.auth import login
from .forms import CustomUserCreationForm

def register_view(request):
    if request.method == 'POST':
        form = CustomUserCreationForm(request.POST)
        if form.is_valid():
            user = form.save()
            login(request, user)
            return redirect('dashboard')
```

Модуль управління бронюванням відповідає за відображення доступності робочих місць, створення та зберігання резервацій, а також їх скасування або зміну. Модель Booking включає інформацію про дату, час початку і завершення, об'єкт бронювання та користувача, який створив запис.

```
# models.py
class Booking(models.Model):
    user = models.ForeignKey(CustomUser, on_delete=models.CASCADE)
    resource = models.ForeignKey('Resource', on_delete=models.CASCADE)
    start_time = models.DateTimeField()
    end_time = models.DateTimeField()
    status = models.CharField(max_length=20, default='active')
```

Після отримання запиту на бронювання сервер перевіряє, чи не зайнятий об'єкт у вибраний період, і, якщо він доступний, створює відповідний запис у базі даних.

```
# views.py
from django.http import JsonResponse

def create_booking(request):
    if request.method == 'POST':
        data = json.loads(request.body)
```

```

overlaps = Booking.objects.filter(
    resource_id=data['resource_id'],
    start_time__lt=data['end_time'],
    end_time__gt=data['start_time']
)
if overlaps.exists():
    return JsonResponse({'error': 'Time slot is already booked'}, status=400)

booking = Booking.objects.create(
    user=request.user,
    resource_id=data['resource_id'],
    start_time=data['start_time'],
    end_time=data['end_time']
)
return JsonResponse({'message': 'Booking created successfully'})

```

Модуль управління об'єктами (ресурсами) реалізовано за допомогою моделі Resource, яка включає опис кожного робочого місця або кімнати, ціну, тип, місткість та додаткові параметри. Адміністратор має можливість додавати, редагувати або видаляти об'єкти через окремий інтерфейс.

```

class Resource(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    capacity = models.IntegerField()
    price_per_hour = models.DecimalField(max_digits=6, decimal_places=2)
    available = models.BooleanField(default=True)

```

Інтерфейс адміністратора побудований на Django Admin або за допомогою кастомних шаблонів, які дозволяють візуально керувати ресурсами. Також реалізовано фільтрацію та сортування за параметрами [54, 55].

Модуль аналітики відповідає за обробку статистичних даних щодо активності користувачів, кількості бронювань, середньої тривалості, найбільш популярних об'єктів тощо. Цей модуль базується на вибірках з таблиць Booking і Resource, а

результати можуть бути виведені у вигляді графіків за допомогою бібліотеки Chart.js на стороні клієнта.

На клієнтському рівні реалізовано адаптивний інтерфейс користувача, створений із використанням HTML5, CSS3, JavaScript і Bootstrap. Наприклад, форма бронювання включає елементи вибору дати і часу, список ресурсів з описом та кнопкою підтвердження.

```
<form id="bookingForm">
  <label for="resource">Оберіть ресурс:</label>
  <select id="resource" name="resource_id" class="form-select"></select>

  <label for="start">Дата та час початку:</label>
  <input type="datetime-local" id="start" name="start_time" class="form-control">

  <label for="end">Дата та час завершення:</label>
  <input type="datetime-local" id="end" name="end_time" class="form-control">

  <button type="submit" class="btn btn-primary mt-2">Забронювати</button>
</form>
```

Дані передаються на сервер асинхронно за допомогою AJAX-запитів. Це забезпечує зручність користування без перезавантаження сторінки.

```
$('#bookingForm').submit(function(e) {
  e.preventDefault();
  const formData = {
    resource_id: $('#resource').val(),
    start_time: $('#start').val(),
    end_time: $('#end').val()
  };
  $.ajax({
    url: '/api/create-booking/',
    method: 'POST',
    data: JSON.stringify(formData),
    contentType: 'application/json',
```

```
success: function(response) {  
    alert(response.message);  
},  
error: function(response) {  
    alert('Бронювання не вдалося: ' + response.responseJSON.error);  
}  
});  
});
```

База даних PostgreSQL виступає у ролі центрального сховища, де міститься інформація про користувачів, ресурси, бронювання та транзакції. Завдяки використанню Django ORM, SQL-запити генеруються автоматично, однак за потреби можуть бути застосовані «сирі» запити для оптимізації важких аналітичних операцій.

Таким чином, структура системи орієнтована на максимальну модульність, гнучкість розширення, підтримку багатьох користувачів та ефективну обробку даних у реальному часі. Всі компоненти взаємодіють через загальну базу даних, забезпечуючи цілісність даних та узгодженість дій між користувачами й адміністраторами.

4.3 Тестування роботи автоматизованої системи управління

В даному підрозділі описується функціональне тестування головних функцій системи, що доступні в модулі Користувача, а також в модулі Адміністратора.

4.3.1 Модуль користувача

Після переходу на головну сторінку автоматизованої системи управління орендою робочих місць у коворкінгах, користувач має можливість залогуватися за допомогою електронної пошти. На головному екрані, що представлено на рисунку 4.1, розміщено інтерактивне меню, що дозволяє фільтрувати ресурси за категоріями

(наприклад, «переговорні кімнати», «робочі місця», «події»), а також перемикатися між варіантами відображення – у вигляді списку, сітки або календаря.

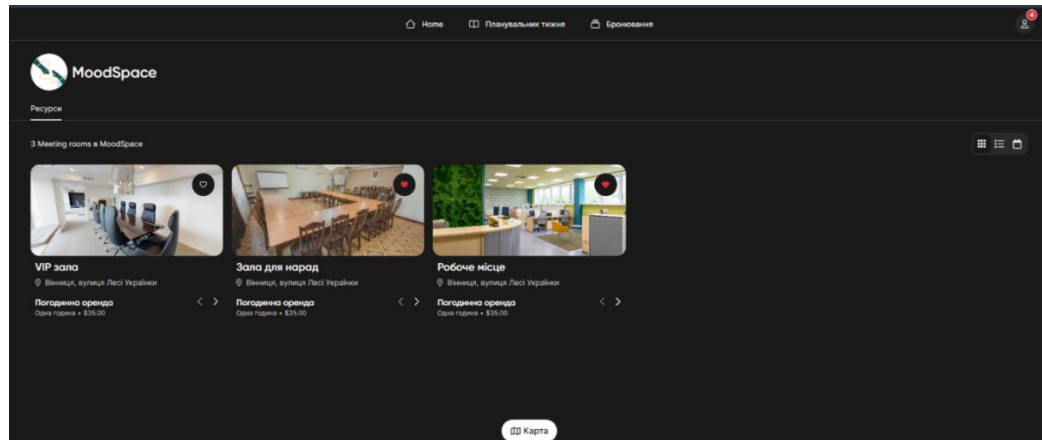


Рисунок 4.1 – Відображення головного меню для користувача

У правій частині інтерфейсу інтегровано мапу Google Maps, яка демонструє геолокацію кожного доступного простору. При наведенні курсора на об'єкт або натисканні на мітку з'являється спливаюча картка з назвою локації, коротким описом та кнопкою переходу до деталей. Відображення місця розташування коворкінгу на мапі представлено на рисунку 4.2.

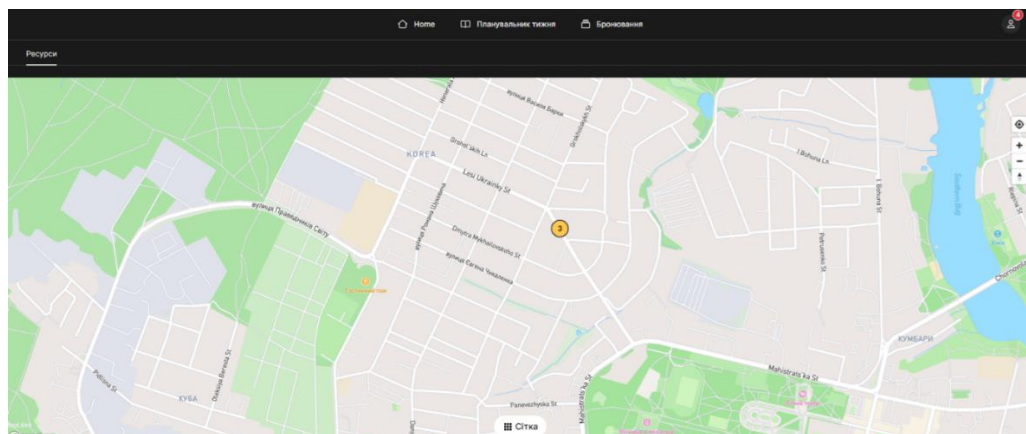


Рисунок 4.2 – Відображення місця розташування коворкінгу на мапі

Після вибору простору відкривається детальна сторінка з інформацією про об'єкт, зокрема: повна назва, адреса, тип ресурсу, кількість осіб, що можуть одночасно перебувати у приміщенні, доступні зручності (Wi-Fi, монітор, кавашина), а також – умови бронювання (повний день, півдня, погодинно). На рисунку 4.3 представлено типовий вигляд картки з детальною інформацією.

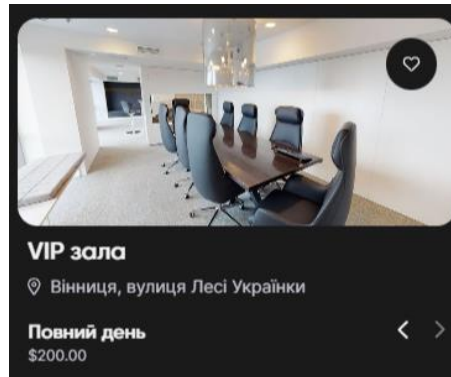


Рисунок 4.3 – Відображення умов оренди конференц зали на повний робочий день

Автоматизована система також відображає візуалізовану сітку доступності, де кожна клітинка відповідає певному часовому слоту. Користувач може вибрати потрібний інтервал одним натисканням. Відображення доступності для оренди конференс зали представлено на рисунку 4.4.

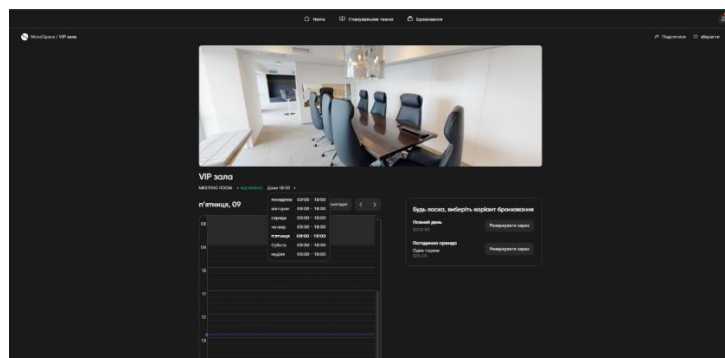


Рисунок 4.4 – Відображення доступності для оренди конференс зали

На наступному кроці система пропонує кілька варіантів бронювання – наприклад, «1 година», «2 години», «цілий день» – з відповідною вартістю.

Додатково можна вказати кількість учасників, за потреби – додати послуги або коментар. Відображення варіантів бронювання робочого місця представлено на рисунку 4.5.

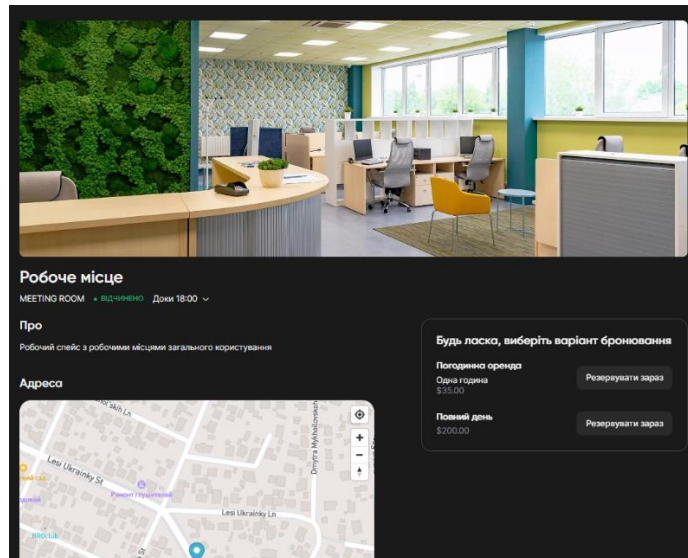


Рисунок 4.5 – Відображення варіантів бронювання робочого місця

Форма бронювання завершується відображенням підсумкової інформації, включаючи назву ресурсу, дату, час, вартість та загальну тривалість оренди. Після підтвердження система створює резервацію та надсилає повідомлення на електронну пошту користувача. Створення резервації робочого місця представлена на рисунку 4.6.

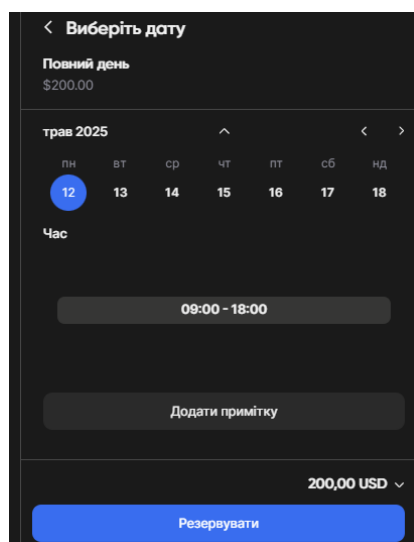


Рисунок 4.6 – Створення резервації робочого місця

Додатково користувачу доступний планувальник тижня, представлений на рисунку 4.7, який дає змогу візуально структурувати всі бронювання у вигляді графіка. Такий підхід особливо зручний для фрілансерів або команд, які планують відвідування заздалегідь.

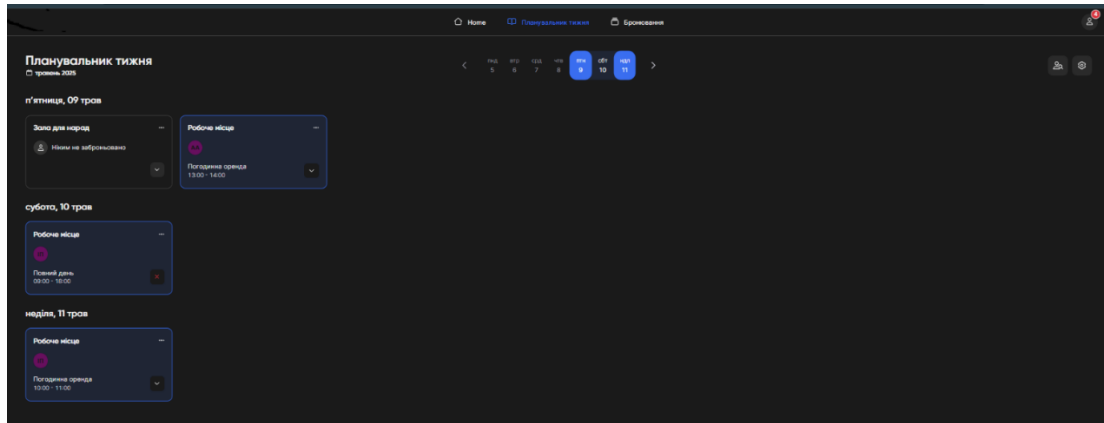


Рисунок 4.7 – Відображення планувальника тижня для користувача

Особистий кабінет надає повний список майбутніх бронювань з можливістю скасування або зміни дати, а також переглядом статусу кожного запису. Відображення запланованих бронювань користувача представлено на рисунку 4.8.

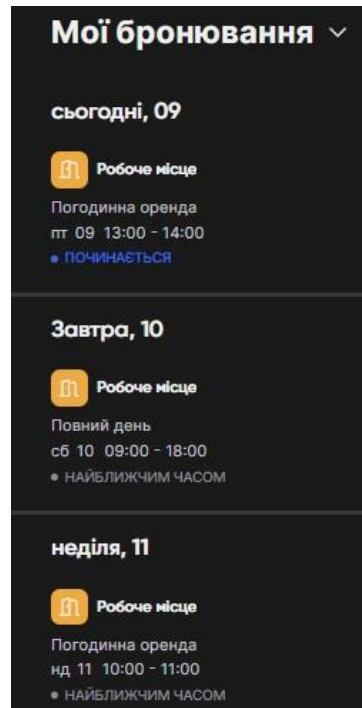


Рисунок 4.8 – Відображення запланованих бронювань користувача

Натиснувши на будь-яке бронювання, користувач може переглянути деталі бронювання, включаючи адресну інформацію, умови скасування, організатора та інше. Відповідний інтерфейс зображено на рисунку 4.9.

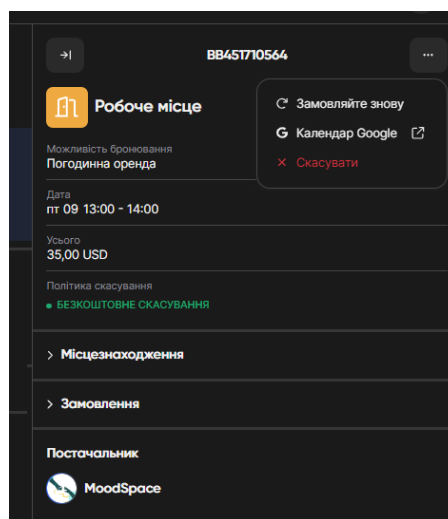


Рисунок 4.9 – Відображення деталей оренди

4.3.2 Модуль адміністратора

Модуль адміністратора автоматизованої системи містить функціональність для повного управління всіма об'єктами, що доступні для оренди. У межах цього модуля представлено список активних ресурсів із короткими описами, статусом (доступний / недоступний), а також можливістю редагування. На рисунку 4.10 продемонстровано типовий вигляд панелі адміністрування об'єктів.

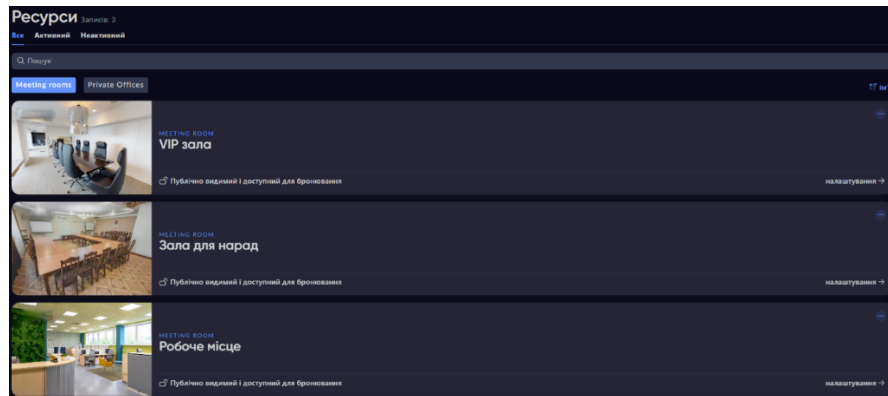


Рисунок 4.10 – Відображення активних об'єктів для оренди в модулі адміністратора

Сторінка налаштування конкретного ресурсу дає змогу змінювати назву, опис, тип ресурсу, фотографії, місткість, умови оренди, а також графік доступності. Доступні додаткові параметри – наприклад, час підготовки до зустрічі або очищення між бронюваннями. Відображення послуг та варіантів бронювання представлено на рисунку 4.11.

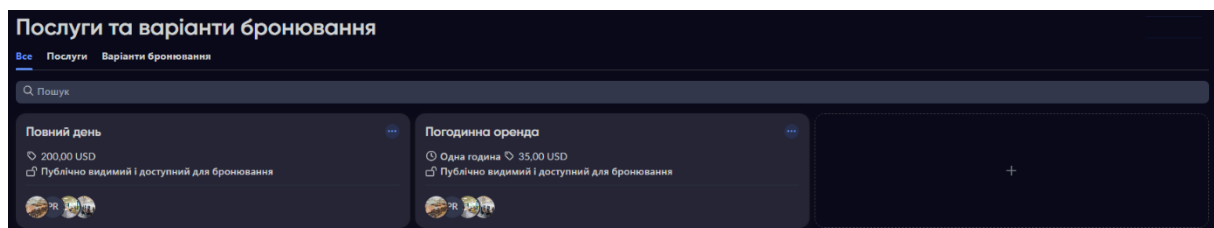
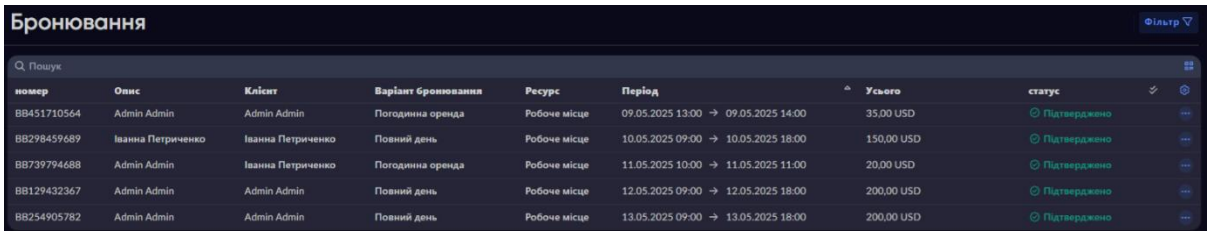


Рисунок 4.11 – Відображення послуг та варіантів бронювання

Окремий розділ модулю відповідає за перегляд історії бронювань, яка може бути відфільтрована за датою, іменем користувача або типом простору. Тут також

відображається статус оплати, підтвердження та тривалість кожної резервації. Відображення історії бронювань представлено на рисунку 4.12.



номер	Опис	Клієнт	Варіант бронювання	Ресурс	Період	Усього	статус
BB451710564	Admin Admin	Admin Admin	Погодина оренда	Робоче місце	09.05.2025 13:00 → 09.05.2025 14:00	35,00 USD	Підтверджено
BB298459689	Іванна Петриченко	Іванна Петриченко	Повний день	Робоче місце	10.05.2025 09:00 → 10.05.2025 18:00	150,00 USD	Підтверджено
BB739794688	Admin Admin	Іванна Петриченко	Погодина оренда	Робоче місце	11.05.2025 10:00 → 11.05.2025 11:00	20,00 USD	Підтверджено
BB129432367	Admin Admin	Admin Admin	Повний день	Робоче місце	12.05.2025 09:00 → 12.05.2025 18:00	200,00 USD	Підтверджено
BB254905782	Admin Admin	Admin Admin	Повний день	Робоче місце	13.05.2025 09:00 → 13.05.2025 18:00	200,00 USD	Підтверджено

Рисунок 4.12 – Відображення історії бронювань

Для кращого контролю над розкладом адміністратор може переглядати всі бронювання у форматі календаря, що хображено на рисунку 4.13, де кожна подія представлена як блок, що відповідає певному часу та ресурсу. Це дозволяє зручно слідкувати за завантаженістю об’єктів.

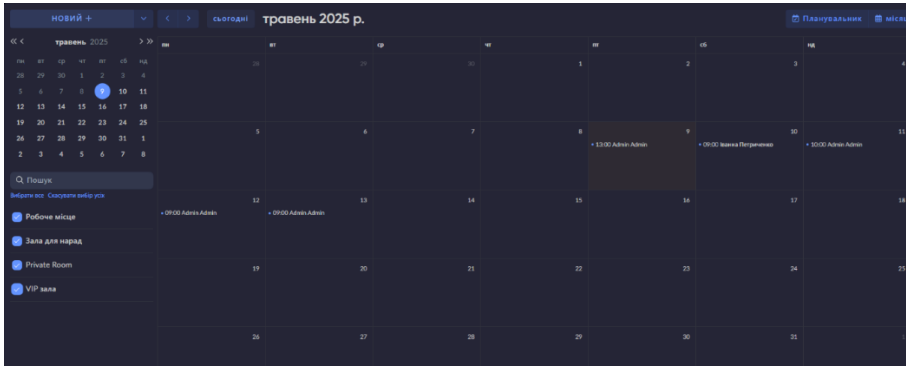


Рисунок 4.13 – Відображення бронювань на календарі

Крім того, в адміністративному інтерфейсі доступна панель аналітики, яка зображена на рисунку 4.14, що надає зведену інформацію про кількість бронювань, середню тривалість, дохід, популярні ресурси та активність користувачів у різні періоди часу. Ці дані можуть використовуватись для оптимізації тарифів і завантаженості.

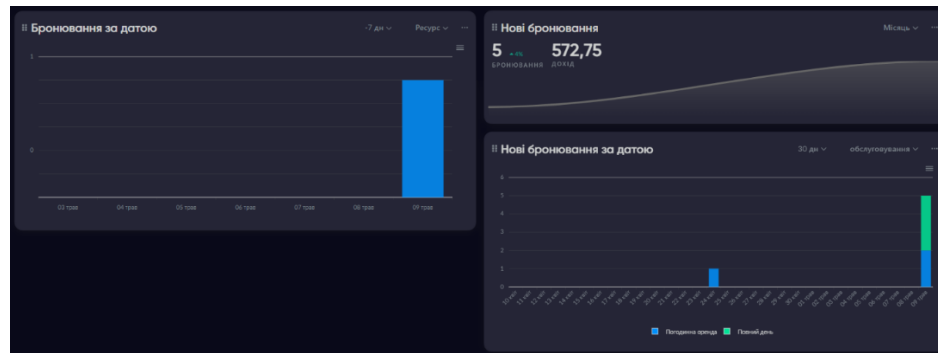


Рисунок 4.14 – Відображення аналітики

Отже, в результаті функціонального тестування підтверджено працездатність усіх основних компонентів системи. Перевірено логіку авторизації, реєстрації та управління користувачами, а також повний цикл створення та обробки бронювання, включаючи перевірку доступності ресурсів, збереження даних у базі та візуалізацію результатів. Усі виявлені під час тестування помилки були усунені, а система продемонструвала стабільність роботи та відповідність заявленим функціональним вимогам.

Таким чином, реалізована система може бути використана як готове рішення для цифровізації процесів управління коворкінг-простором, з можливістю подальшої інтеграції із зовнішніми сервісами, розширення інтерфейсів і адаптації до нових бізнес-потреб.

ВИСНОВКИ

В процесі виконання бакалаврської кваліфікаційної роботи було здійснено комплексне дослідження актуальності та об'єкта автоматизації управління орендою робочих місць у сфері коворкінгів

Проведений аналіз показав, що сучасні системи управління у сфері оренди коворкінгів мають комплексну структуру та високий рівень функціональності. Було визначено основні компоненти таких систем, їхні функціональні можливості, архітектурні рішення та інтеграційні особливості. Також виявлено ключові проблеми існуючих рішень, окреслено напрями їх удосконалення та виокремлено актуальні інноваційні тенденції, що формують розвиток цієї галузі.

Існуючі програмні рішення для управління орендою робочих місць — зокрема Nexodus, OfficeRnD, Coworkify, Cobot і Satellite Deskworks — відрізняються як за функціональним наповненням, так і за технічною реалізацією. Систематизація їхніх можливостей дозволила сформулювати перелік вимог до нової системи, враховуючи рівень автоматизації, інтеграцію з іншими сервісами та зручність для користувачів.

Було запропоноване проектне рішення системи управління орендою робочих місць у коворкінгах відповідає визначеним вимогам та враховує необхідні аспекти масштабованості, гнучкості та функціональності. Розроблені ER-діаграми, UML-діаграми класів, сценаріїв використання і потоків діяльності забезпечили повне уявлення про логіку та структуру майбутньої системи.

Реалізована система, створена з використанням сучасного технологічного стеку, повністю відповідає проектному задуму. Усі ключові компоненти — інтерфейс, серверна частина та база даних — були успішно впроваджені та протестовані.

Результати тестування підтвердили стабільну роботу системи, її функціональність та готовність до практичного застосування.

Таким чином, виконане дослідження і розробка створили підґрунтя для впровадження ефективної автоматизованої системи управління орендою робочих місць у коворкінгах, здатної задовольнити сучасні вимоги галузі та забезпечити її подальший розвиток.

Список використаних джерел

1. Дяконов А.Г. Інформаційні технології управління. – М.: Фінанси та статистика, 2010. – 288 с.
2. Костін В.В. Автоматизовані системи управління підприємствами: навч. посіб. – К.: НАУ, 2014. – 278 с.
3. Олійник О.І. Інформаційні системи і технології в економіці: навч. посіб. – К.: Центр учбової літератури, 2016. – 392 с.
4. Turban E., Volonino L. Information Technology for Management: Driving Digital Transformation to Increase Local and Global Performance, Growth and Sustainability. 12th ed. Hoboken: Wiley, 2021. 512 p. [Електронний ресурс]. Режим доступу: <https://www.wiley.com/en-us/Information+Technology+for+Management%3A+Driving+Digital+Transformation+to+Increase+Local+and+Global+Performance%2C+Growth+and+Sustainability%2C+12th+Edition-p-9781119702841>
5. Ерліх Ю. Системи управління базами даних. – К.: Либідь, 2012. – 303 с.
6. Stair R., Reynolds G. *Principles of Information Systems* / R. Stair, G. Reynolds. – 14-е вид. – Boston: Cengage Learning, 2021. – 758 с. [Електронний ресурс]. Режим доступу: <https://www.cengage.com/c/etextbook-principles-of-information-systems-14e-reynolds-stair/9780357710784>
7. Котляренко В.В. Технології розподілених обчислень. – Х.: ХНУРЕ, 2015. – 320 с.
8. Russell S., Norvig P. *Artificial Intelligence: A Modern Approach*. 4-те вид. — Harlow: Pearson, 2020. — 1152 с. [Електронний ресурс]. Режим доступу: <https://www.pearson.com/en-us/subject-catalog/p/artificial-intelligence-a-modern-approach/P200000003500/9780137505135>
9. Слепов В.В. Архітектура корпоративних інформаційних систем. – К.: Ліра-К, 2013. – 240 с.

10. Davenport T.H., Harris J.G. *Competing on Analytics: The New Science of Winning*. — Boston: Harvard Business School Press, 2007. — 218 с. [Електронний ресурс]. Режим доступу: <https://www.perlego.com/book/837308/competing-on-analytics-updated-with-a-new-introduction-the-new-science-of-winning-pdf>
11. Барановський В. А. Конкурентоспроможність підприємств в умовах цифрової економіки. — К.: КНЕУ, 2018. — 212 с.
12. Schwab K. *The Fourth Industrial Revolution*. — World Economic Forum, 2016. — 192 р.
13. Гнатюк С.Н., Калюжний А.В. Інтернет речей: основи, архітектура, протоколи. — К.: НТУУ «КПІ», 2020. — 288 с.
14. Ray P.P. A survey on Internet of Things architectures // *Journal of King Saud University – Computer and Information Sciences* [Електронний ресурс]. — 2016. — Vol. 30(3). — Р. 291–319. — Режим доступу: <https://doi.org/10.1016/j.jksuci.2016.10.003> — Дата звернення: 16.06.2025.
15. Dhanjani N. *Abusing the Internet of Things: Blackouts, Freakouts, and Stakeouts* [Електронний ресурс]. — O'Reilly Media, 2015. — 296 р. — Режим доступу: <https://www.oreilly.com/library/view/abusing-the-internet/9781491922393/> — Дата звернення: 16.06.2025.
16. Witten I.H., Frank E., Hall M.A., Pal C.J. *Data Mining: Practical Machine Learning Tools and Techniques*. — 4th ed. [Електронний ресурс]. — Morgan Kaufmann, 2016. — 654 р. — Режим доступу: <https://www.sciencedirect.com/book/9780128042915/data-mining> — Дата звернення: 16.06.2025.
17. Фролов О.Ю. Корпоративні інформаційні системи: інтеграція, взаємодія, безпека. — К.: КНЕУ, 2017. — 200 с.
18. Sommerville I. *Software Engineering*. — 10th ed. [Електронний ресурс]. — Pearson Education, 2016. — 816 р. — Режим доступу:

<https://www.pearson.com/en-us/subject-catalog/p/software-engineering/P200000006047> – Дата звернення: 16.06.2025.

19. Provost F., Fawcett T. Data Science for Business [Електронний ресурс]. – O'Reilly Media, 2013. – 414 p. – Режим доступу: <https://www.oreilly.com/library/view/data-science-for/9781449374273/> – Дата звернення: 16.06.2025.
20. Stallings W. Cryptography and Network Security: Principles and Practice. – 8th ed. [Електронний ресурс]. – Pearson, 2023. – 784 p. – Режим доступу: <https://www.pearson.com/en-us/subject-catalog/p/cryptography-and-network-security-principles-and-practice/P200000006037> – Дата звернення: 16.06.2025.
21. Рябченко О.В. Безпека інформаційних технологій. – К.: ДУІКТ, 2015. – 272 с.
22. ISO/IEC 27001:2013. Information security management systems – Requirements [Електронний ресурс]. – Режим доступу: <https://www.iso.org/standard/54534.html> – Дата звернення: 16.06.2025.
23. Chen H., Chiang R.H.L., Storey V.C. Business Intelligence and Analytics: From Big Data to Big Impact // *MIS Quarterly* [Електронний ресурс]. – 2012. – Vol. 36(4). – P. 1165–1188. – Режим доступу: <https://www.jstor.org/stable/41703503> – Дата звернення: 16.06.2025.
24. Aggarwal C.C. Outlier Analysis. – 2nd ed. [Електронний ресурс]. – Springer, 2016. – 466 p. – Режим доступу: <https://link.springer.com/book/10.1007/978-3-319-47578-3>
25. Nexodus Official Website. – [Електронний ресурс]. – Режим доступу: <https://www.nexodus.com/>
26. OfficeRnD Knowledge Base. – [Електронний ресурс]. – Режим доступу: <https://help.officernd.com/>
27. Coworkify Documentation. – [Електронний ресурс]. – Режим доступу: <https://www.coworkify.com/>

- 28.Cobot User Guide. – [Электронный ресурс]. – Режим доступа:
<https://www.cobot.me/>
- 29.Shmueli G., Bruce P.C., Patel N.R. Data Mining for Business Analytics. – Wiley, 2017. – 560 p. [Электронный ресурс]. – Режим доступа:
<https://www.wiley.com/en-us/Data+Mining+for+Business+Analytics%3A+Concepts%2C+Techniques%2C+and+Applications+in+Python-p-9781119540691> – Дата звернення:
16.06.2025.
- 30.Brynjolfsson E., McAfee A. The Second Machine Age. – W.W. Norton & Company, 2014. – 320 p. [Электронный ресурс]. – Режим доступа:
<https://wnorton.com/books/the-second-machine-age/> – Дата звернення:
16.06.2025.
- 31.Jordan M.I., Mitchell T.M. Machine learning: Trends, perspectives, and prospects // *Science* [Электронный ресурс]. – 2015. – Vol. 349(6245). – P. 255–260. – Режим доступа: <https://www.science.org/doi/10.1126/science.aaa8415> – Дата звернення: 16.06.2025.
32. Talluri K., van Ryzin G. The Theory and Practice of Revenue Management. – Springer, 2004. – 712 p. [Электронный ресурс]. – Режим доступа:
<https://link.springer.com/book/10.1007/978-0-387-21824-5> – Дата звернення:
16.06.2025.
- 33.Nielsen J. Mobile Usability. – New Riders, 2012. – 240 p. [Электронный ресурс]. – Режим доступа: <https://www.nngroup.com/books/mobile-usability/> – Дата звернення: 16.06.2025.
- 34.Zhang Y., Deng R.H. Security and Privacy in Smart Office Environments // *IEEE Internet of Things Journal* [Электронный ресурс]. – 2020. – Vol. 7(5). – P. 4202–4212. – Режим доступа: <https://ieeexplore.ieee.org/document/9032347> – Дата звернення: 16.06.2025.

35. Tapscott D., Tapscott A. Blockchain Revolution. – Penguin, 2016. – 384 p.
[Электронный ресурс]. – Режим доступа: <https://blockchain-revolution.com/> –
Дата звернения: 16.06.2025.
36. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System [Электронный
ресурс]. – 2008. – Режим доступа: <https://bitcoin.org/bitcoin.pdf> – Дата
звернения: 16.06.2025.
37. Buterin V. A Next-Generation Smart Contract and Decentralized Application
Platform [Электронный ресурс]. – Ethereum White Paper, 2014. – Режим
доступу: <https://ethereum.org/en/whitepaper/> – Дата звернения: 16.06.2025.
38. Marr B. Big Data in Practice. – Wiley, 2016. – 320 p. [Электронный ресурс]. –
Режим доступа: [https://www.wiley.com/en-
us/Big+Data+in+Practice%3A+How+17+Successful+Companies+Used+Big+Da
ta+Analytics+to+Deliver+Extraordinary+Results-p-9781119231387](https://www.wiley.com/en-us/Big+Data+in+Practice%3A+How+17+Successful+Companies+Used+Big+Data+Analytics+to+Deliver+Extraordinary+Results-p-9781119231387) – Дата
звернения: 16.06.2025.
39. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and
TensorFlow. – 2nd ed. [Электронный ресурс]. – O'Reilly Media, 2019. – 856 p.
– Режим доступа: [https://www.oreilly.com/library/view/hands-on-machine-
learning/9781492032632/](https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/)
40. Milgram P., Kishino F. A taxonomy of mixed reality visual displays. – IEICE
Transactions on Information and Systems, 1994. – Vol. E77-D(12). – P. 1321–
1329. – [Электронный ресурс]. – Режим доступа:
[\[https://ieeexplore.ieee.org/document/386262\]](https://ieeexplore.ieee.org/document/386262)([https://ieeexplore.ieee.org/docume
nt/386262](https://ieeexplore.ieee.org/document/386262))
41. Craig A.B. Understanding Augmented Reality: Concepts and Applications. –
Morgan Kaufmann, 2013. – 296 p. – [Электронный ресурс]. – Режим доступа:
[\[https://www.elsevier.com/books/understanding-augmented-reality/craig/978-0-
12-394424-5\]](https://www.elsevier.com/books/understanding-augmented-reality/craig/978-0-12-394424-5)([https://www.elsevier.com/books/understanding-augmented-
reality/craig/978-0-12-394424-5](https://www.elsevier.com/books/understanding-augmented-reality/craig/978-0-12-394424-5))

42. Siciliano B., Khatib O. Springer Handbook of Robotics. – 2nd ed. – Springer, 2016. – 2226 p. – \[Електронний ресурс]. – Режим доступу: https://link.springer.com/book/10.1007/978-3-319-32552-1](<https://link.springer.com/book/10.1007/978-3-319-32552-1>)
43. Козлов В.Ю. Хмарні технології в управлінні підприємством. – К.: Ліра-К, 2019. – 312 с.
44. Richardson L., Ruby S. RESTful Web Services. – O'Reilly Media, 2007. – 454 p. – \[Електронний ресурс]. – Режим доступу: https://learning.oreilly.com/library/view/restful-web-services/9780596529260/](<https://learning.oreilly.com/library/view/restful-web-services/9780596529260/>)
45. Hightower K., Burns B., Beda J. Kubernetes: Up and Running. – 3rd ed. – O'Reilly Media, 2022. – 350 p. – \[Електронний ресурс]. – Режим доступу: https://learning.oreilly.com/library/view/kubernetes-up-and-running/9781492056458/](<https://learning.oreilly.com/library/view/kubernetes-up-and-running/9781492056458/>)
46. Turnbull J. The Docker Book: Containerization is the new virtualization. – James Turnbull, 2014. – 348 p. – \[Електронний ресурс]. – Режим доступу: https://www.dockerbook.com/](<https://www.dockerbook.com/>)
47. Luhn H.P. A Business Intelligence System. – IBM Journal, 2018. – Vol. 2(4). – P. 314–319. – \[Електронний ресурс]. – Режим доступу: https://ieeexplore.ieee.org/document/4563028](<https://ieeexplore.ieee.org/document/4563028>)
48. Krug S. Don't Make Me Think: A Common Sense Approach to Web Usability. – 3rd ed. – New Riders, 2014. – 216 p. – \[Електронний ресурс]. – Режим доступу: https://sensible.com/dont-make-me-think/](<https://sensible.com/dont-make-me-think/>)

49. Voigt P., von dem Bussche A. The EU General Data Protection Regulation (GDPR): A Practical Guide. – Springer, 2017. – 384 p. – \[Электронный ресурс]. – Режим доступа: [\[https://www.springer.com/gp/book/9783319579597\]](https://www.springer.com/gp/book/9783319579597)(<https://www.springer.com/gp/book/9783319579597>)
50. Calder A. IT Governance: An International Guide to Data Security and ISO27001/ISO27002. – Kogan Page, 2015. – 384 p. – \[Электронный ресурс]. – Режим доступа: [\[https://www.koganpage.com/product/it-governance-9780749479858\]](https://www.koganpage.com/product/it-governance-9780749479858)(<https://www.koganpage.com/product/it-governance-9780749479858>)
51. Schwaber K., Sutherland J. The Scrum Guide. – Scrum.org, 2020. – \[Электронный ресурс]. – Режим доступа: [\[https://scrumguides.org/\]](https://scrumguides.org/)(<https://scrumguides.org/>)
52. Holovaty A., Kaplan-Moss J. The Definitive Guide to Django. – Apress, 2009. – 528 p. – \[Электронный ресурс]. – Режим доступа: [\[https://link.springer.com/book/10.1007/978-1-4302-1993-8\]](https://link.springer.com/book/10.1007/978-1-4302-1993-8)(<https://link.springer.com/book/10.1007/978-1-4302-1993-8>)
53. Spurlock J. Bootstrap: Responsive Web Development. – O'Reilly Media, 2013. – 140 p. – \[Электронный ресурс]. – Режим доступа: [\[https://learning.oreilly.com/library/view/bootstrap-responsive-web/9781449303031/\]](https://learning.oreilly.com/library/view/bootstrap-responsive-web/9781449303031/)(<https://learning.oreilly.com/library/view/bootstrap-responsive-web/9781449303031/>)

Додатки

(обов'язковий)

Додаток А
(обов'язковий)

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Автоматизована система управління орендою робочих місць у коворкінгах

Тип роботи: бакалаврська кваліфікаційна робота
(бакалаврська кваліфікаційна робота / магістерська кваліфікаційна робота)

Підрозділ кафедра КСУ
(кафедра, факультет, навчальна група)

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism 1,2 %

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

☒ Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту

У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.

У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

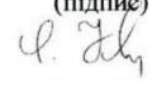
Експертна комісія:

Завідувач кафедри КСУ
(прізвище, ініціали, посада)


(підпис)

В'ячеслав КОВТУН

Гарант ОПІ
(прізвище, ініціали, посада)


(підпис)

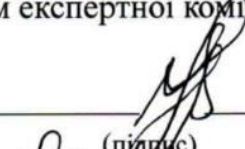
Олег КОВАЛЮК

Особа, відповідальна за перевірку


(підпис)

Володимир ДУБОВОЙ_
(прізвище, ініціали)

З висновком експертної комісії ознайомлений(-на)

Керівник 
(підпис)

Марія ЮХИМЧУК
(прізвище, ініціали, посада)

Здобувач 
(підпис)

Вікторія БРАТКОВА
(прізвище, ініціали)

Додаток Б

(обов'язковий)
ВНТУ

ЗАТВЕРДЖЕНО

Зав. кафедри КСУ ВНТУ,

д.т.н., проф. Вячеслав КОВТУН

“ 24 ” травня 2025 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання бакалаврської кваліфікаційної роботи
АВТОМАТИЗОВАНА СИСТЕМА УПРАВЛІННЯ ОРЕНДОЮ РОБОЧИХ МІСЦЬ
У КОВОРКІНГАХ

Студент групи



2АКІТ-216 Вікторія БРАТКОВА

Керівник



д. т. н. проф. Марія ЮХИМЧУК

Вінниця 2025

1. Назва та галузь застосування

1.1. Назва – Автоматизована система управління орендою робочих місць у коворкінгах.

1.2. Галузь застосування – Комп'ютеризовані системи управління ресурсами та послугами у сфері гнучких робочих просторів та коворкінг-індустрії.

2. Підстава для проведення розробки.

Тема бакалаврської кваліфікаційної роботи затверджена наказом по ВНТУ № 97 від 20. 03. 2025 р.

3. Мета та призначення розробки.

Метою дипломної роботи є розробка автоматизованої системи управління орендою робочих місць у коворкінгах, яка б забезпечувала ефективне та зручне керування процесами бронювання, оплати та контролю доступу, а також сприяла підвищенню якості обслуговування клієнтів і оптимізації внутрішніх операцій.

;

4. Джерела розробки.

Бакалаврська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Костін В.В. Автоматизовані системи управління підприємствами: навч. посіб. – К.: НАУ, 2014. – 278 с.
2. Козлов В.Ю. Хмарні технології в управлінні підприємством. – К.: Ліра-К, 2019. – 312 с.
3. Spurlock J. Bootstrap: Responsive Web Development. – O'Reilly Media, 2013. – 140 р.
4. Zhang Y., Deng R.H. Security and Privacy in Smart Office Environments. – IEEE Internet of Things Journal, 2020. – Vol. 7(5). – P. 4202–4212.

5. Вимоги до розробки.

5.1. Перелік головних функцій системи:

- реєстрація та авторизація користувачів із розподілом прав доступу (звичайний користувач / адміністратор);
- перегляд доступних ресурсів із фільтрацією за типом, кількістю місць і ціною;
- створення, редагування та скасування бронювання об'єктів оренди;
- надсилання сповіщень користувачам про статус бронювання;
- додавання та редагування ресурсів у адміністративній панелі;
- завантаження зображень до об'єктів оренди;
- перегляд персональної історії бронювань користувача.

5.2.1. Вимоги до програмної платформи:

- WINDOWS 7\8\10;

- веб-браузери: останні версії Chrome, Firefox;
- серверна частина: Linux-сервер з підтримкою Python, Django або Node.js;
- база даних: PostgreSQL;
- інтерфейс користувача: сучасний, адаптивний.

5.2.2. Умови експлуатації системи:

- робота системи на стандартних ПК або серверах у приміщеннях з нормальними умовами навколишнього середовища;
- можливість цілодобового функціонування без перерв;
- технічна підтримка та оновлення системи – за графіком, з урахуванням потреб користувачів;
- текст програмного забезпечення системи – закритий, ліцензійний.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

- Дослідження актуальності поставленої задачі «10» травня 2025 р.
- Дослідження та аналіз роботи систем управління послугами в сфері коворкінгів «14» травня 2025 р.
- Дослідження функціональних можливостей сучасних програмних рішень «19» травня 2025 р.
- Проектування автоматизованої системи управління орендою робочих місць у коворкінгах «20» травня 2025 р.
- Розробка автоматизованої системи управління орендою робочих місць у коворкінгах «21» травня 2025 р.

6.2 Графічні матеріали:

- Реляційна схема бази даних «09» червня 2025 р.
- UML-діаграми «09» червня 2025 р.
- Графічне відображення автоматизованої системи «09» червня 2025 р.

7. Порядок контролю і приймання.

- 7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «04» червня 2025 р.
- 7.2. Атестація роботи здійснюється на попередньому захисті. Попередній захист бакалаврської кваліфікаційної роботи провести до «10» червня 2025 р.
- 7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК.
- 7.4. Захист бакалаврської кваліфікаційної роботи провести «17» червня 2025 р.

Додаток В

(довідковий)

Лістинг програмного коду

Модуль користувача (реєстрація та вхід)

models.py:

```
from django.contrib.auth.models import AbstractUser
from django.db import models
```

```
class CustomUser(AbstractUser):
    phone_number = models.CharField(max_length=15)
    role = models.CharField(max_length=20,
choices=[('user', 'User'), ('admin', 'Admin')])
```

forms.py:

```
from django import forms
from django.contrib.auth.forms import UserCreationForm
from .models import CustomUser
```

```
class CustomUserCreationForm(UserCreationForm):
    class Meta:
        model = CustomUser
        fields = ('username', 'email', 'phone_number', 'role')
```

views.py:

```
from django.contrib.auth import login, authenticate
from django.shortcuts import render, redirect
from .forms import CustomUserCreationForm
```

```
def register_view(request):
    if request.method == 'POST':
        form = CustomUserCreationForm(request.POST)
        if form.is_valid():
            user = form.save()
            login(request, user)
            return redirect('dashboard')
    else:
```

```
        form = CustomUserCreationForm()
        return render(request, 'register.html', {'form': form})

def login_view(request):
    if request.method == 'POST':
        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username,
password=password)
        if user is not None:
            login(request, user)
            return redirect('dashboard')
        return render(request, 'login.html')
```

Модуль управління ресурсами (робочі місця, переговорні кімнати)

models.py:

```
class Resource(models.Model):
    name = models.CharField(max_length=100)
    description = models.TextField()
    capacity = models.PositiveIntegerField()
    price_per_hour = models.DecimalField(max_digits=8,
decimal_places=2)
    is_active = models.BooleanField(default=True)
```

views.py:

```
from django.shortcuts import render, get_object_or_404
from .models import Resource

def resource_list(request):
    resources = Resource.objects.filter(is_active=True)
    return render(request, 'resources/list.html', {'resources':
resources})
```

```
def resource_detail(request, pk):
    resource = get_object_or_404(Resource, pk=pk)
    return render(request, 'resources/detail.html',
                  {'resource': resource})
```

Модуль бронювання

models.py:

```
class Booking(models.Model):
    user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
    resource = models.ForeignKey(Resource,
on_delete=models.CASCADE)
    start_time = models.DateTimeField()
    end_time = models.DateTimeField()
    status = models.CharField(max_length=20,
default='active')
```

views.py:

```
from django.http import JsonResponse
from django.views.decorators.csrf import csrf_exempt
import json
from .models import Booking, Resource
from django.utils.dateparse import parse_datetime
```

@csrf_exempt

```
def create_booking(request):
    if request.method == 'POST':
        data = json.loads(request.body)
        resource_id = data['resource_id']
        start = parse_datetime(data['start_time'])
        end = parse_datetime(data['end_time'])

        overlaps = Booking.objects.filter(
            resource_id=resource_id,
            start_time__lt=end,
            end_time__gt=start
        )
        if overlaps.exists():
            return JsonResponse({'error': 'Slot is already
```

```
booked'}, status=400)
```

```
        booking = Booking.objects.create(
            user=request.user,
            resource_id=resource_id,
            start_time=start,
            end_time=end
        )
        return JsonResponse({'message': 'Booking created'})
```

Модуль перегляду бронювань користувачем

views.py:

```
from django.contrib.auth.decorators import login_required

@login_required
def my_bookings(request):
    bookings = Booking.objects.filter(user=request.user).order_by('-
start_time')
    return render(request, 'bookings/my_bookings.html',
                  {'bookings': bookings})
```

Модуль адміністратора (керування ресурсами)

views.py:

```
from django.contrib.admin.views.decorators import
staff_member_required

@staff_member_required
def admin_resource_list(request):
    resources = Resource.objects.all()
    return render(request, 'admin/resources/list.html',
                  {'resources': resources})
```

@staff_member_required

```
def add_resource(request):
    if request.method == 'POST':
        name = request.POST['name']
        description = request.POST['description']
        capacity = int(request.POST['capacity'])
        price = float(request.POST['price_per_hour'])
```

```

Resource.objects.create(name=name,
description=description,          capacity=capacity,
price_per_hour=price)
    return redirect('admin_resource_list')
return render(request, 'admin/resources/add.html')

```

Модуль аналітики бронювань

views.py:

```
from django.db.models import Count, Avg, Sum
```

```
@staff_member_required
```

```
def booking_analytics(request):
```

```
    total_bookings = Booking.objects.count()
```

```
    total_users = CustomUser.objects.count()
```

```
    most_popular_resource = Resource.objects.annotate(
```

```
        count=Count('booking')
```

```
    ).order_by('-count').first()
```

```
    total_income = Booking.objects.aggregate(
```

```
        revenue=Sum('resource__price_per_hour')
```

```
    )['revenue']
```

```
    return render(request, 'admin/analytics.html', {
```

```
        'total_bookings': total_bookings,
```

```
        'total_users': total_users,
```

```
        'popular_resource': most_popular_resource,
```

```
        'total_income': total_income
```

```
    })
```

HTML-шаблони (фрагменти)

resources/list.html:

```
<h2>Доступні ресурси</h2>
```

```
<ul>
```

```
    {% for resource in resources %}
```

```
    <li>
```

```
        <a href="{% url 'resource_detail' resource.id %}">{{
```

```
resource.name }}</a>
```

```
    </li>
```

```
    {% endfor %}
```

```
</ul>
```

bookings/my_bookings.html:

```
<h2>Мої бронювання</h2>
```

```
<table>
```

```
<tr><th>Ресурс</th><th>Початок</th><th>Кінець</th>
</tr>
```

```
    {% for booking in bookings %}
```

```
    <tr>
```

```
        <td>{{ booking.resource.name }}</td>
```

```
        <td>{{ booking.start_time }}</td>
```

```
        <td>{{ booking.end_time }}</td>
```

```
    </tr>
```

```
    {% endfor %}
```

```
</table>
```

JavaScript – форма бронювання

booking.js:

```
$('#bookingForm').submit(function(e) {
```

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

```
    const data = {
```

```
        resource_id: $('#resource').val(),
```

```
        start_time: $('#start_time').val(),
```

```
        end_time: $('#end_time').val()
```

```
    };
```

```
    fetch('/api/create-booking/', {
```

```
        method: 'POST',
```

```
        headers: {
```

```
            'Content-Type': 'application/json',
```

```
            'X-CSRFToken': getCookie('csrftoken')
```

```
        },
```

```
        body: JSON.stringify(data)
```

```
    })
```

```
    .then(response => response.json())
```

```
    .then(result => alert(result.message))
```

```
    .catch(error => alert('Booking failed'));
```

```
});
```

URL-маршрути


```

urls.py:
from django.urls import path
from . import views

urlpatterns = [
    path('register/', views.register_view, name='register'),
    path('login/', views.login_view, name='login'),
    path('resources/', views.resource_list,
name='resource_list'),
    path('resources/<int:pk>/', views.resource_detail,
name='resource_detail'),
    path('bookings/', views.my_bookings,
name='my_bookings'),
    path('api/create-booking/', views.create_booking,
name='create_booking'),
    path('admin/resources/', views.admin_resource_list,
name='admin_resource_list'),
    path('admin/resources/add/', views.add_resource,
name='add_resource'),
    path('admin/analytics/', views.booking_analytics,
name='booking_analytics'),
]

```

Модуль календаря бронювань

```

views.py:
from django.http import JsonResponse
from django.core.serializers import serialize

def calendar_data(request):
    bookings = Booking.objects.filter(status='active')
    data = []
    for booking in bookings:
        data.append({
            'title': booking.resource.name,
            'start': booking.start_time.isoformat(),
            'end': booking.end_time.isoformat(),
        })
    return JsonResponse(data, safe=False)

```

```

urls.py:
path('api/calendar/', views.calendar_data,
name='calendar_data'),
calendar.js (використовується FullCalendar):

```

```

document.addEventListener('DOMContentLoaded',
function () {
    var calendarEl = document.getElementById('calendar');
    var calendar = new FullCalendar.Calendar(calendarEl, {
        initialView: 'timeGridWeek',
        events: '/api/calendar/'
    });
    calendar.render();
});

```

Система повідомлень користувача (вбудовані сповіщення)

```

models.py:
class Notification(models.Model):
    user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
    message = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
    read = models.BooleanField(default=False)

```

```

views.py:
@login_required
def notifications_view(request):
    notifications = Notification.objects.filter(user=request.user).order_by('-created_at')
    return render(request, 'notifications.html',
{'notifications': notifications})

urls.py:
path('notifications/', views.notifications_view,
name='notifications')

```

notifications.html:

```
<h3>Сповідання</h3>
<ul>
  {% for n in notifications %}
    <li {% if not n.read %}style="font-weight: bold" {%
endif %}>{{ n.message }}</li>
  {% endfor %}
</ul>
```

Модуль оцінювання ресурсу

models.py:

```
class Rating(models.Model):
    user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
    resource = models.ForeignKey(Resource,
on_delete=models.CASCADE)
    score = models.IntegerField()
    comment = models.TextField(null=True, blank=True)
    created_at =
models.DateTimeField(auto_now_add=True)
```

views.py:

```
@login_required
def submit_rating(request, resource_id):
    if request.method == 'POST':
        score = int(request.POST['score'])
        comment = request.POST.get('comment', '')
        Rating.objects.create(
            user=request.user,
            resource_id=resource_id,
            score=score,
            comment=comment
        )
        return redirect('resource_detail', pk=resource_id)
```

resource_detail.html (фрагмент):

```
<form method="post" action="{% url 'submit_rating'
```

```
resource.id %}">
```

```
{% csrf_token %}
```

```
<label for="score">Оцінка (1-5):</label>
```

```
<input type="number" name="score" min="1" max="5"
required>
```

```
<textarea name="comment" placeholder="Коментар
(необов'язково)"></textarea>
```

```
<button type="submit">Залишити оцінку</button>
```

```
</form>
```

Керування профілем користувача

views.py:

```
@login_required
```

```
def profile_view(request):
```

```
    if request.method == 'POST':
```

```
        user = request.user
```

```
        user.email = request.POST['email']
```

```
        user.phone_number =
```

```
request.POST['phone_number']
```

```
        user.save()
```

```
        return redirect('profile')
```

```
        return render(request, 'profile.html')
```

urls.py:

```
path('profile/', views.profile_view, name='profile'),
```

profile.html:

```
<h2>Профіль</h2>
```

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

```
{% csrf_token %}
```

```
<label>Email:</label>
```

```
<input type="email" name="email" value="{{ user.email
}}">
```

```
<label>Телефон:</label>
```

```
<input type="text" name="phone_number" value="{{
user.phone_number }}">
```

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

```
</form>
```

Модуль завантаження зображень для об'єктів

```
models.py:
class ResourceImage(models.Model):
    resource = models.ForeignKey(Resource,
on_delete=models.CASCADE, related_name='images')
    image = models.ImageField(upload_to='resource_images/')
```

```
views.py:
```

```
@staff_member_required
def upload_image(request, resource_id):
    if request.method == 'POST' and
request.FILES['image']:
        ResourceImage.objects.create(
            resource_id=resource_id,
            image=request.FILES['image']
        )
        return redirect('resource_detail', pk=resource_id)
```

```
urls.py:
```

```
path('admin/resources/<int:resource_id>/upload/',
views.upload_image, name='upload_image'),
```

```
resource_detail.html (фрагмент):
```

```
{% for img in resource.images.all %}

{% endfor %}
<form method="post" enctype="multipart/form-data"
action="{% url 'upload_image' resource.id %}">
    {% csrf_token %}
    <input type="file" name="image">
    <button type="submit">Завантажити</button>
</form>
```

Модуль відгуків

```
models.py:
```

```
class Review(models.Model):
    user = models.ForeignKey(CustomUser,
on_delete=models.CASCADE)
    resource = models.ForeignKey(Resource,
on_delete=models.CASCADE)
    text = models.TextField()
    created_at = models.DateTimeField(auto_now_add=True)
```

```
views.py:
```

```
@login_required
def submit_review(request, resource_id):
    if request.method == 'POST':
        text = request.POST['text']
        Review.objects.create(
            user=request.user,
            resource_id=resource_id,
            text=text
        )
        return redirect('resource_detail', pk=resource_id)
```

```
resource_detail.html (фрагмент):
```

```
<h4>Відгуки</h4>
<ul>
    {% for review in resource.review_set.all %}
        <li>{{ review.user.username }}: {{ review.text }}</li>
    {% endfor %}
</ul>
<form method="post" action="{% url 'submit_review'
resource.id %}">
    {% csrf_token %}
    <textarea name="text" required></textarea>
    <button type="submit">Залишити відгук</button>
</form>
```

Додаток Г

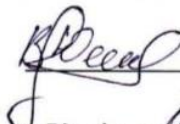
(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

«Атоматизована система управління орендою робочих місць у коворкінгах»

1. ER-діаграма автоматизованої системи управління орендою робочих місць у коворкінгах
2. UML-діаграма класів автоматизованої системи управління орендою робочих місць у коворкінгах
3. UML-діаграма варіантів використання автоматизованої системи управління орендою робочих місць у коворкінгах з акторами Користувач та Адміністратор
4. Опис функції створення бронювання у вигляді UML-діаграми активностей
5. Опис функції залишення відгуків у вигляді UML-діаграми активностей
6. UML-діаграма компонентів автоматизованої системи управління орендою робочих місць у коворкінгах
7. Відображення головного меню для користувача
8. Відображення місця розташування коворкінгу на мапі
9. Відображення умов оренди конференц зали на повний робочий день
10. Відображення доступності для оренди конференс зали
11. Відображення варіантів бронювання робочого місця
12. Створення резервації робочого місця
13. Відображення планувальника тижня для користувача
14. Відображення запланованих бронювань користувача
15. Відображення деталей оренди
16. Відображення активних об'єктів для оренди в модулі адміністратора
17. Відображення послуг та варіантів бронювання
18. Відображення історії бронювань
19. Відображення бронювань на календарі
20. Відображення аналітики

Виконав: студент 4-го курсу, групи 2АКІТ-206
спеціальності 151 – Автоматизація та
комп'ютерно-інтегровані технології
(шифр і назва спеціальності)

 Вікторія БРАТКОВА
(ім'я та прізвище)

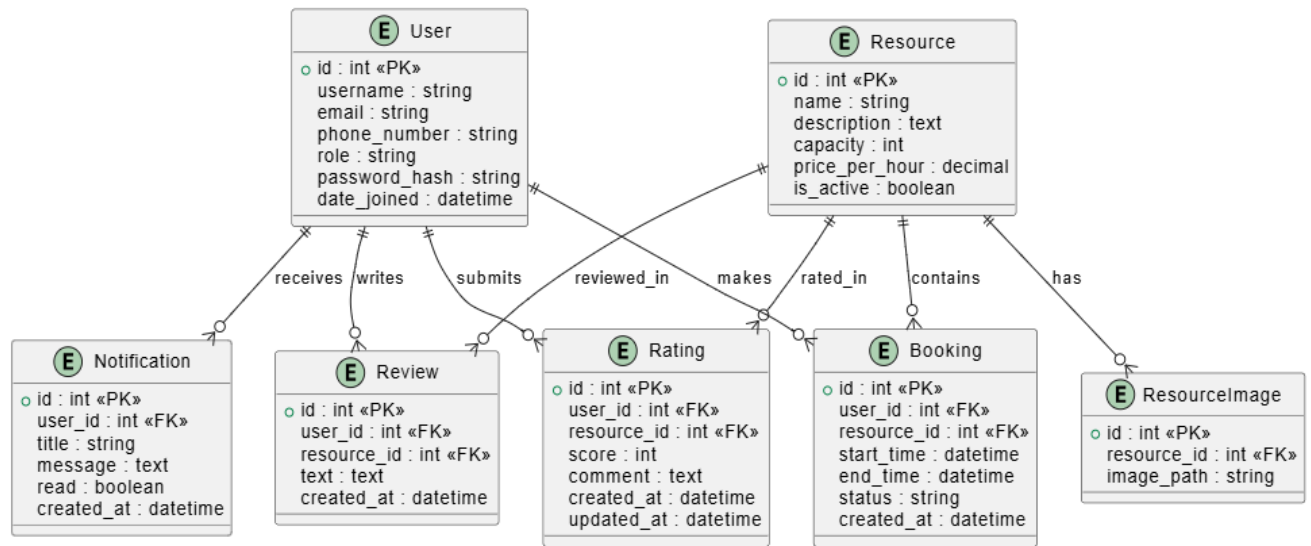
Керівник: д.т.н., проф. каф. КСУ

 Марія ЮХИМЧУК
(ім'я та прізвище)
 « 10 » 06 2025 р.

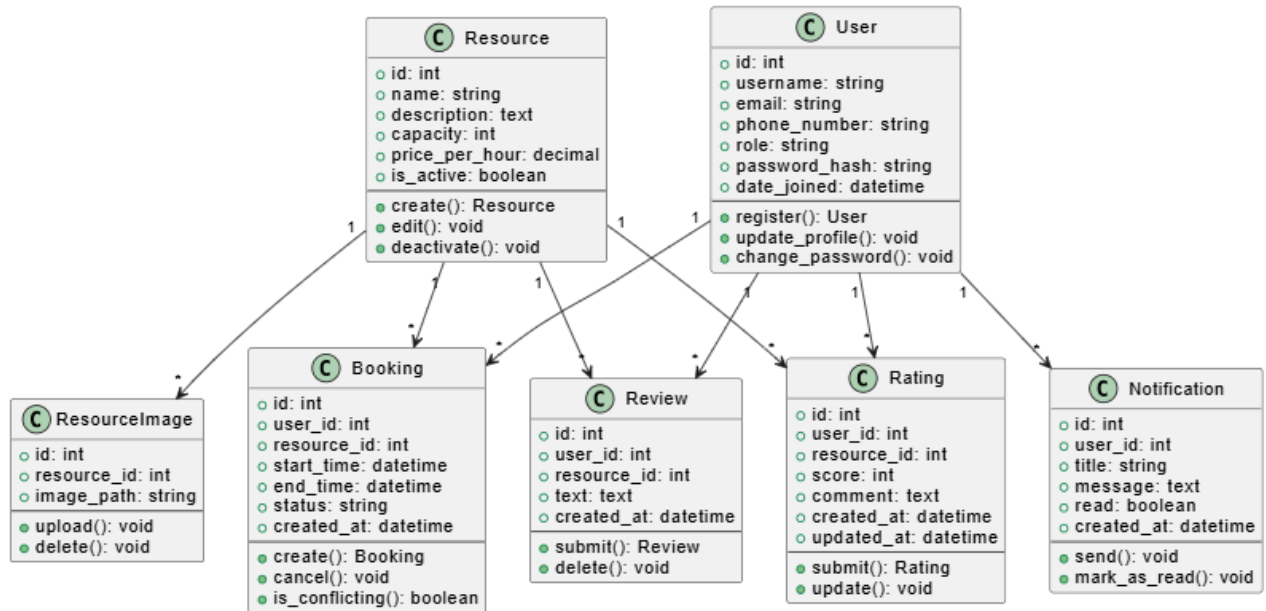
Рецензент: к.т.н., доцент каф. АІТ

 Володимир ГАРМАШ
(ім'я та прізвище)
 « 11 » 06 2025 р.

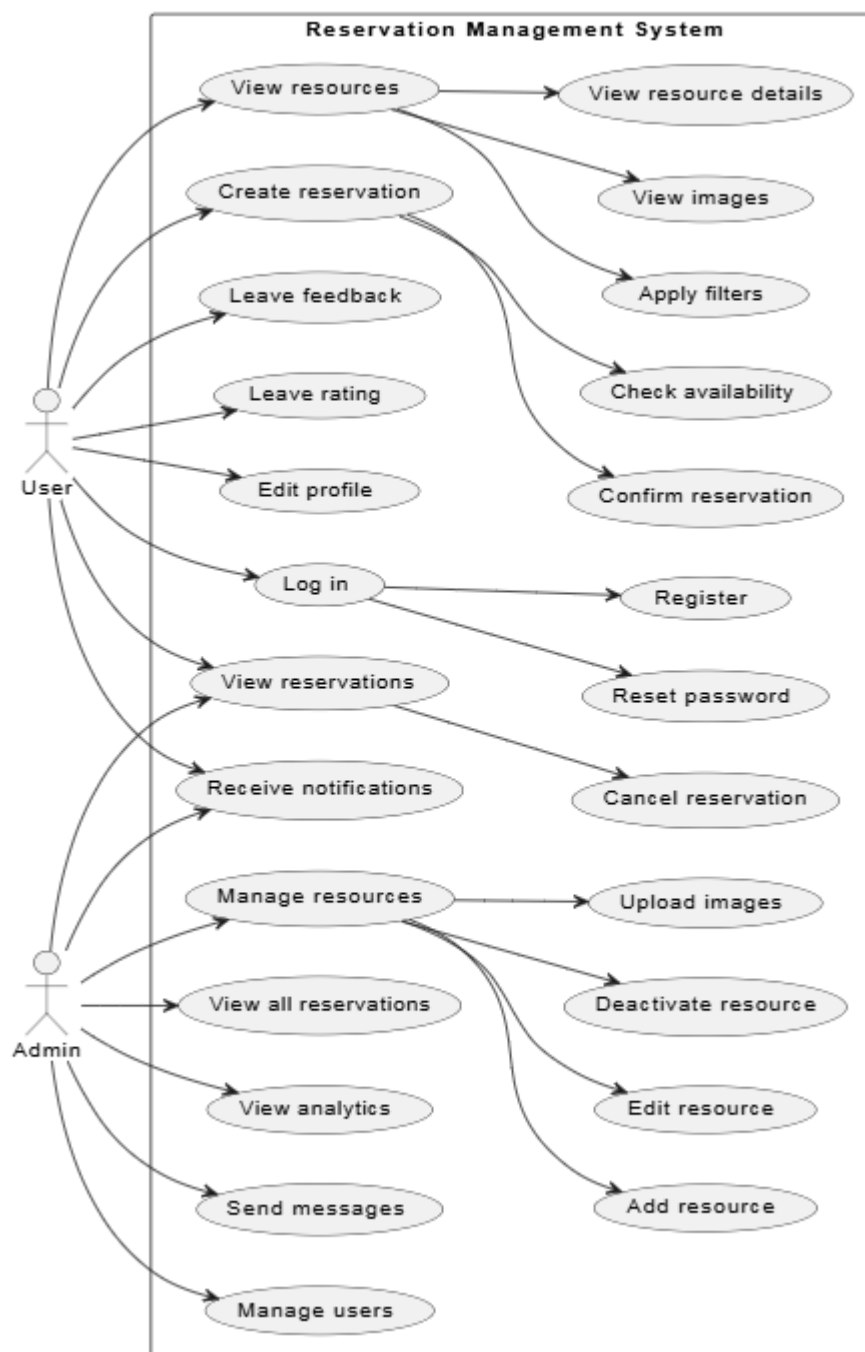
ER-діаграма автоматизованої системи управління орендою робочих місць у коворкінгах



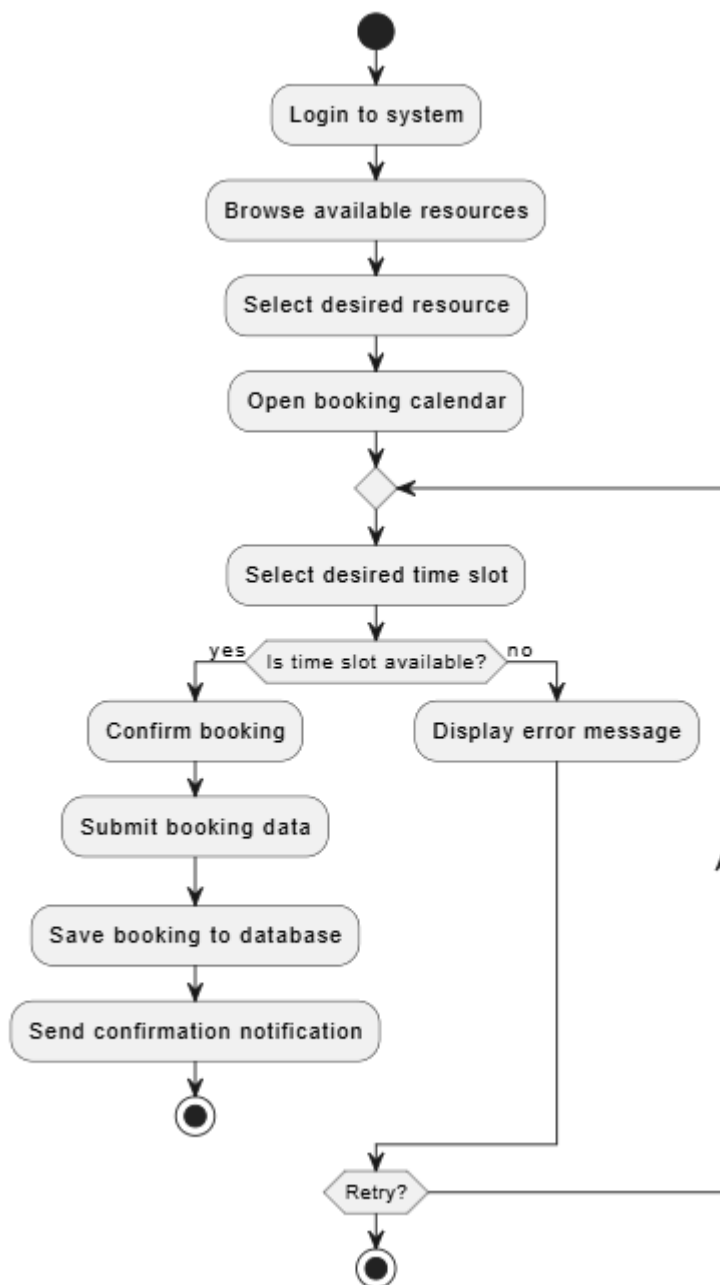
UML-діаграма класів автоматизованої системи управління орендою робочих місць у коворкінгах



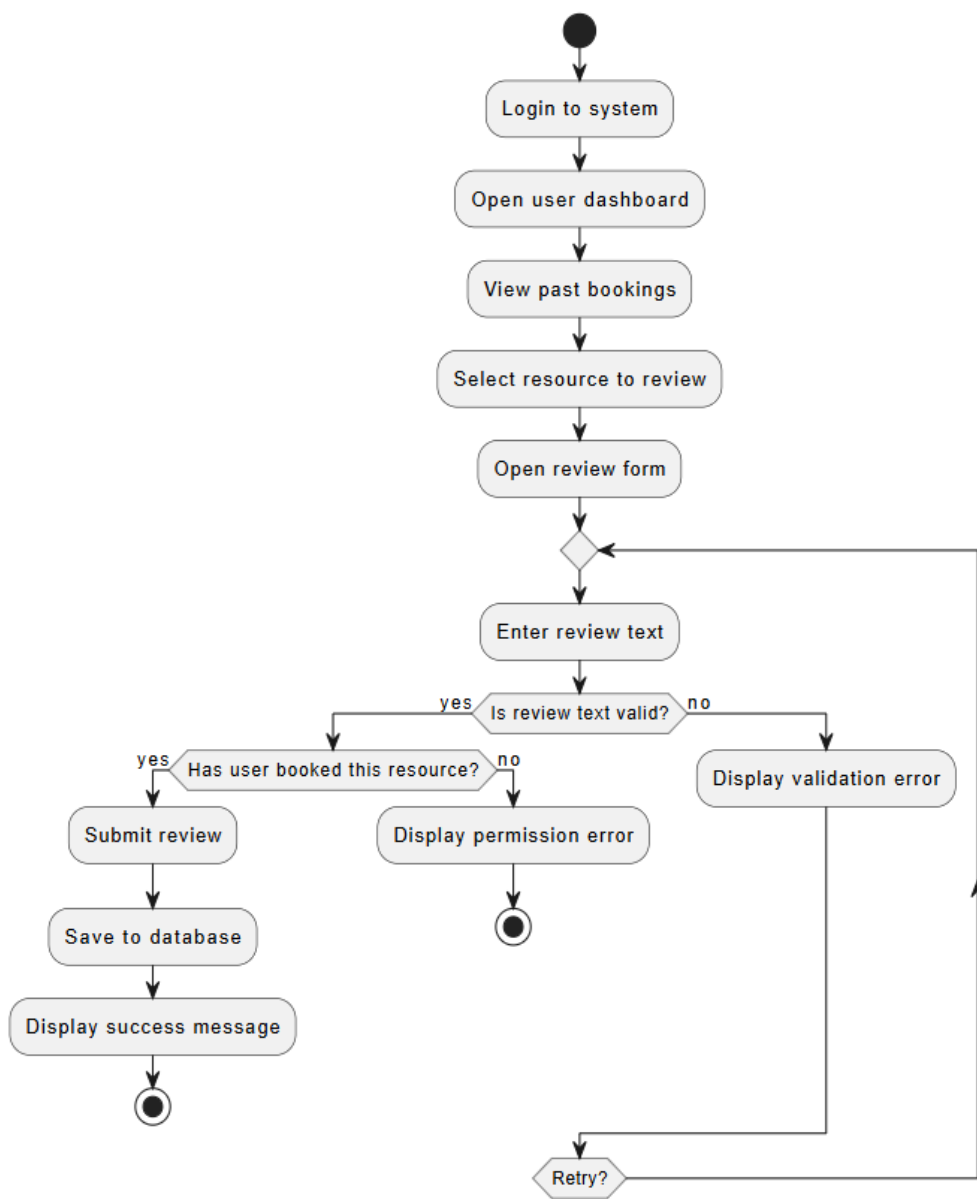
UML-діаграма варіантів використання автоматизованої системи управління орендою робочих місць у коворкінгах з акторами Користувач та Адміністратор



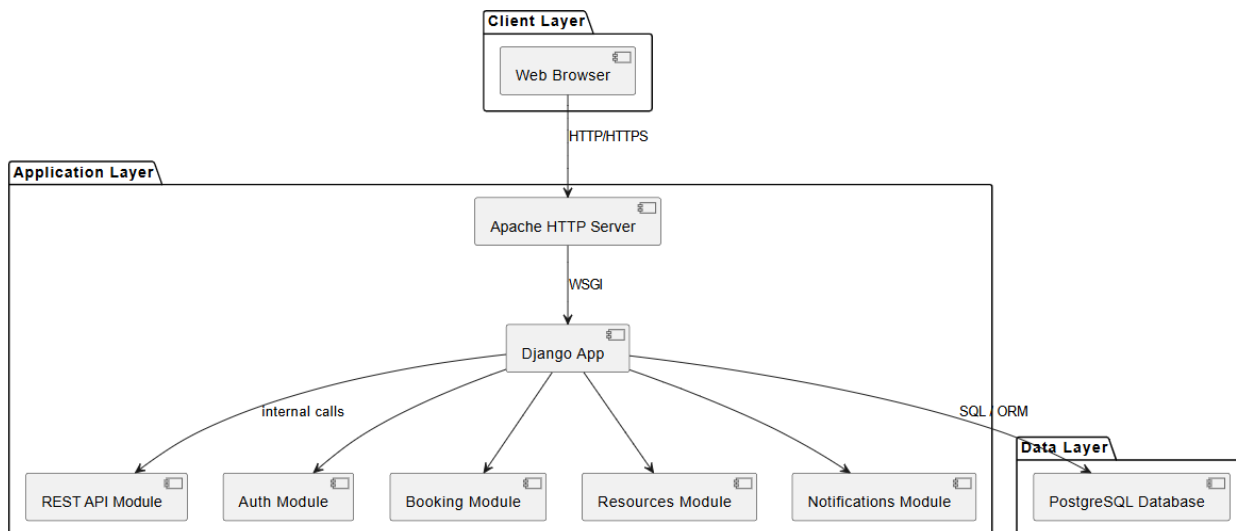
Опис функції створення бронювання у вигляді UML-діаграми активностей



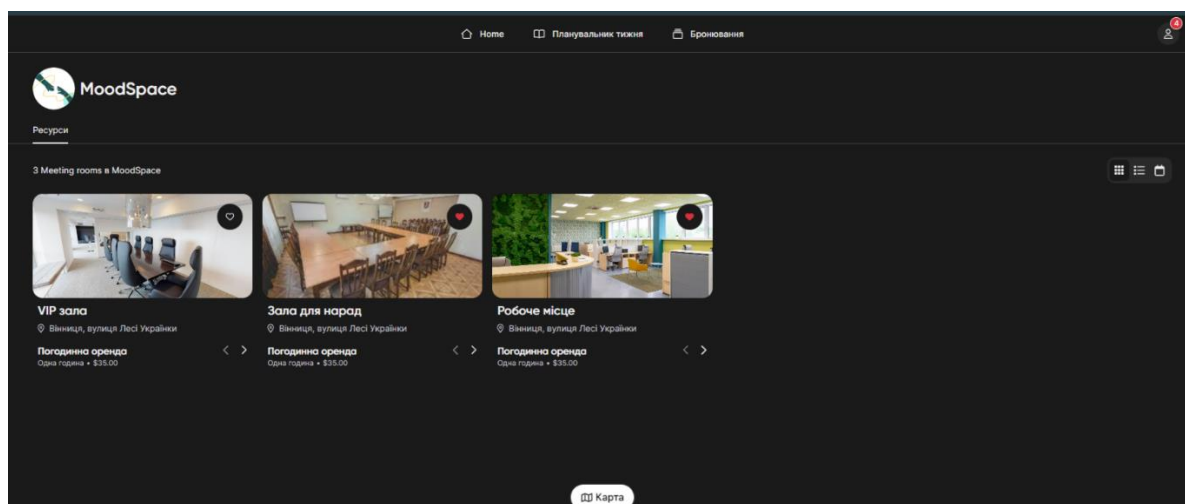
Опис функції залишення відгуків у вигляді UML-діаграми активностей



UML-діаграма компонентів автоматизованої системи управління орендою робочих місць у коворкінгах

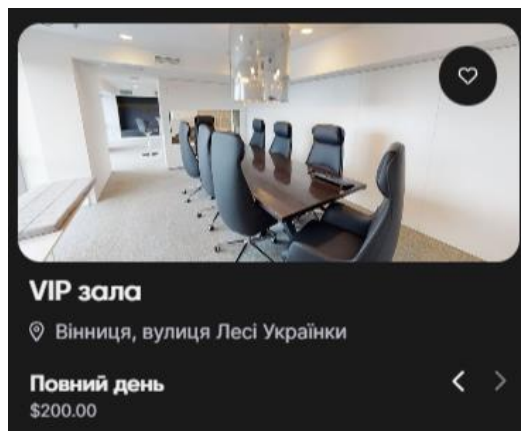


Відображення головного меню для користувача

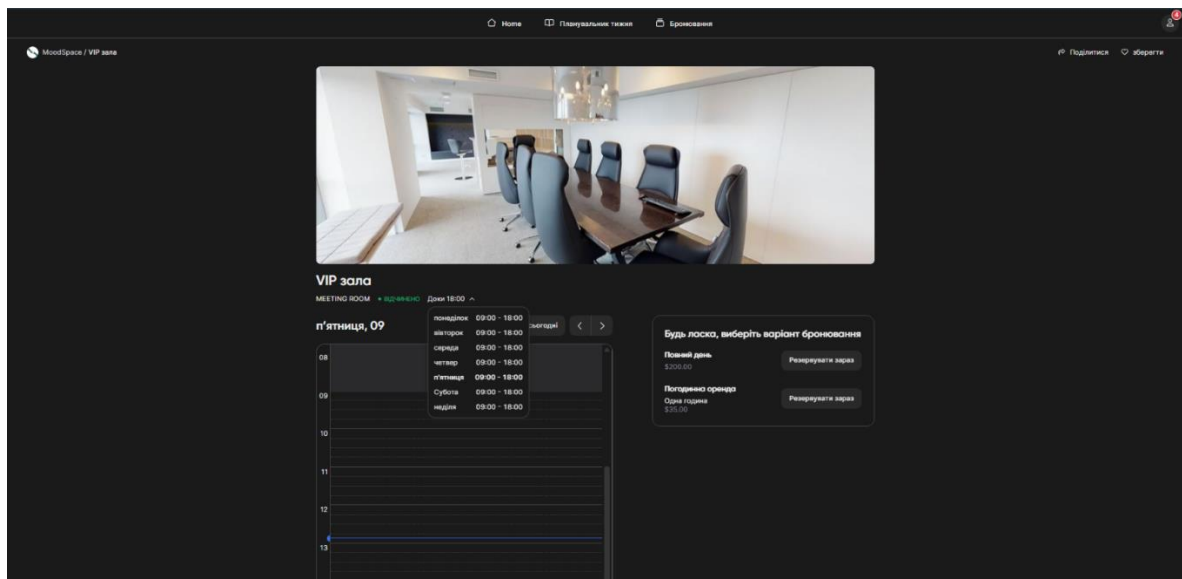




Відображення умов оренди конференц зали на повний робочий день



Відображення доступності для оренди конференс зали



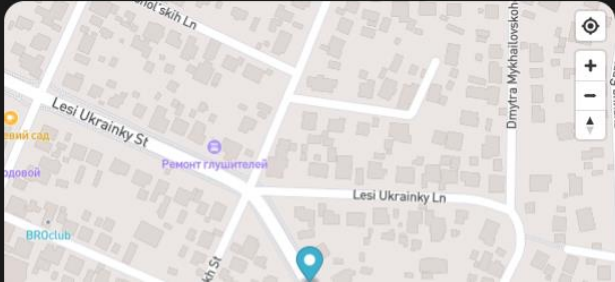
Відображення варіантів бронювання робочого місця



Робоче місце
MEETING ROOM • **ВІДЧИНЕНО** Доки 18:00 ▾

Про
Робочий спейс з робочими місцями загального користування

Адреса



Будь ласка, виберіть варіант бронювання

Погодинна оренда	
Одна година	Резервувати зараз
\$35.00	
Повний день	
\$200.00	Резервувати зараз

Створення резервації робочого місця

<

Виберіть дату

Повний день

\$200.00

трав 2025

^

<

>

пн	вт	ср	чт	пт	сб	нд
12	13	14	15	16	17	18

Час

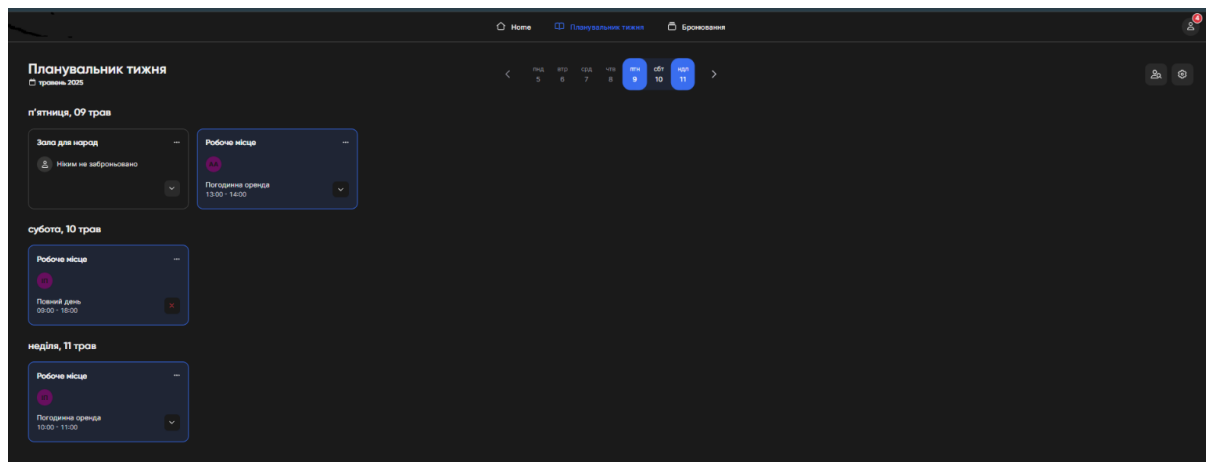
09:00 - 18:00

Додати примітку

200,00 USD ▾

Резервувати

Відображення планувальника тижня для користувача



Відображення запланованих бронювань користувача

Мої бронювання ▾

сьогодні, 09

 **Робоче місце**

Погодинна оренда
пт 09 13:00 - 14:00
● [ПОЧИНАЄТЬСЯ](#)

Завтра, 10

 **Робоче місце**

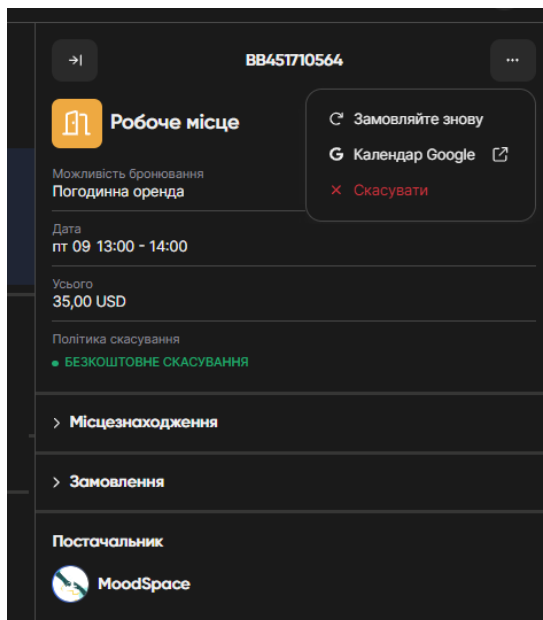
Повний день
сб 10 09:00 - 18:00
● [НАЙБЛИЖЧИМ ЧАСОМ](#)

неділя, 11

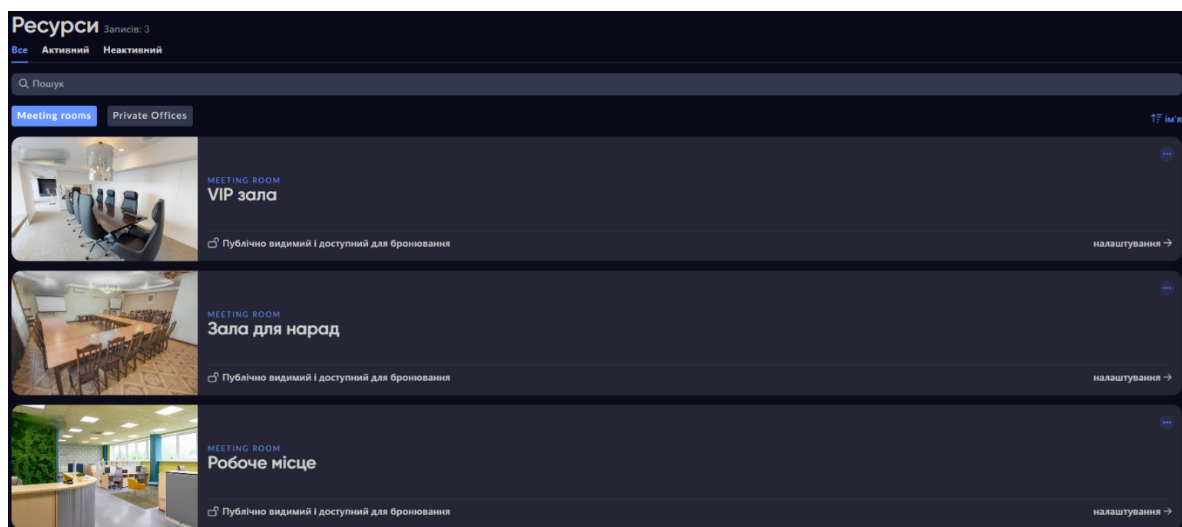
 **Робоче місце**

Погодинна оренда
нд 11 10:00 - 11:00
● [НАЙБЛИЖЧИМ ЧАСОМ](#)

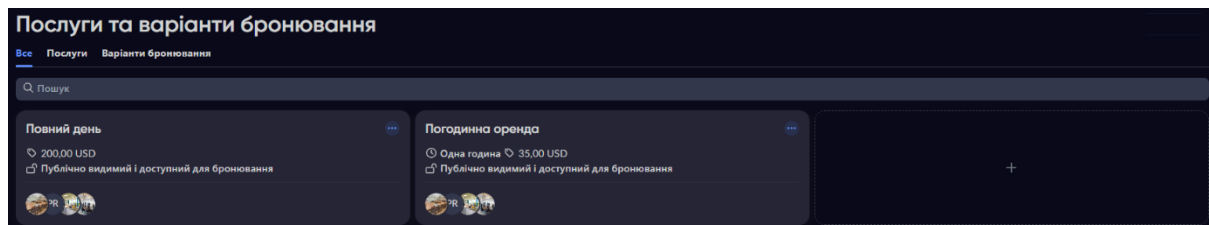
Відображення деталей оренди



Відображення активних об'єктів для оренди в модулі адміністратора



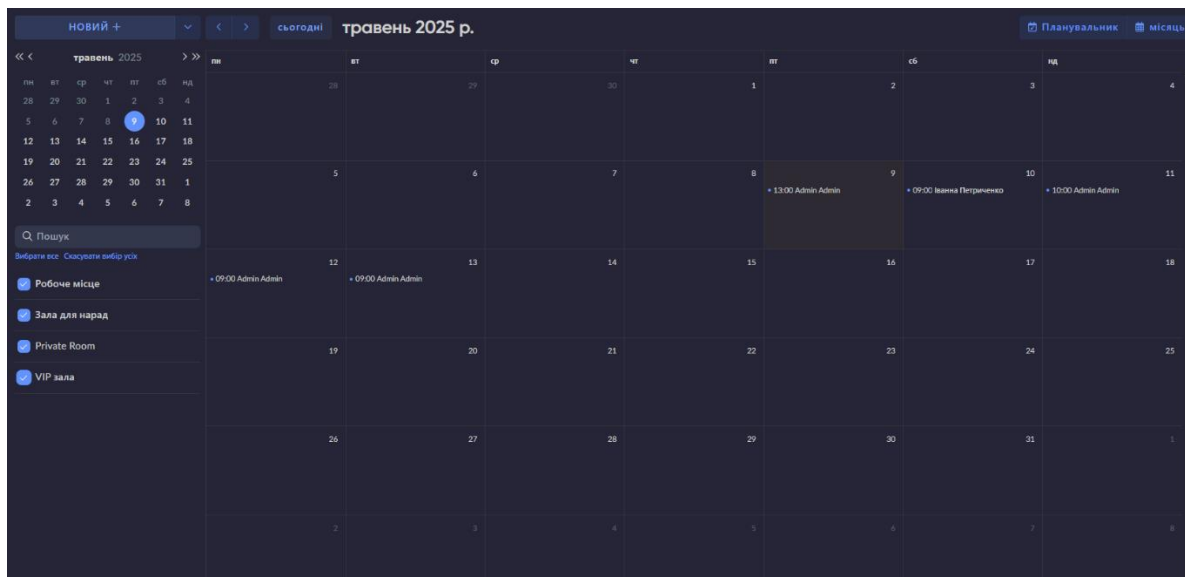
Відображення послуг та варіантів бронювання



Відображення історії бронювань

Бронювання								Фільтр
Пошук								
номер	Опис	Клієнт	Варіант бронювання	Ресурс	Період	Усього	статус	
BB451710564	Admin Admin	Admin Admin	Погодина оренда	Робоче місце	09.05.2025 13:00 → 09.05.2025 14:00	35,00 USD	Підтверджено	
BB298459689	Іванна Петриченко	Іванна Петриченко	Повний день	Робоче місце	10.05.2025 09:00 → 10.05.2025 18:00	150,00 USD	Підтверджено	
BB739794688	Admin Admin	Іванна Петриченко	Погодина оренда	Робоче місце	11.05.2025 10:00 → 11.05.2025 11:00	20,00 USD	Підтверджено	
BB129432367	Admin Admin	Admin Admin	Повний день	Робоче місце	12.05.2025 09:00 → 12.05.2025 18:00	200,00 USD	Підтверджено	
BB254905782	Admin Admin	Admin Admin	Повний день	Робоче місце	13.05.2025 09:00 → 13.05.2025 18:00	200,00 USD	Підтверджено	

Відображення бронювань на календарі



Відображення аналітики

