

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

«\_\_» \_\_\_\_\_ 2024 р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інженерія програмного  
забезпечення мультимедійних та інформаційно-пошукових систем»**

**спеціальності 121 Інженерія програмного забезпечення**

**на тему: «Вебзастосунок для обліку обладнання телеканалу для виїзних  
зйомок»**

Виконала:

студентка IV курсу, групи КП-02

Гудзіцька Дарина Сергіївна \_\_\_\_\_

Керівник:

Доцент кафедри ПЗКС, к.т.н., доцент,

Заболотня Тетяна Миколаївна \_\_\_\_\_

Консультант з нормоконтролю:

Доцент кафедри ПЗКС, к.т.н., доцент,

Онай Микола Володимирович \_\_\_\_\_

Рецензент:

Старший викладач каф. ПМА,

Мальчиков Володимир Вікторович \_\_\_\_\_

Засвідчую, що у цьому дипломному  
проєкті немає запозичень з праць інших  
авторів без відповідних посилань.

Студентка \_\_\_\_\_

Київ – 2024 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет прикладної математики**

**Кафедра програмного забезпечення комп'ютерних систем**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

«\_\_» \_\_\_\_\_ 2023 р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

Гудзіцькій Дарині Сергіївні

1. Тема проєкту «Вебзастосунок для обліку обладнання телеканалу для виїзних зйомок», керівник проєкту Заболотня Тетяна Миколаївна, к.т.н., доцент, затверджені наказом по університету від «30» травня 2024 р. № 2205-с.
2. Термін подання студентом проєкту «13» червня 2024 р.
3. Вихідні дані до проєкту: див. Технічне завдання.
4. Зміст пояснювальної записки:
  - огляд існуючих програмних рішень;
  - обґрунтування вибору засобів реалізації;
  - структурно-алгоритмічна організація системи;
  - особливості реалізації програмного забезпечення.
5. Перелік обов'язкового графічного матеріалу:
  - структура бази даних (креслення);
  - алгоритм перевірки доступності обладнання на певний період часу (креслення);
  - структура вебзастосунку (плакат);
  - діаграма прецедентів вебзастосунку (плакат).

## 6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	Онай М.В., доцент		

## 7. Дата видачі завдання «31» жовтня 2023 р.

### Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	13.11.2023	
2.	Розроблення та узгодження технічного завдання	22.11.2023	
3.	Розроблення структури вебзастосунку	15.12.2023	
4.	Підготовка матеріалів першого розділу дипломного проєкту	28.12.2023	
5.	Розроблення дизайну сторінок та графічних елементів	01.02.2024	
6.	Підготовка матеріалів другого розділу дипломного проєкту	19.02.2024	
7.	Програмна реалізація вебзастосунку	11.03.2024	
8.	Тестування вебзастосунку	18.03.2024	
9.	Підготовка матеріалів третього розділу дипломного проєкту	27.03.2024	
10.	Підготовка матеріалів четвертого розділу дипломного проєкту	15.04.2024	
11.	Підготовка графічної частини дипломного проєкту	22.04.2024	
12.	Оформлення документації дипломного проєкту	27.05.2024	

Студент

Дарина ГУДЗІЦЬКА

Керівник проєкту

Тетяна ЗАБОЛОТНЯ

## АНОТАЦІЯ

Даний дипломний проєкт присвячений створенню вебзастосунку для обліку обладнання телеканалу для виїзних зйомок.

Вебзастосунок реалізовано у вигляді вебсайту, призначеного для користування працівниками компаній, які працюють у галузі телекомунікацій та прагнуть автоматизувати процес створення та відстеження бронювань обладнання телеканалу для зйомок. Після входу в систему користувачі отримують можливість не лише створювати, оперувати власними бронюванням, але й виконувати інші функції, такі як перегляд доступності обладнання у певний час, перегляд історії бронювань та ремонтів обладнання. Крім того, користувач з правами адміністратора матиме доступ до спеціальної панелі для управління різними аспектами системи, такими як користувачі, обладнання, бронювання та зйомки.

Метою даного дипломного проєкту є забезпечення ефективного використання обладнання телеканалу, уникнення конфліктів у бронюванні обладнання та розробка відповідного програмного забезпечення.

В рамках дипломного проєкту було розроблено архітектуру серверної та клієнтської частин системи, інтерфейс користувача у формі вебзастосунку, модуль для створення бронювань, панель адміністратора, спроектовано базу даних, а також реалізовано алгоритм перевірки доступності обладнання на певний період часу.

## **ABSTRACT**

This diploma project is dedicated to the development of a web application for accounting for TV channel equipment for on-site filming.

The web application is implemented in the form of a website intended for use by employees of companies operating in the telecommunications industry and seeking to automate the process of creating and tracking reservations for equipment for filming. After logging in, users are able to not only create and manage their own bookings, but also perform other functions such as viewing the availability of equipment at a certain time, viewing the history of bookings and equipment repairs. In addition, a user with administrator rights will have access to a special panel to manage various aspects of the system, such as users, equipment, bookings, and shoots.

The aim of this diploma project is to ensure the efficient use of the filming equipment, avoid conflicts in equipment booking, and develop appropriate software.

As part of the diploma project, we developed the architecture of the server and client parts of the system, a user interface in the form of a web application, a module for creating reservations, an administrator panel, designed a database, and implemented an algorithm for checking the availability of equipment for a certain period of time.

ДП.045440-01-90 Вебзастосунок для обліку обладнання телеканалу для виїзних зйомок. Відомість проекту

Позначення	Найменування	Кіл-ть	Примітка
	Документація проекту		
ДП.045440-02-91	Вебзастосунок для обліку обладнання для виїзних зйомок. Технічне завдання	5	
ДП.045440-03-81	Вебзастосунок для обліку обладнання для виїзних зйомок. Пояснювальна записка	58	
ДП.045440-04-51	Вебзастосунок для обліку обладнання для виїзних зйомок. Програма та методика тестування	4	
ДП.045440-05-34	Вебзастосунок для обліку обладнання для виїзних зйомок. Керівництво користувача	12	
ДП.045440-06-99	Вебзастосунок для обліку обладнання для виїзних зйомок. Структура бази даних. ERD-діаграма	1	
ДП.045440-07-99	Вебзастосунок для обліку обладнання для виїзних зйомок. Алгоритм перевірки доступності обладнання на певний період часу. Блок-схема	1	
ДП.045440-08-98	Вебзастосунок для обліку обладнання для виїзних зйомок. Компакт-диск	1	

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

“ \_\_\_ ” \_\_\_\_\_ 2023 р.

**ВЕБЗАСТОСУНОК ДЛЯ ОБЛІКУ ОБЛАДНАННЯ ТЕЛЕКАНАЛУ**  
**ДЛЯ ВИЇЗНИХ ЗЙОМОК**

**Технічне завдання**

ДП.045440-02-91

“ПОГОДЖЕНО”

Керівник проекту:

\_\_\_\_\_ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Дарина ГУДЗІЦЬКА

## ЗМІСТ

1. Найменування та галузь застосування .....	3
2. Підстава для розроблення .....	3
3. Призначення розробки .....	3
4. Вимоги до програмного продукту .....	3
5. Вимоги до проєктної документації .....	4
6. Етапи проєктування.....	5
7. Порядок тестування розробки.....	5

## **1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ**

**Назва розробки:** Вебзастосунок для обліку обладнання телеканалу для виїзних зйомок.

**Галузь застосування:** інформаційні технології.

## **2. ПІДСТАВА ДЛЯ РОЗРОБЛЕННЯ**

Підставою для розроблення є завдання на дипломне проектування, затверджене кафедрою програмного забезпечення комп'ютерних систем Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» (КПІ ім. Ігоря Сікорського).

## **3. ПРИЗНАЧЕННЯ РОЗРОБКИ**

Розробка призначена для використання в якості інформаційного забезпечення роботи телеканалу з метою ефективного обліку обладнання для виїзних зйомок.

## **4. ВИМОГИ ДО ПРОГРАМНОГО ПРОДУКТУ**

Вебзастосунок повинен забезпечувати такі основні функції:

- 1) можливість перегляду та фільтрації обладнання;
- 2) можливість створення бронювання;
- 3) можливість редагування бронювання;
- 4) можливість створювати записи про ремонт обладнання;
- 5) застосування стандарту шифрування даних SHA-256;
- 6) можливість користувачем отримати пароль на пошту при його реєстрації адміністратором системи.

Додаткові вимоги:

- 1) відправка оповіщень на електронну пошту користувача;
- 2) інтуїтивно зрозумілий інтерфейс;
- 3) зберігання паролів користувачів у захешованому вигляді;
- 4) дизайн сторінок з використанням в якості базових білого та синьо-жовтих кольорів.

## **5. ВИМОГИ ДО ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ**

У процесі виконання проекту повинна бути розроблена наступна документація:

- 1) пояснювальна записка;
- 2) програма та методика тестування;
- 3) керівництво користувача;
- 4) креслення:
  - «Структура бази даних. ERD-діаграма»;
  - «Алгоритм перевірки доступності обладнання на певний період часу. Блок-схема».

## **6. ЕТАПИ ПРОЄКТУВАННЯ**

Вивчення літератури за тематикою роботи.....	13.11.2023
Розроблення та узгодження технічного завдання.....	22.11.2023
Розроблення структури вебзастсунку.....	15.12.2023
Розроблення дизайну сторінок та графічних елементів.....	01.02.2024
Програмна реалізація вебзастсунку .....	15.03.2024
Тестування вебзастсунку .....	08.04.2024
Підготовка матеріалів текстової частини проєкту.....	22.04.2024
Підготовка матеріалів графічної частини проєкту.....	15.05.2024
Оформлення технічної документації проєкту.....	27.05.2024

## **7. ПОРЯДОК ТЕСТУВАННЯ РОЗРОБКИ**

Тестування розробленого програмного продукту виконується відповідно до “Програми та методики тестування”.

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ВЕБЗАСТОСУНОК ДЛЯ ОБЛІКУ ОБЛАДНАННЯ ТЕЛЕКАНАЛУ**  
**ДЛЯ ВИЇЗНИХ ЗЙОМОК**

**Пояснювальна записка**

ДП.045440-03-81

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Дарина ГУДЗІЦЬКА

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	3
ВСТУП.....	4
1. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ.....	5
1.1. Огляд проблеми, яка вирішується ПЗ.....	5
1.2. Аналіз існуючих програмних рішень .....	6
1.3. Результати проведеного аналізу .....	13
1.4. Висновки до розділу 1 .....	14
2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ .....	15
2.1. Вибір мови програмування для реалізації серверної частини .....	15
2.2. Вибір технологій для розроблення клієнтської частини.....	19
2.3. Вибір СКБД.....	22
2.4. Висновки до розділу 2.....	25
3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ СИСТЕМИ.....	26
3.1. Опис вимог до програмного забезпечення.....	26
3.2. Опис структурної організації застосунку.....	30
3.3. Структура таблиць бази даних .....	33
3.4. Алгоритмічне забезпечення застосунку .....	36
3.5. Висновки до розділу 3.....	39
4. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ВЕБЗАСТОСУНКУ .....	41
4.1. Реалізація серверної частини.....	41
4.2. Реалізація клієнтської частини .....	42
4.3. Тестування вебзастосунку .....	48
4.4. Рекомендації щодо подальшого вдосконалення .....	50
4.5. Висновки до розділу 4.....	51
ВИСНОВКИ .....	53
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ.....	55
ДОДАТКИ.....	58

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

ПЗ – програмне забезпечення;

Букінг (*від англ. booking*) – бронювання;

Виїзна зйомка – це процес створення відеоматеріалів за межами студійного комплексу телеканалу;

СКБД – система керування базами даних;

БД (база даних) – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між елементами);

XML (eXtensible Markup Language) – розширювана мова розмітки, призначена для зберігання та передачі даних у структурованому форматі;

CSV (Comma-Separated Values) – формат для представлення табличних даних у текстовому вигляді,

JSON (JavaScript Object Notation) – легкий формат обміну даними для передачі структурованих даних між різними програмами;

ООП – об'єктно-орієнтоване програмування;

AJAX (Asynchronous JavaScript and XML) – технологія, яка дозволяє вебсторінці асинхронно обмінюватися даними з сервером без необхідності перезавантаження сторінки.

MTV (Model-Template-View) – архітектурний шаблон, який використовується під час проектування та розроблення ПЗ;

HTTP (HyperText Transfer Protocol) – протокол передачі даних, що використовується в комп'ютерних мережах;

UML-діаграма (Unified Modeling Language) – графічне зображення, яке використовується для моделювання структури та поведінки системи за допомогою мови моделювання;

ORM (Object-Relational Mapping) – технологія програмування, яка дозволяє взаємодіяти з базами даних з використанням об'єктно-орієнтованого підходу.

## ВСТУП

Сучасна галузь телекомунікацій стрімко розвивається, що вимагає впровадження новітніх технологій для підвищення ефективності бізнесу. Зокрема, одним з важливих аспектів є забезпечення ефективного обліку даних про обладнання для виїзних зйомок. На сьогоднішній день більшість телеканалів стикаються з проблемою неорганізованості та труднощів в управлінні своїм технічним оснащенням під час виїзних зйомок.

Існуючі програмні рішення дозволяють вирішувати ряд завдань, пов'язаних з обліком обладнання. Вони забезпечують базові функції інвентаризації, відстеження місцезнаходження техніки та планування використання ресурсів. Однак, більшість з них мають певні недоліки, такі як недостатня гнучкість, обмежена інтеграція з іншими системами, а також відсутність адаптації до специфічних потреб телеканалів, що робить їх неуніверсальними для використання при вирішенні ряду задач.

Виходячи з вищезазначеного, розроблення вебсервісу для обліку обладнання телеканалу для виїзних зйомок є актуальним завданням. Такий додаток повинен враховувати специфічні вимоги галузі, забезпечувати зручність користування, високу надійність та інтеграцію з іншими системами телеканалу. Це дозволить значно підвищити ефективність управління технічними ресурсами під час виїзних зйомок, знизити ризик конфліктів інтересів та оптимізувати робочі процеси.

## **1. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ**

У першому відділі пояснювальної записки представлені дані про специфіку роботи у сфері телекомунікацій, а також проведено аналіз наявних програмних рішень існуючих проблем. Нижче сформовано перелік завдань, що повинні бути виконані у рамках даного розділу:

- 1) аналіз ринку та визначення проблем, з якими стикаються в аспекті обліку технічного забезпечення для мобільних зйомок;
- 2) дослідження найпопулярніших програмних забезпечень, якими користуються фахівці у галузі телекомунікацій для бронювання обладнання, визначення переваг та недоліків кожного з них;
- 3) побудова порівняльної таблиці результатів аналізу сервісів;
- 4) формулювання переліку потрібної функціональності розроблюваного ПЗ.

### **1.1. Огляд проблеми, яка вирішується ПЗ**

У сучасних реаліях медіа-групи активно використовують виїзні зйомки для широкого охоплення подій і створення емоційно насиченого контенту. У процесі організації та управління обладнанням для зйомок виникають численні труднощі та проблеми.

Однією з ключових проблем, з якою стикаються співробітники телеканалів, є неефективне управління обладнанням. Медіа структури мають велику кількість різноманітного обладнання, такі як камери, звукове та світлове обладнання та інше необхідне для зйомок. Контролювання стану та наявності устаткування вимагає витрати значних зусиль та часу технічних асистентів. До вищезазначеного додається конфлікт щодо бронювання необхідного обладнання для проведення зйомок. Всі ці фактори можуть призвести до збільшення часу на підготовку обладнання, також можуть виникнути ситуації, коли потрібне обладнання недоступне через його відсутність або несправність після попередніх зйомок. Тому

напередодні зйомок технічному асистенту потрібно знати перелік потрібного обладнання та на який період його бронюють.

Наведені проблеми можуть вплинути на роботу телеканалу, наявність сюжетів для ефіру та стан обладнання. Тому для запобігання подібних ризиків технічні асистенти повинні знати наявність та стан обладнання, а працівники телеканалу дотримуватися графіків бронювання та завчасно попереджати про необхідність того чи іншого обладнання.

Метою даного дипломного проєкту є забезпечення ефективного управління обладнанням для виїзних зйомок, уникнення конфліктів зустрічей при бронюванні обладнання, шляхом розроблення відповідного програмного забезпечення.

Для досягнення поставленої мети сформовано перелік задач, які потрібно виконати:

- 1) аналіз ринку ПЗ для бронювання знімального обладнання;
- 2) порівняння конкурентів, виявлення їхніх сильних та слабких сторін;
- 3) визначення функціональних можливостей системи, яких потребують потенційні користувачі;
- 4) підготовка дизайну застосунку;
- 5) обґрунтування вибору стеку технологій для розроблення ПЗ;
- 6) програмна реалізація вебзастосунку для обліку обладнання телеканалу для виїзних зйомок;
- 7) тестування системи.

## **1.2. Аналіз існуючих програмних рішень**

Аналіз існуючих рішень допомагає краще зрозуміти предметну галузь, дослідити переваги та недоліки конкурентів, а також розробити стратегію розвитку власного програмного продукту.

Першим етапом у процесі аналізу є підготовка до нього, а саме пошук застосунків, які задовольняли б основні потреби потенційних користувачів.

Як виявилось, таких систем налічується невелика кількість. Це спричинено тим, що такі системи використовують компанії, які надають послуги бронювання знімальне обладнання. Телеканали ж в якості вирішення проблем наведених в попередньому підрозділі використовують або зроблене на замовлення програмне забезпечення, якого немає у відкритому доступі, або не використовують жодного специфічного інструменту. Серед наявних рішень виділено найпопулярніші: Frame Rental [1], Google Calendar [2] і Microsoft Excel [3].

### ***1.1.1. Frame Rental***

Frame Rental [1] є вебплатформою, що дозволяє людям орендувати таке фото- та відео-обладнання, як фотоапарати, об'єктиви, відеокамери, світлові прилади, звукове обладнання, мікрофони, екшн-камери GoPro. Вона надає можливість користувачу знайти потрібне йому обладнання та орендувати його на потрібний період. Однак, основний фокус Frame Rental – це надання оренди фізичним особам на короткий період, що не може бути ідеальним варіантом для масштабного підприємства, яке має великі об'єми знімального виробництва. Також для телеканалів важливо оперувати власним обладнанням, а не використовувати сторонні компанії для співпраці. Тому для телеканалів, що прагнуть зберегти незалежність та мати постійний доступ до знімальної техніки, власний парк обладнання є стратегічно важливим. Власне обладнання дозволяє телеканалам уникнути можливих проблем, пов'язаних з затримками в доставці чи відсутністю необхідного устаткування у сторонніх постачальників. Крім того, це забезпечує постійний доступ до технічного ресурсу, що є критично важливим для оперативної роботи знімальних груп і забезпечення безперервного виробничого процесу.

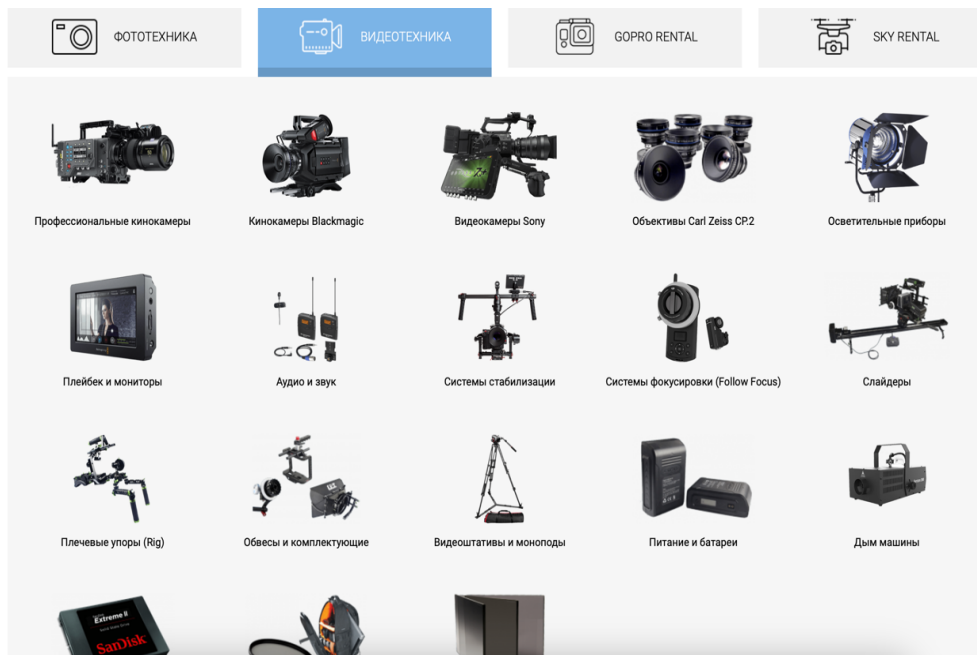


Рис. 1.1. Вибір доступних категорій відео-обладнання для оренди в застосунку Frame Rental

Переваги сервісу Frame Rental:

- 1) великий вибір фото- та відео-обладнання;
- 2) надає можливість здійснювати пошук за категоріями, такі як бренд та тип техніки, а також здійснити пошук вручну;
- 3) надання бонусів постійним клієнтам;
- 4) надає послуги доставки обладнання за вказаною адресою.

Незважаючи на наведені переваги сервісу Frame Rental, наявні і його недоліки:

- 1) є доступним лише у деяких географічних областях, що ускладнює процес отримання обладнання;
- 2) для підприємства з великим масштабом виробництва бронювання всього потрібного обладнання у сторонніх компаній ускладнює планування знімального процесу;
- 3) не забезпечує коректне відображення наявності обладнання у вебдодатку.

Отже, вебсервіс Frame Rental покриває більшу частину поставлених перед ним вимог. Серед переваг варто виділити асортимент обладнання та

зручність навігації інтерфейсом. Якщо говорити про недоліки Frame Rental, то найбільшими мінусом є складність взаємодії та укладання домовленостей з масштабними підприємствами для покриття всіх потреб знімального виробництва.

### 1.1.2. Google Calendar

Досить популярним інструментом для створення подій, що охоплюють певний період часу є електронний календар. Найпопулярнішим сервісом такого типу є Google Calendar, це безкоштовний вебзастосунок для тайм-менеджменту розроблений Google. Щоб отримати можливість використання застосунку, користувачі повинні мати акаунт Google [2].

Розглянемо як електронний календар можна використовувати для бронювання обладнання для виїзних зйомок. Маючи звичайний акаунт Google, можна з легкістю створити запис щодо бронювання на визначений час. До того ж, функція «Appointment schedule» дозволяє у календарі створити періоди часу, доступні для бронювання та розповсюдити календар поміж працівників, які мають повноваження бронювати обладнання.

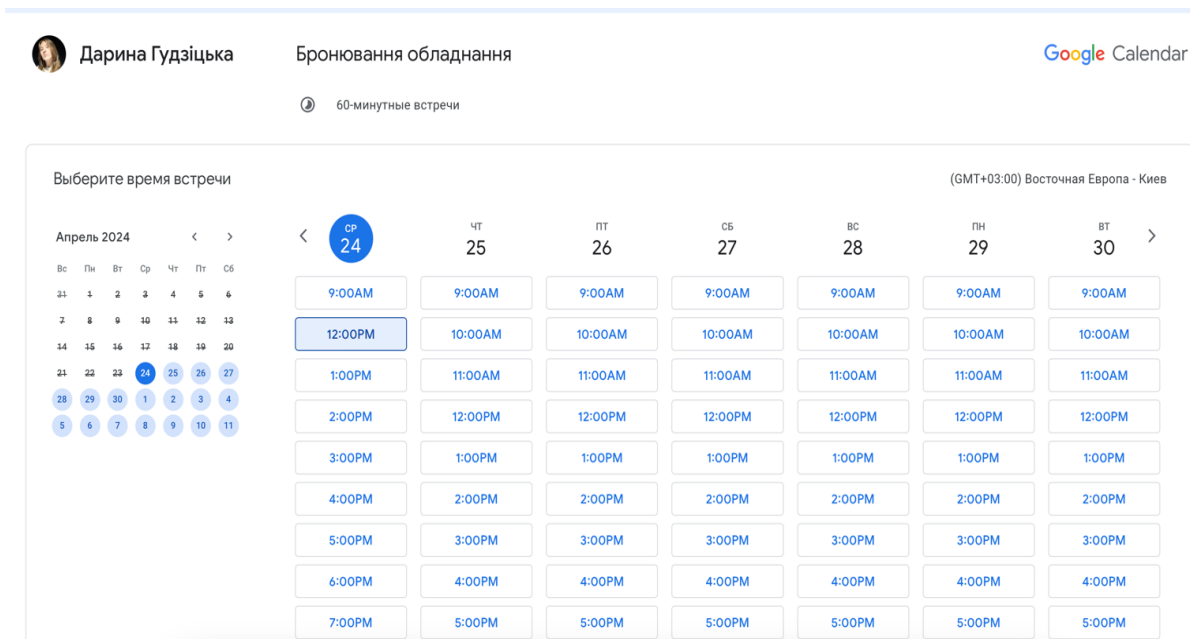


Рис. 1.2. Фрагмент сторінки з доступними проміжками часу для бронювання у Google Calendar

Для того, щоб переглядати вже створені записи про бронювання, потрібно обрати відповідний календар з доступних йому. Процес бронювання схожий на планування подій. Створюючи запис щодо бронювання, користувач може відмітити інших учасників зйомки та повний список необхідного обладнання.

Перевагами Google Calendar як інструменту для обліку обладнання є:

- 1) спільний доступ до календаря, усі працівники, що мають повноваження бронювати обладнання, матимуть доступ до нього та зможуть відслідковувати завантаженість конкретних пристроїв;
- 2) режим роботи у реальному часі;
- 3) мінімальні витрати часу на створення запису завдяки інтуїтивно зрозумілому інтерфейсу;
- 4) використання цього сервісу не потребує витрат на його придбання.

З недоліків цього сервісу виділено:

- 1) незахищеність даних, усі учасники, що мають доступ до календарю, можуть редагувати та видаляти вже існуючі записи;
- 2) інструмент не є гнучким – потрібно, або для кожного пристрою створювати окремий календар, або створити один календар зйомок, що збільшить кількість помилок та перетинів бронювання одного обладнання.

Отже, використання вебсервісу Google Calendar як інструменту для обліку обладнання має вагомі переваги, що робить його корисним інструментом для телеканалів, але за умови, що підприємство має малий об'єм телевиробництва та невелику кількість власного обладнання. В іншому випадку підприємство може стикнутися з низкою проблем, що створять перешкоди для стабільного функціонування телевиробництва.

### 1.1.3. Microsoft Excel

Досить часто у компаніях зустрічається використання інструменту Microsoft Excel для обліку обладнання.

Microsoft Excel – це програма для роботи з електронними таблицями. Вона дозволяє користувачам виконувати різні операції з даними, такі як введення, обчислення, аналіз, візуалізація та редагування даних у табличному форматі. Excel пропонує широкий спектр функцій, таких як форматування, створення графіків і діаграм, а також автоматизація обчислень за допомогою функцій [3].

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1			01.04.2024	02.04.2024	03.04.2024	04.04.2024	05.04.2024	06.04.2024	07.04.2024	08.04.2024	09.04.2024	10.04.2024	11.04.2024	12.04.2024	13.04.2024	14.04.2024	15.04.2024	16.04.2024	17.04.2024	18.04.2024	
2																					
3			09:00																		
4																					
5																					
6																					
7																					
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					
18																					
19			09:00																		
20																					
21																					
22																					
23																					
24																					
25																					
26																					
27																					
28																					
29																					
30																					
31																					
32																					
33																					

Рис. 1.3. Приклад таблиці обліку обладнанням для виїзних зйомок при використанні програми Microsoft Excel

Якщо використовувати Microsoft Excel для створення графіку бронювання обладнання, то потрібно дотримуватися алгоритму, який зробить цей процес простим і легким для всіх учасників. Спочатку технічний асистент або інша відповідальна особа створює електронну таблицю. По горизонталі відкладаються дати, на які співробітники телеканалу можуть забронювати необхідне їм обладнання, а по вертикалі – список наявного на підприємстві обладнання і години, які доступні для бронювання, наприклад, з годинним інтервалом. Особа, якій потрібно

забронювати зустріч, відкриває електронну таблицю, створену офіс-менеджером, і виділяє кольоровою заливкою клітинки, які перетинаються з потрібною датою і часом для обраного обладнання. На цьому бронювання завершено.

Під час виконання алгоритму, наведеного вище, виявлено низку недоліків:

- 1) необхідність підтримувати таблиці в актуальному стані – технічним асистентам регулярно доводиться заповнювати комірки датами кількадечної давнини та видаляти попередні бронювання, що може призвести до випадкового видалення необхідних записів;
- 2) відсутність захисту даних – всі користувачі, які мають право на редагування, можуть змінювати та видаляти записи інших користувачів;
- 3) система не є безпечною – неможливо дізнатися, хто та коли здійснив бронювання конкретного обладнання;
- 4) не підходить для використання в компаніях з великою кількістю однотипних пристроїв – в такому випадку висота таблиці займатиме занадто велику площу.

Незважаючи на велику кількість недоліків, такий спосіб бронювання має декілька переваг:

- 1) інтуїтивно зрозумілий інтерфейс, що робить процес бронювання простим та швидким;
- 2) адаптивність до змін – таблицю можна розширити, додавши години доступні для бронювання, нове обладнання та іншу необхідну інформацію.

Отже, Microsoft Excel як інструмент для обліку обладнання телеканалу для виїзних зйомок не відповідає поставленим вимогам або відповідає, маючи низку вимог. Проте у той же час програма має такі переваги, як: гнучкість до змін, простота у використанні та оновлення даних у режимі реального часу.

### 1.3. Результати проведеного аналізу

Після проведення аналізу існуючих сервісів, які досить часто користуються попитом серед підприємств, що працюють у галузі телекомунікацій та прагнуть автоматизувати процес обліку обладнання для виїзних зйомок, сформовано декілька висновків.

По-перше, слід зазначити, що хоча Frame Rental, Google Calendar і Microsoft Excel виконують основну функцію, дозволяючи співробітникам телеканалу бронювати необхідне їм обладнання для виїзних зйомок, проте рівень ефективності та зручність їх використання різні.

По-друге, жодне з цих програмних забезпечень не відповідає висунутим вимогам в повній мірі, усі вони мають значні недоліки. Наприклад, Google Calendar і Microsoft Excel не передбачають захисту інформації щодо бронювання, а також процес бронювання ускладнюється зі збільшенням кількості пристроїв у власності телеканалу. Щодо Frame Rental даний сервіс підходить для маленьких підприємств, що не мають власного обладнання.

Якщо розглядати переваги проаналізованих сервісів, то Frame Rental вирізняється поміж конкурентів своїм інтерфейсом та алгоритмом сторення бронювання. Також значною перевагою є захищеність даних – одна людина, адміністратор сервісу, має доступ до всіх бронювань, звичайні ж користувачі можуть переглядати та змінювати тільки власні запити.

У висновку можна зазначити, що жодне з розглянутих програмних забезпечень не задовольняє усі функціональні вимоги, які могли б вирішити основні проблеми потенційних користувачів системи для обліку обладнання для виїзних зйомок. Тому на даному етапі виявлено та сформовано перелік можливостей, які повинні бути наявними у ПЗ:

- 1) перегляд завантаженості усього обладнання на час, день, тиждень та місяць;
- 2) можливість пошуку конкретного обладнання за допомогою фільтрації за певними параметрами;

- 3) наявність адміністратора системи, який має доступ до всіх записів щодо бронювання;
- 4) сповіщення технічного асистента про створення, зміну та видалення бронювання;
- 5) сповіщення співробітників телеканалу щодо готовності заброньованого обладнання до видачі;
- 6) можливість забронювати декілька девайсів одночасно;
- 7) можливість редагування бронювання;
- 8) скасування бронювання;
- 9) нотифікація інформації щодо зйомки.

#### **1.4. Висновки до розділу 1**

Проведений аналіз проблеми обліку обладнання для виїзних зйомок на телевиробництві підкреслив значимість застосунків, спрямованих на спрощення цього процесу, оскільки вони вирішують проблему, пов'язану з необхідністю ефективного використання часу та ресурсів. Такі застосунки дозволяють зберігати великий обсяг даних про обладнання, його технічний стан та історію використання. Це сприяє підвищенню продуктивності та ефективності роботи зйомочної групи, а також допомагає у запобіганні можливих збоїв та неполадок під час зйомок.

Отримана інформація дозволила виявити основні проблеми існуючих рішень і, на основі цього, сформулювати функціональні вимоги до розроблюваного програмного забезпечення. Вебдодаток має потенціал об'єднати сильні сторони існуючих програмних рішень і надати нові функції, які відповідають потребам користувачів. Це створить конкурентну перевагу, забезпечить високу задоволеність користувачів і сприятиме швидкому процесу бронювання обладнання.

## **2. ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РОЗРОБЛЕННЯ**

На цьому етапі мають бути сформульовані основні вимоги до інструментів реалізації програмного продукту, проведений структурний аналіз обраних інструментів, наданий перелік функцій кожного інструменту та створена таблиця для порівняння їх за критеріями.

Система обліку обладнання телеканалу для виїзних зйомок поділяється на дві основні частини: клієнтську та серверну. Такий поділ гарантує високу ефективність роботи програми, оскільки навантаження на клієнтську частину значно зменшується. Крім того, серверна частина може авторизувати і контролювати права доступу користувачів до конфіденційної інформації, що підвищує безпеку системи, а отже, знижує ризик витоку даних.

Для забезпечення ефективної та надійної роботи всієї системи необхідно враховувати сумісність обраної технології для серверної та клієнтської частин, а також для бази даних. Успішне впровадження та функціонування програмного забезпечення залежить від гармонійної взаємодії між цими компонентами. Тому важливо підбирати технології, які не тільки відповідають потребам проєкту, але й сумісні між собою та спрощують процес розроблення, впровадження та підтримки програмного забезпечення.

### **2.1. Вибір мови програмування для реалізації серверної частини**

Мова програмування, яка використовується для реалізації серверної частини програмного забезпечення для обліку обладнання для виїзних зйомок, має відповідати наступним вимогам:

- 1) великий вибір модулів та бібліотек;
- 2) підтримка асинхронного програмування;
- 3) наявність засобів для роботи з мережевими протоколами;
- 4) масштабованість та модульність;

5) широка підтримка та активна спільнота розробників.

### **2.1.1. C#**

C# – об'єктно-орієнтована мова програмування загального призначення. Розроблена групою інженерів компанії Microsoft як мова розроблення додатків для платформи Microsoft .NET Framework і .NET Core. C# належить до родини мов із C-подібним синтаксисом. Мова має статичну типізацію, підтримує поліморфізм, перевантаження операторів (зокрема операторів явного та неявного приведення типу), делегати, атрибути, події, змінні, властивості, узагальнені типи та методи, ітератори, анонімні функції з підтримкою замикань, LINQ, винятки, коментарі у форматі XML [4].

Серед переваг C# можна зазначити:

- 1) є мовою програмування з великим екосистемою, що включає в себе багато інструментів і бібліотек для розроблення вебсайтів;
- 2) завдяки своєму компільованому характеру, вебсайти, розроблені на C#, можуть мати високу швидкість роботи та оптимальне використання ресурсів сервера;
- 3) має вбудовані механізми безпеки та відмінні механізми оброблення помилок.

Окрім переваг є і значні недоліки:

- 1) традиційно C# використовується для розроблення додатків під платформу Windows, що може створювати обмеження щодо переносимості вебсайту на інші операційні системи.

### **2.1.2. Python**

Python – це інтерпретована, інтерактивна, об'єктно-орієнтована та високорівнева мова програмування загального призначення з динамічною суворою типізацією та автоматичним управлінням пам'яттю, орієнтована на підвищення продуктивності розробника, читабельність коду, а також на забезпечення портування написаних на ній програм [5].

Вагомою перевагою Python перед іншими мовами програмування є простота використання завдяки зрозумілому та читабельному синтаксису. Python має велику кількість вбудованих функцій, що є як перевагою, так і недоліком. Перевага полягає в тому, що ці функції значно спрощують процес написання коду, оскільки розробникам не потрібно створювати власні алгоритми. Однак з цієї причини Python споживає більше пам'яті, ніж інші мови програмування. Ще одна характеристика мови програмування Python, яка впливає на продуктивність – відсутність етапу компіляції. Виконання коду інтерпретатором рядок за рядком, дозволяє розробникам отримувати миттєвий зворотній зв'язок від програми. Це означає, що вони можуть швидко вносити зміни в код і бачити результати, не чекаючи на компіляцію. Такий інтерактивний підхід не тільки сприяє швидкому розробленню додатків, але й полегшує виправлення помилок та тестування коду.

Перевагами Python можна зазначити:

- 1) зрозумілий та лаконічний синтаксис;
- 2) є мовою програмування з великою екосистемою, що включає в себе багато інструментів і бібліотек;
- 3) можливість автоматичної інтерпретації типів даних під час виконання програми;
- 4) можливість розроблення та запуску програмного забезпечення на різних операційних системах, таких як Windows, macOS та Linux, без необхідності внесення змін у вихідний код;
- 5) інтерпретованість – відсутність етапу компіляції перед запуском, що пришвидшує процес розроблення;
- 6) підтримка модульності та парадигм ООП.

Крім переваг мови Python, наявні і її недоліки, серед яких:

- 1) динамічна типізація даних вимагає постійної перевірки типів змінних, що негативно впливає на швидкість;

- 2) Python не оптимізований для ефективного використання пам'яті і може використовувати більше пам'яті, ніж інші мови програмування;
- 3) через свою архітектуру мова Python не підтримує багатопоточність – натомість використовується багатопроцесорність.

### ***2.1.3. Javascript***

JavaScript – об'єктно-орієнтована клієнтська мова, що підтримується додатками, які займаються дизайном вебсайтів, стала ще більш популярною серед розробників, коли з'явилася технологія AJAX, що призвело до нового етапу в розробці вебсайтів [6]. Свою ключову роль вона відіграє у світі фронтенду, за допомогою JavaScript сторінки вебсайтів стали більш інтерактивними.

Однак JavaScript нерідко використовується як інструмент для розроблення серверної частини вебдодатку. Завдяки середовищу виконання Node.js розширились можливості JavaScript, перетворивши його з технології для побудови лише клієнтської частини програми на універсальну мову програмування. Використання JavaScript на базі платформи Node.js значно спрощує процес побудови як клієнтської, так і серверної частин програми, що пришвидшує розроблення додатка.

Серед переваг JavaScript можна зазначити:

- 1) можливість її використання для створення як серверної, так і клієнтської частини застосунку;
- 2) підтримка асинхронного програмування;
- 3) підтримка модульності, яка полегшує та пришвидшує процес розроблення.

Проте наявні і недоліки JavaScript, такі як:

- 1) через динамічну типізацію, можуть виникати помилки виконання, якщо типи даних не перевіряються на етапі компіляції;

2) JavaScript вразливий до певних типів атак, таких як впровадження коду через вебдодатки або перехоплення інформації про користувача [7].

Кожна з перерахованих вище мов програмування надає розробникам всі необхідні інструменти для створення надійної серверної частини програмного забезпечення для обліку обладнання для виїзних зйомок. Після детального аналізу прийнято рішення, що оптимальним варіантом є використання Python з Django фреймворком.

## **2.2. Вибір технологій для розроблення клієнтської частини**

JavaScript є провідною мовою для створення динамічних та інтерактивних вебсайтів і відіграє фундаментальну роль у сучасній веброзробці. Щоб спростити структуру проєкту, оптимізувати обробку даних і підвищити загальну продуктивність, веброзробники активно використовують фреймворки JavaScript [8].

Наразі веброзробникам доступні декілька фреймворків JavaScript, основними з яких є React.js та Vue.js [9]. Ці фреймворки надають готові рішення для типових завдань веброзробки і роблять процес створення вебсайтів більш ефективним та організованим.

Важливу роль при розробці вебінтерфейсів, окрім JavaScript-фреймворків, відіграють CSS-фреймворки, такі як Bootstrap. Завдяки ним значно спрощується процес створення лаконічних та інтуїтивно зрозумілих користувацьких інтерфейсів, водночас гарантуючи високу швидкість рендерингу сторінок і оптимізоване використання пам'яті [10].

### **2.2.1. React.js**

React.js – це JavaScript бібліотека для розроблення інтерфейсів користувача. React.js дозволяє розробникам будувати вебдодатки з високим рівнем взаємодії та динамічним змінним контентом. Завдяки використанню компонентного підходу до розроблення, в якому інтерфейси розділені на

невеликі, незалежні блоки – компоненти, React.js дозволяє легко модулювати код і повторно використовувати компоненти в різних частинах програми [11]. Це спрощує розроблення, підтримку та розширення проєкту, дозволяючи розробникам ефективно керувати окремими частинами програми та впроваджувати зміни, не впливаючи на інші компоненти.

Основною особливістю React.js є Virtual DOM (Document Object Model), яка дозволяє автоматично оновлювати інтерфейс при зміні даних та вирішує проблему швидкості рендерингу.

Отже перевагами React.js є:

- 1) модульна структура, яка спрощує розроблення, підтримку та розширення проєкту;
- 2) Virtual DOM, що покращує продуктивність шляхом автоматичного оновлення інтерфейсу;
- 3) велика кількість документації та багато сторонніх бібліотек.

Однак, існують і недоліки:

- 1) потребує використання додаткових розширень та бібліотек, що збільшує розмір проєкту та складність управління залежностями.

### **2.2.2. Vue.js**

Vue.js – JavaScript-фреймворк для створення користувацьких інтерфейсів вебзастосунків. Vue.js також використовує компонентний підхід до розроблення, подібний до React.js, де інтерфейс розділений на невеликі незалежні блоки. Це спрощує розроблення та підтримку проєкту, оскільки код можна легко модифікувати, а компоненти можна повторно використовувати в різних частинах програми [12]. Ще однією важливою особливістю Vue.js є Virtual DOM, який, подібно до того, як працює React.js, може ефективно та автоматично оновлювати інтерфейс при зміні даних, покращуючи продуктивність.

Переваги Vue.js:

- 1) зрозумілий та лаконічний синтаксис;

2) гнучкість Vue.js дозволяє використовувати його для розроблення як простих, так і складних вебдодатків.

3) підтримка модульної структури та наявність Virtual DOM.

Крім переваг фреймворку Vue.js, наявні і його недоліки, серед яких:

- 1) Vue.js достатньо молода технологія, тому ще не має такої великої спільноти та кількості документації як React.js;
- 2) менша можливість інтеграції у великі проекти через надмірну гнучкість [13].

### **2.2.3. Javascript і Bootstrap**

Javascript – мова програмування, що створена для розроблення динамічних та інтерактивних вебсайтів. Вона широко використовується для створення динамічного контенту, обробки подій, маніпулювання DOM (Document Object Model) та асинхронних запитів до серверів за допомогою таких технологій, як AJAX [14].

Bootstrap – це потужний інструментарій для створення інтерфейсів, який включає готові CSS- і JavaScript-компоненти для швидкого створення адаптивних та естетичних вебінтерфейсів. Bootstrap надає набір готових CSS-класів для стилізації багатьох елементів HTML, а також JavaScript-компонентів, що дозволяє досягнути однотипності їх зовнішнього вигляду та поведінки [15].

Поєднання JavaScript і Bootstrap є потужним інструментом. Ця комбінація гарантує швидке розгортання, що робить процес створення клієнтської частини значно простішим.

Перевагами комбінації JavaScript і Bootstrap є:

- 1) швидкість проєктування, завдяки наборам готових компонентів Bootstrap та їх розширення а допомогою JavaScript;
- 2) велика кількість документації.

Недоліком такої комбінації є залежність від власних файлів Bootstrap та їх розміру, що може вплинути на продуктивність.

З огляду на те, що розроблювана система обліку обладнання телеканалу для виїзних зйомок є внутрішнім продуктом підприємства і не вимагає складних інтерфейсів, поєднання JavaScript і Bootstrap залишається найкращим варіантом для розроблення клієнтської частини вебдодатку. JavaScript дозволяє додавати необхідну функціональність та інтерактивність, таку як форми введення, перевірка даних і взаємодія з користувачем. Bootstrap, з іншого боку, надає готові стилі та компоненти, що полегшує організацію та стилізацію інтерфейсу додатку без великих затрат часу.

Використання цієї комбінації технологій забезпечує швидке розроблення, єдиний зовнішній вигляд інтерфейсу та надійну функціональність, необхідну для системи обліку польового знімального обладнання телеканалу. Такий підхід дозволяє зосередитися на вирішенні конкретних завдань і потреб бізнесу, зберігаючи при цьому оптимальне співвідношення між ефективністю проектування та якістю кінцевого результату.

## **2.3. Вибір СКБД**

### ***2.3.1. Вибір між реляційною та не реляційною базою даних***

Важливою складовою розроблення вебдодатку для обліку обладнання телеканалу для виїзних зйомок є ефективне управління та збереження даних, що ставить гостре питання вибору відповідної бази даних. На сьогоднішні існує два основних типи баз даних: реляційні (SQL) та не реляційні бази даних (NoSQL).

Реляційні бази даних – це тип баз даних, в яких дані організовані у вигляді таблиць. Кожна таблиця має стовпці, що представляють різні атрибути даних, і набір рядків, що представляють конкретні записи. Реляційна база даних використовує спеціальні зв'язки між таблицями, щоб

зберігати залежності між даними [16]. Такий тип баз даних підходить додаткам, де присутні складні взаємозв'язки між різними типами даних. Вони вимагають систематичного підходу до організації та моделювання структури даних, щоб забезпечити ефективну та надійну роботу з інформацією. Однак з великими обсягами даних можуть виникати проблеми з продуктивністю.

Не реляційні або NoSQL бази даних зберігають та організують дані за відмінними моделями, ніж реляційна модель таблиць і стовпців, характерна для SQL баз даних. Ці бази даних застосовуються у випадках, коли необхідно опрацьовувати великі обсяги різноманітних за структурою даних або з даними, які постійно змінюються. Вони підходять там, де потрібна висока масштабованість і продуктивність, а також там, де важливо швидко реагувати на мінливі вимоги до даних [17]. Однак, структура не реляційної бази даних може бути складною для аналізу та її розуміння.

При виборі бази даних для вебзастосунку обліку обладнання телеканалу для виїзних зйомок слід враховувати наступні фактори:

- 1) додаток може потребувати зберігання складних та структурованих даних, які мають численні взаємозв'язки між собою; щоб задовольнити таку вимогу підходять реляційні бази, вони забезпечують потужні можливості для зберігання і управління складними даними та їх взаємозв'язками.
- 2) додаток може потребувати складних запитів та аналітичних операцій для виконання фільтрації та сортування; реляційні бази даних надають потужну SQL-мову та оптимізатор запитів, що дозволяє ефективно виконувати різноманітні запити.

Враховуючи вищезазначені переваги, реляційна база даних є розумним вибором для вебдодатку для обліку обладнання телеканалу для виїзних зйомок. Надаючи структуровані та організовані дані у вигляді таблиць і зв'язків між таблицями, реляційні бази даних сприяють простоті

управління та обробки інформації, що є важливим фактором для такого типу додатків.

### ***2.3.2. PostgreSQL***

PostgreSQL – потужна реляційна СКБД з відкритим вихідним кодом, яка володіє широкими можливостями і використовується як у невеликих стартапах, так і у великих корпораціях.

Перевагами даної СКБД є:

- 1) підтримує широкий спектр типів даних, включаючи цілочисельні, числові, логічні, символічні, текстові, дату/час і мітки часу, а також різні формати даних, таких як XML, CSV, JSON [18];
- 2) підтримує повнотекстовий пошук, який дозволяє користувачам шукати текстові дані в базі даних;
- 3) розширюваність: є можливість додавати власні оператори, індекси і перетворення типів.

Однак, PostgreSQL має недолік:

- 1) у порівнянні зі своїми конкурентами, більш вимогливий до ресурсів при великих обсягах даних та складних запитах.

### ***2.3.3. MySQL***

MySQL – одна з найпопулярніших відкритих об'єктно-реляційних СУБД, яка широко використовується у веброзробці та великих корпоративних системах.

Перевагами MySQL є:

- 1) підтримує різні режими реплікації та кластеризації, що дозволяє масштабувати базу даних під потреби різних за масштабом проєктів;
- 2) має багатий набір функціональності, включаючи різноманітні типи даних, оптимізатор запитів [19];
- 3) має вбудовані інструменти для забезпечення безпеки даних.

Недоліки:

- 1) відсутність вбудованої підтримки JSON;
- 2) низька продуктивність при роботі з великою кількістю одночасних запитів.

У результаті проведеного аналізу виявлено, що обидві реляційні СКБД відповідають поставленим вимогам. Однак, прийнято рішення використовувати PostgreSQL для створення додатку. Зручність у роботі з неструктурованими даними, повна підтримка повнотекстового пошуку та можливість використання JSON переконали нас у виборі цієї системи як оптимального рішення.

#### **2.4. Висновки до розділу 2**

Для серверної частини вебзастосунку обрано фреймворк Django на мові програмування Python через його надійність, масштабованість і швидкість розроблення застосунків. Python – популярна мова програмування з багатою екосистемою та великою кількістю розробників, що робить її гарним вибором для проектів веброзробки. Пропонуючи повноцінний стек технологій веброзробки, Django дозволяє розробникам ефективно створювати вебдодатки, використовуючи широкий спектр інструментів і модулів.

Для клієнтської частини вебзастосунку обрано комбінацію технологій JavaScript та Bootstrap, оскільки JavaScript має велике ком'юніті та велику екосистему, в той час як Bootstrap дозволяє швидко та ефективно створювати естетично привабливі та адаптивні дизайни вебсторінок з мінімальними зусиллями.

Реляційна база даних PostgreSQL обрана для зберігання та керування даними для вебзастосунку, оскільки вона може забезпечити структуровані та ефективні запити завдяки широкому спектру функцій, які дозволяють ефективно управляти даними та обробляти їх у реляційній моделі.

### **3. СТРУКТУРНО-АЛГОРИТМІЧНА ОРГАНІЗАЦІЯ СИСТЕМИ**

#### **3.1. Опис вимог до програмного забезпечення**

Враховуючи важливість функціональних вимог вебзастосунку, які описують необхідні функції та особливості його роботи, стає очевидним, що визначення та управління вимогами є критично важливими елементами при розробленні програмного забезпечення. Ці вимоги дозволяють оцінити відповідність вебзастосунку поставленим завданням і досягнення поставлених цілей. Саме від чіткого та детального опису вимог залежить якість та успішність реалізації кінцевого продукту.

Визначення вимог на ранніх етапах розроблення є фундаментом для створення успішного програмного продукту, що повністю відповідає потребам користувачів та стандартам індустрії. Чітке формулювання цих вимог стане запорукою вдалої реалізації проєкту та забезпечить високу якість кінцевого продукту.

Для успішного виконання запланованих завдань щодо розроблення програмного забезпечення для застосунку з обліку обладнання телеканалу для виїзних зйомок були сформульовані функціональні вимоги, які враховують специфіку різних ролей користувачів. Оскільки цей тип програмного забезпечення є вузькоспрямованим та, як правило, розробляється на замовлення підприємства, система передбачає можливість взаємодії лише з авторизованими користувачами, які мають такі ролі як «Працівник телеканалу», «Технічний асистент», «Адміністратор». У таблицях 3.1-3.4 (див.нижче) наведено перелік визначених функціональних вимог до розроблюваного вебзастосунку для всіх вищезазначених ролей користувачів.

Таблиця 3.1

Функціональні вимоги до роботи вебзастосунку з неавторизованими користувачами

Код вимоги	Опис вимоги
F-1	Користувач повинен мати можливість увійти в систему використовуючи адресу електронної пошти (email) та пароль, на сторінці входу в систему.
F-2	Користувач повинен мати можливість отримати повідомлення про некоректно введені дані.

Таблиця 3.2

Функціональні вимоги до роботи вебзастосунку з авторизованими користувачами в ролі «Працівник телеканалу»

Код вимоги	Опис вимоги
F-3	Користувач повинен мати можливість вийти з системи.
F-4	Користувач повинен мати змогу переглянути власний профіль.
F-5	Користувач повинен мати можливість переглядати список усього обладнання, яке містить таку інформацію: назву, характеристики, фото, наявність у потрібну дату.
F-6	Користувач повинен мати можливість здійснювати фільтрацію за типом обладнання та його характеристиками.
F-7	Користувач повинен мати можливість пошуку потрібного обладнання за назвою.
F-8	Користувач повинен мати змогу перейти на сторінку обраного зі списку обладнання.

F-9	Користувач повинен мати змогу створити бронювання на дату, коли необхідне обладнання є доступним.
F-10	Користувач повинен мати змогу редагувати власне бронювання, а саме: <ul style="list-style-type: none"> <li>- перенести його на іншу дату, коли необхідне обладнання є доступним;</li> <li>- вилучити певні позиції з бронювання.</li> </ul>
F-11	Користувач повинен мати можливість відмінити власне бронювання.
F-12	Користувач повинен мати можливість отримати повідомлення про готовність його бронювання.

Таблиця 3.3

Функціональні вимоги до роботи вебзастосунку з авторизованими користувачами в ролі «Технічний асистент»

Код вимоги	Опис вимоги
F-13	Користувач повинен мати можливість додавати нове обладнання до системи, а саме вносити інформацію про: назву, характеристики, фото, стан.
F-14	Користувач повинен мати можливість редагувати наявне у системі обладнання, а саме змінювати інформацію про: назву, характеристики, фото, стан.
F-15	Користувач повинен мати можливість видаляти з системи обладнання.
F-16	Користувач повинен мати можливість переглядати список усіх бронювань.
F-17	Користувач повинен мати можливість здійснювати фільтрацію обладнання за датою.

F-18	Користувач повинен мати можливість здійснювати підтвердження готовності бронювання до видачі.
F-19	Користувач повинен мати можливість робити та редагувати записи щодо ремонту обладнання.

Таблиця 3.4

Функціональні вимоги до роботи вебзастосунку з авторизованими користувачами в ролі «Адміністратор»

Код вимоги	Опис вимоги
F-20	Користувач повинен мати можливість реєструвати нових користувачів у системі.
F-21	Користувач повинен мати можливість редагувати інформацію про наявних користувачів у системі.
F-22	Користувач повинен мати можливість видаляти дані про наявних користувачів із системи.
F-23	Користувач повинен мати можливість додавати та редагувати інформацію щодо зйомки.
F-24	Користувач повинен мати можливість видаляти інформацію щодо наявної у системі зйомки.

Нефункціональні вимоги вебзастосунку визначають критерії, що стосуються його характеристик, продуктивності та надійності, і дозволяють зробити висновок про те, наскільки вебзастосунок задовольняє очікування користувачів щодо загального рівня та ефективності. Ці вимоги орієнтовані на забезпечення стабільної та безперебійної роботи системи, її здатності обробляти великий обсяг даних і користувачів, а також на відповідність стандартам безпеки та доступності.

Нефункціональні вимоги до розроблювального застосунку були наведені у таблиці 3.5.

Таблиця 3.5

Нефункціональні вимоги до вебзастосунку

Код вимоги	Опис вимоги
NF-1	Користувацький інтерфейс повинен коректно відображатися телефонах, планшетах та комп'ютерах.
NF-2	Паролі користувачів повинні зберігатись у захешованому вигляді SHA-256.
NF-3	Клієнтська частина повинна підтримуватись браузерами Google Chrome, Mozilla Firefox, Safari та Microsoft Edge.
NF-4	Вебзастосунок має працювати з наступними мінімальними вимогами до пристрою: <ul style="list-style-type: none"><li>- 2 ГБ оперативної пам'яті;</li><li>- процесор з тактовою частотою 1 ГГц.</li></ul>
NF-5	Має використовуватися база даних PostgreSQL.

### 3.2. Опис структурної організації застосунку

Розроблюваний вебзастосунок має клієнт-серверну архітектуру, що означає, що він складається з клієнтської та серверної частин. Клієнтська частина відповідає за взаємодію з користувачем та відображення інформації на його пристрої. Серверна частина обробляє запити від клієнтів, реалізує логіку застосунку та доступ до бази даних, яка також може бути використана для збереження інформації. Структура вебзастосунку зображена на рис. 3.1.

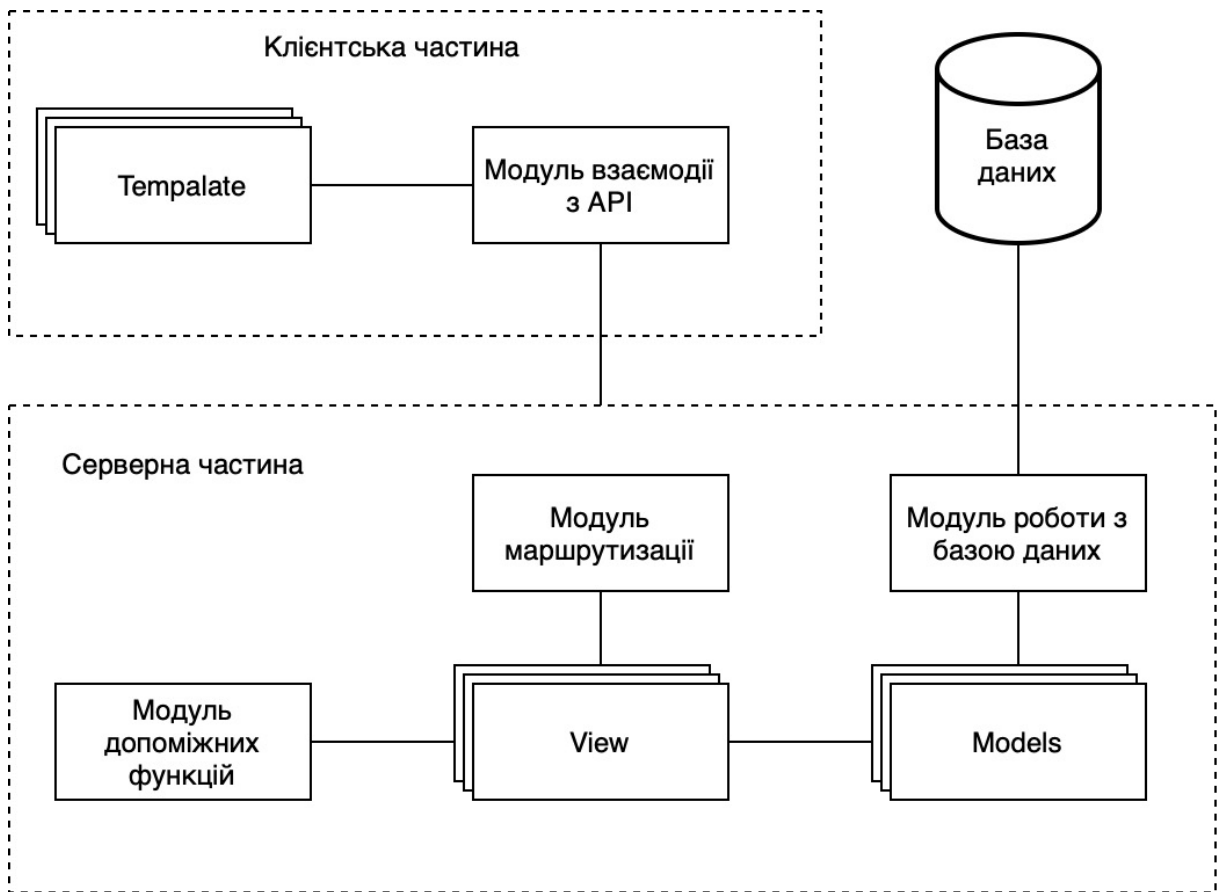


Рис. 3.1. Загальна архітектура застосунку

Розглянемо серверну частину застосунку. Обраний фреймворк Django використовує шаблон проектування MTV, що поділяється на три модулі: Model – Template – View. Кожен з цих компонентів відповідає за свої функціональні обов'язки у процесі розроблення та функціонування вебзастосунку.

Маршрутизація є ключовим завданням серверної частини вебзастосунку, що відповідає за співвіднесення URL-адрес з відповідними обробниками запитів. У фреймворку Django цю функціональність реалізовано за допомогою URL-конфігурацій. URL-конфігурації описують шаблони URL-адрес та зв'язують їх з відповідними функціями оброблення запитів [20].

Model (моделі) описують структуру даних, що зберігаються в базі даних, та реалізують методи для взаємодії з нею. Використання Django ORM (Django Object Relational Mapper) забезпечує об'єктно-орієнтований

інтерфейс для роботи з даними. Це дозволяє взаємодіяти з базою даних у контексті об'єктно-орієнтованого програмування [21].

View (види) реалізують бізнес-логіку додатку та обробляють запити, які надходять до сервера. Вони отримують дані з запиту, взаємодіють з моделями та іншими сервісами, і повертають відповідь клієнту. Види можуть використовувати різні HTTP-методи для опрацювання різних типів запитів, такі як GET для отримання даних і POST для створення нових записів.

Допоміжні функції є корисним інструментом для розширення можливостей шаблонів MVT. Вони дозволяють виносити логіку з шаблонів, роблячи їх більш читабельними та легко підтримуваними. За рахунок допоміжних функцій можна реалізувати додаткову обробку даних, логічні перевірки та інтеграції зі сторонніми API, зберігаючи правила чистого коду та гнучкість шаблонів MVT.

Модуль взаємодії з базою даних реалізує механізми зберігання, отримання, зміни та видалення даних у базі даних. Він взаємодіє зі структурою даних, визначеною в моделях, та здійснює виконання SQL-запитів до бази. Цей модуль забезпечує відокремлення рівня доступу до даних від інших компонентів додатку, що сприяє підтримці коду та робить роботу з базою даних більш простою та ефективною.

Також, структура серверної частини вебзастосунку для обліку обладнання телеканалу для виїзних зйомок може бути організована в кілька основних модулів:

- 1) модуль для управління даними про обладнання;
- 2) модуль авторизації, що реалізує механізми авторизації, включаючи перевірку прав доступу та захист ресурсів;
- 3) модуль роботи з бронюваннями, дозволяє користувачам переглядати дані про доступне обладнання, робити бронювання, відмінити їх, а також отримувати підтвердження про успішно зареєстровані бронювання.

Клієнтська частина вебзастосунку складається з двох основних компонентів.

Перший модуль – `template` (шаблони), які відповідають за відображення даних користувачеві у вигляді HTML-сторінок. Вони використовують спеціальний синтаксис для динамічного вставлення змінних та логіки відображення, що дозволяє створювати сторінки змісту, які адаптуються до конкретних даних [22]. Використання шаблонів сприяє чистоті та організованості коду, оскільки вони відділяють логіку від представлення.

Другий модуль – модуль взаємодії з API, який відповідає за взаємодію з серверною частиною додатку через його API. Цей модуль може відправляти HTTP-запити до сервера за допомогою вбудованих функцій або сторонніх бібліотек, отримувати відповіді та обробляти їх. Його роль полягає в забезпеченні доступу до даних і оновленні їх на сервері, що дозволяє зберігати інформацію актуальною та забезпечує взаємодію клієнтської частини з сервером.

### **3.3. Структура таблиць бази даних**

Перш ніж переходити до проєктування бази даних, необхідно визначити структуру даних. У даному підрозділі пояснювальної записки здійснено аналіз та визначено перелік об'єктів та детально описано їх поля.

#### **1) Зйомка (Shoot):**

- `id` (первинний ключ) – унікальний ідентифікатор зйомки;
- `title` – назва зйомки;
- `date` – дата зйомки;
- `description` – опис зйомки;
- `userid` – ідентифікатор користувача, який створив зйомку.

#### **2) Бронювання (Booking):**

- `id` (первинний ключ) – унікальний ідентифікатор бронювання;
- `hours` – тривалість бронювання в годинах;

- start date – дата початку бронювання;
- end date – дата закінчення бронювання;
- equipmentId – ідентифікатор обладнання, яке заброньовано;
- shootId – ідентифікатор зйомки, до якої прив'язане бронювання;
- userId – ідентифікатор користувача, який здійснив бронювання;
- description – примітки до бронювання;
- status – статус бронювання, приймає значення «Створено», «Виконується», «Виконано», «Повернуто».

### 3) Обладнання (Equipment):

- id (первинний ключ) – унікальний ідентифікатор обладнання;
- title – назва обладнання;
- description – опис обладнання;
- photo – фотографія обладнання;
- typeId – ідентифікатор типу обладнання;
- serialId – серійний номер обладнання.

### 4) Тип обладнання (Equipment type):

- id (первинний ключ) – унікальний ідентифікатор типу обладнання;
- type – назва типу обладнання.

### 5) Користувач (User):

- id (первинний ключ) – унікальний ідентифікатор користувача;
- first name – ім'я користувача;
- last name – прізвище користувача;
- position – посада користувача;
- email – адреса електронної пошти користувача;
- phone – номер телефону користувача;
- password – пароль користувача.

### 6) Посада (Position):

- id (первинний ключ) – унікальний ідентифікатор посади;
- job title – назва посади;

- is admin – прапор, який вказує, чи є користувач адміністратором, потрібен для дозволу редагування списку обладнання та зміни статусу бронювання.

### 7) Історія ремонту (Repair history):

- id (первинний ключ) – унікальний ідентифікатор запису в історії ремонту;
- equipmentId – зовнішній ключ, ідентифікатор обладнання, до якого застосовано ремонт;
- date – дата проведення ремонту;
- description – опис робіт;
- status – статус ремонту, приймає значення «Виконується», «Виконано».

ER-діаграму взаємозв'язків між таблицями описаними вище зображено на рисунку 3.2.

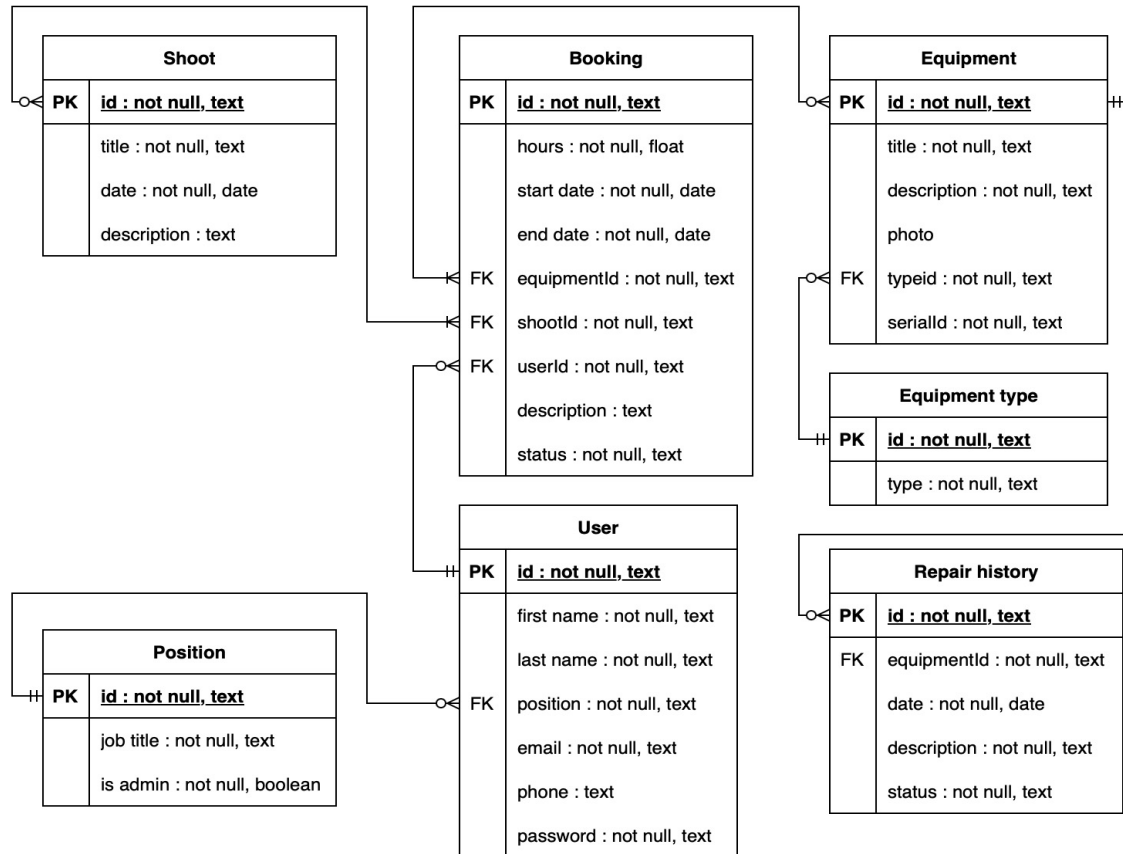


Рис. 3.2. Структура бази даних. ERD-діаграма

### 3.4. Алгоритмічне забезпечення застосунку

Для зображення функціональних реалізованих функціональних можливостей розроблюваного вебзастосунку побудовано діаграму варіантів використання (use case diagram). Такі діаграми використовуються для представлення взаємодії користувачів з системою. Вона показує різні способи використання системи для досягнення певних цілей або виконання завдань. Діаграма варіантів використання програмного забезпечення для застосунку з обліку обладнання телеканалу для виїзних зйомок включає в себе чотири актори: неавторизований користувач, адміністратор, технічний асистент, працівник телеканалу, варіанти використання та зв'язки між цими сутностями. UML-діаграму варіантів використання зображено на рисунку 3.3.

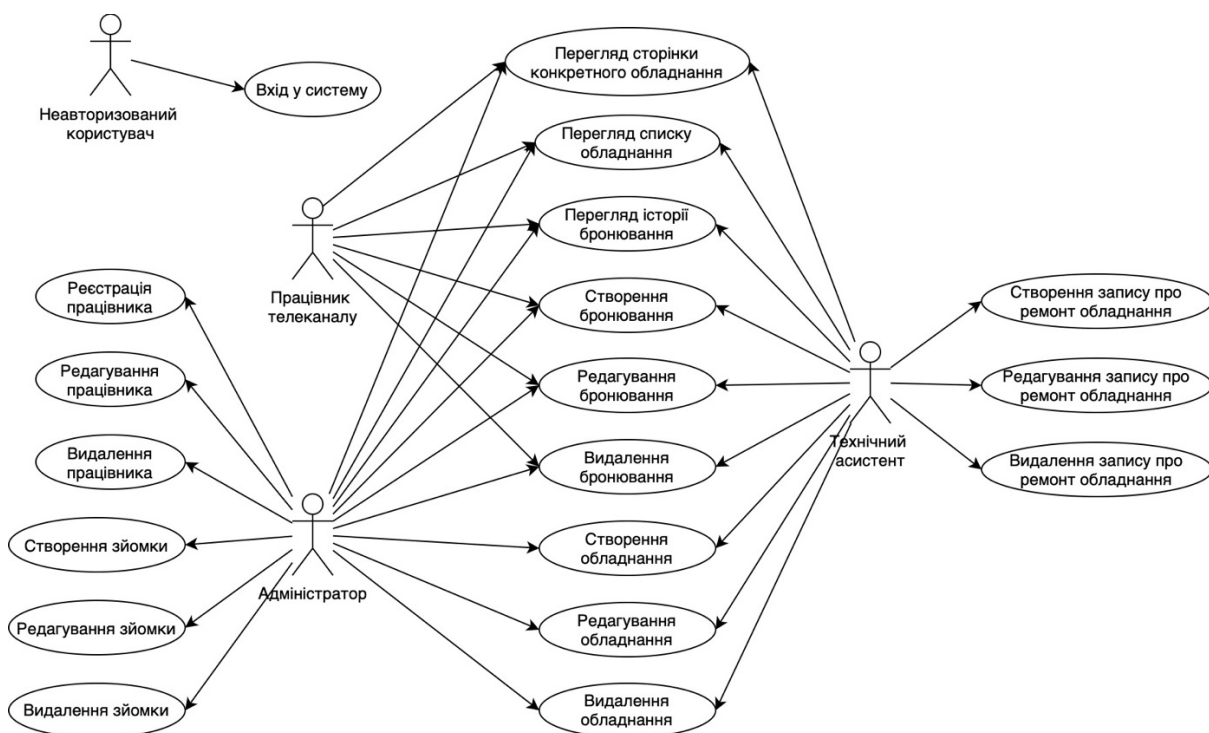


Рис. 3.3. UML-діаграма варіантів використання

Також варто розглянути алгоритмічне забезпечення створеного вебдодатку.

Основним алгоритмом у вебзастосунку для обліку обладнання телеканалу, призначеного для виїзних зйомок, є перевірка доступності обладнання у заданий період часу. Цей алгоритм застосовується у декількох місцях вебзастосунку. Перш за все цей алгоритм застосовується на сторінці, де відображений список доступного обладнання. При виборі користувачем дати або діапазону дат система перевіряє доступність кожного елементу обладнання на цей період. Після цього користувачу відображається список доступного обладнання для вибраних дат, допомагаючи йому зрозуміти, яке обладнання є доступним у потрібну йому дату.

Алгоритм також застосовується на сторінці, присвяченій конкретному елементу обладнання. Тут користувач може перевірити доступність саме цього обладнання на бажаний період. При виборі дати система виконує перевірку наявності конфліктів з існуючими бронюваннями. Якщо обладнання доступне, користувач отримує підтвердження та можливість забронювати його. У випадку, якщо обладнання вже зайняте, користувачу пропонуються альтернативні дати, коли воно буде вільне, або видалити зі списку бронювання обладнання, яке конфліктує з щойно обраним.

Ще одне місце застосування алгоритму – сторінка для створення та редагування бронювання. Коли користувач встановлює межі періоду бронювання, система перевіряє можливість створення бронювання з урахуванням доступності усього списку бронювань на вибрану дату. Користувачу відображаються доступні часові межі для бронювання, що дозволяє уникнути конфліктів з існуючими бронюваннями і забезпечує ефективне управління ресурсами.

На рисунку 3.4 представлено блок-схему алгоритму перевірки доступності обладнання на певний період часу.

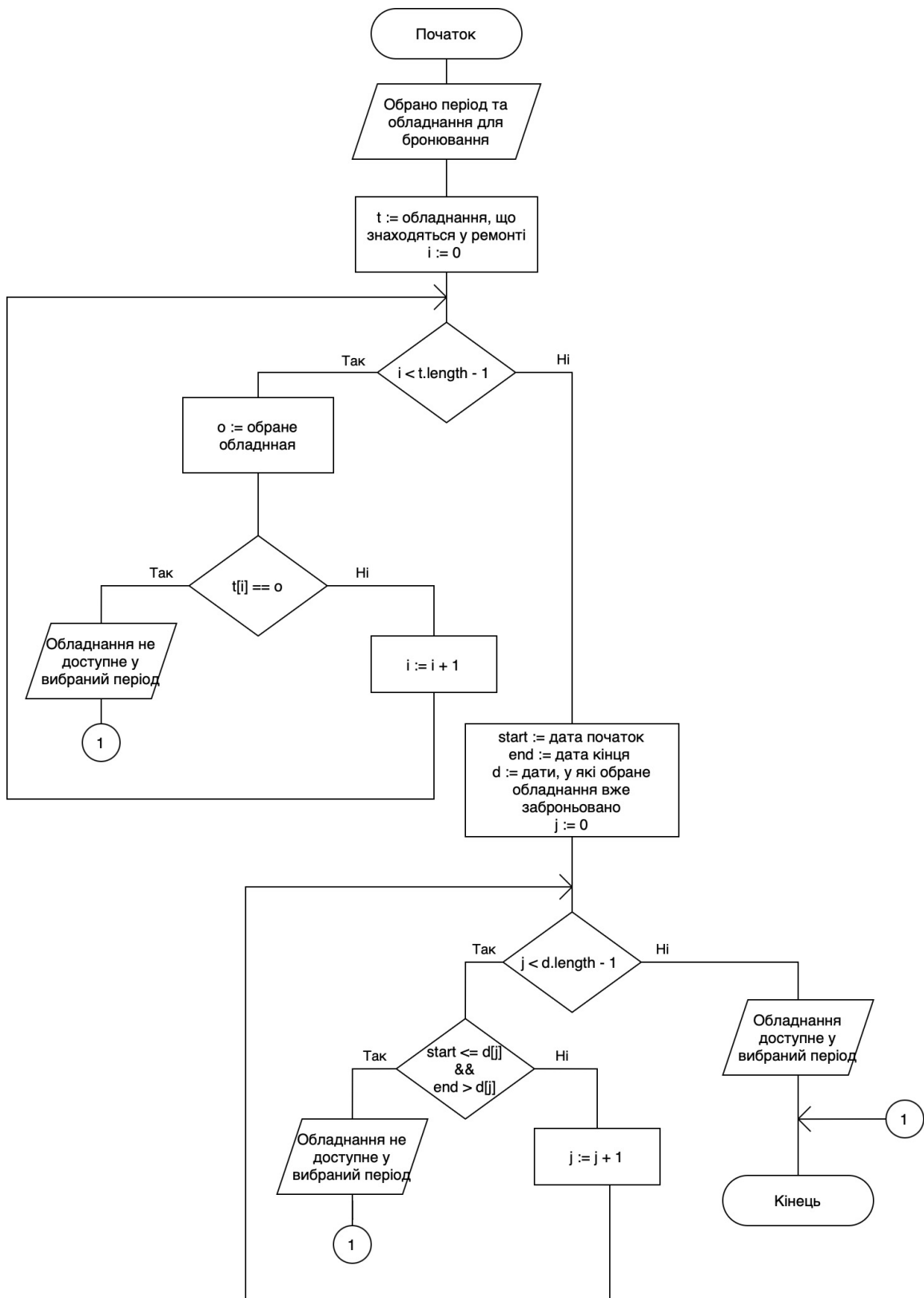


Рис. 3.4. Алгоритм перевірки доступності обладнання на певний період часу. Блок-схема

Алгоритм роботи системи бронювання включає такі основні кроки:

- 1) Вибір користувачем дати початку бронювання та обладнання.
- 2) Перевірка системою, чи знаходиться обладнання в ремонті.
  - Якщо обладнання в ремонті, користувач отримує повідомлення про неможливість створення бронювання.
- 3) Перевірка наявності існуючих бронювань для обраного обладнання на зазначену дату.
  - Якщо в базі немає записів про зайнятість обладнання на обрану дату, система дозволяє користувачу створити бронювання.
  - Якщо обладнання зайняте, користувачу повідомляється про неможливість створення бронювання.

### **3.5. Висновки до розділу 3**

У третьому розділі встановлено, що вебзастосунок неавторизовані користувачі можуть використовувати лише для авторизації, вся функціональність доступна лише авторизованим користувачам, що отримали ролі «Працівник телеканалу», «Технічний асистент», «Адміністратор». Також була створена діаграма варіантів використання, яка описує функціональні можливості програми відносно цих ролей.

На основі можливостей, якими користуються користувачі з відповідними ролями, були сформульовані функціональні вимоги до розроблюваного вебзастосунку. Також були визначені нефункціональні вимоги, що описують додаткові характеристики вебзастосунку.

Була описана структура організації вебзастосунку, зокрема модулі, які будуть присутні в клієнтській та серверній частинах, а також спосіб взаємодії між ними та з базою даних. Після цього був визначений перелік сутностей, які будуть присутні в базі даних та була побудована ERD-діаграма бази даних, яка відображає всі сутності, поля та зв'язки між таблицями.

Був розроблений алгоритм перевірки доступності обладнання на певний період часу, що передбачає аналіз доступності кожного елемента обладнання на заданий проміжок часу. Після цього була побудована його блок-схема для зручної візуалізації та розуміння логіки його роботи.

## 4. ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ВЕБЗАСТОСУНКУ

### 4.1. Реалізація серверної частини

У серверній частині розроблювального проєкту використовувався фреймворк Django, який містить безліч інструментів для реалізації бізнес-логіки.

Для роботи з базою даних в проєкті використовується ORM Django, що значно спрощує взаємодію з базою. Замість написання SQL запитів, ORM дозволяє використовувати Python-код для модифікації бази даних з використанням об'єктно-орієнтованого підходу. Це спрощує процес розроблення, покращує читабельність коду та зменшує ймовірність виникнення помилок.

Django моделі були використані для визначення структур даних, в яких кожен клас представляє таблицю в базі даних. Використовуючи відповідні поля в моделі, можна легко встановлювати зв'язки між таблицями. Крім того, в Django реалізована можливість автоматичного створення і оновлення схеми бази даних, при внесенні змін до структури моделей.

Одним з важливих аспектів розроблювального проєкту є безпечне зберігання паролів. Для цього було використано хеш-алгоритм SHA-256, який є частиною модуля шифрування hashlib в Python. SHA-256 генерує унікальний хеш для кожного введеного пароля і є стійким до зворотного відновлення. Процес хешування складається з декількох кроків. Спочатку пароль користувача перетворюється на байтовий рядок для правильної роботи алгоритму. Потім цей рядок хешується за допомогою функції `hashlib.sha256()`, яка повертає хеш-об'єкт. Метод `hexdigest()` цього об'єкта використовувався для отримання хешу у вигляді шістнадцяткового рядка, який потім зберігався в базі даних.

Для підвищення безпеки були використані значення солі. Сіль (salt) - це випадково згенерований рядок, який додається до пароля перед

хешуванням. Це додаткове значення значно ускладнювало потенційні атаки, такі як атаки "таблиця веселки" [23]. Кожен пароль мав унікальну сіль, яка зберігалася в базі даних разом з хешованим паролем. Завдяки такому підходу до хешування паролів було досягнуто високого рівня безпеки даних користувачів, відповідності найсучаснішим стандартам інформаційної безпеки та захисту від багатьох видів, спрямованих на викрадення паролів. Це дозволило створити надійну систему, яка гарантує конфіденційність і безпеку даних користувачів.

## 4.2. Реалізація клієнтської частини

Для реалізації інтерфейсу користувача в даному дипломному проєкті використано JavaScript та бібліотеку Bootstrap. Це дозволило розробити швидкий та адаптивний вебдодаток. Компоненти інтерфейсу були створені та стилізовані за допомогою Bootstrap. Це забезпечило сучасний вигляд інтерфейсу, який коректно відображається на пристроях з різними розмірами екрану. JavaScript був використаний для додавання динаміки вебдодаткам та покращення їх функціональності.

### 4.2.1. Маршрутизація

Маршрутизація є важливою функціональністю даного вебдодатку, яка дозволяє користувачам взаємодіяти з різними частинами програми, посилаючись на різні URL-адреси. У Django є вбудована система маршрутизації, яка дозволяє визначати, яким чином різні URL-адреси маршрутизуються до відповідних відображень (views) або функцій.

Таблиця 4.1

Маршрутизація у клієнтській частині

Вебсторінка	Шлях	Назва компоненту
Список обладнання	/equipment-list	EquipmentList

Обладнання	/equipment/:id	EquipmentInfo
Логін	/login	Login
Профіль користувача	/profile	Profile
Створення бронювання	/booking	Booking
Редагування бронювання	/booking-edit/:id	BookingEdit
Список бронювання	/booking-list	BookingList
Створення користувача	/user	User
Редагування користувача	/user-edit/:id	UserEdit
Створення запису про ремонт	/equipment-repair	Repair
Редагування запису про ремонт	/equipment-repair-edit/:id	RepairEdit

#### ***4.2.2. Опис графічного інтерфейсу***

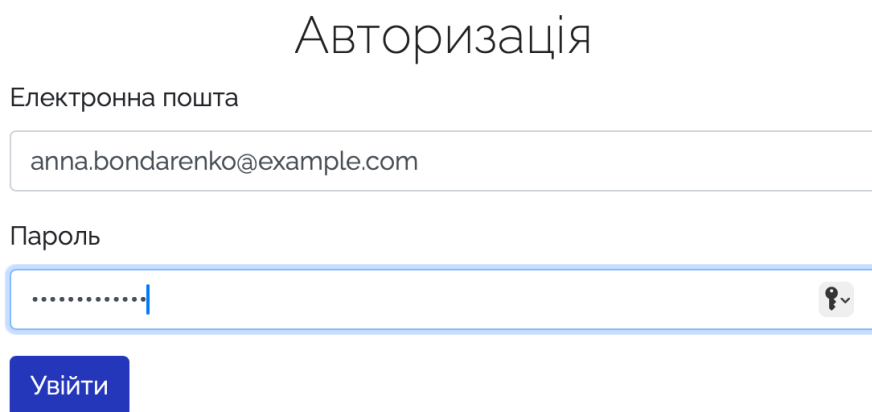
Під час розроблення графічного інтерфейсу для вебдодатку, що призначений для обліку обладнання телеканалу для виїзних зйомок, приділено особливу увагу деталям. Через те, що програмне забезпечення даного типу розробляється індивідуально для кожного підприємства з метою автоматизації виробничих процесів, іноді дизайн вважається менш пріоритетним порівняно з функціональними можливостями. Однак, для забезпечення зручного користування додатком важливо дотримуватися принципів ергономіки та дизайну [24].

Враховуючи унікальність потреб і вимог до розроблюємого програмного забезпечення, особлива увага приділялася забезпеченню

легкості навігації, зрозумілості розміщення елементів, а також зручності користування функціями.

Кольорова палітра та використання візуальних елементів створюють гармонійне візуальне сприйняття для користувачів. Це підкреслює надійність та високий рівень якості додатку. При розробленні дизайну було використано наступні кольори в якості основної палітри: #10165C (темно-синій), #F3B84A (жовтогарячий), #FFFFFF (білий) та #000000 (чорний). Ці кольори були обрані з урахуванням їхньої взаємодії та ефекту на сприйняття користувачів, що забезпечує гармонійний та привабливий зовнішній вигляд вебсайту.

Перша сторінка, на яку переходить користувач, – це сторінка авторизації, де він має ввести свої дані для входу в систему. Це обов’язкова дія, адже без автентифікації користувач не може використовувати основну функціональність додатку. На рис. 4.1 наведена сторінка авторизації.



Авторизація

Електронна пошта

Пароль

Увійти

Рис. 4.1. Сторінка входу в систему

Після введення користувачем логіну (електронної пошти) та пароллю, ця інформація перевіряється на правильність. Якщо ж користувач ввів

некоректні дані, система повідомляє про це користувача. Приклад повідомлення зображено на рис. 4.2.

## Авторизація

Електронна пошта

Пароль

Невірно введений логін або пароль! Повторіть спробу

[Увійти](#)

Рис. 4.2. Приклад повідомлення про помилку під час введення некоректних даних

Одразу після успішної авторизації користувач переходить на сторінку зі списком обладнання, де його можна переглядати, фільтрувати. Сторінка представлена на рис. 4.3.

[Пошук](#)

[Панель адміністратора](#) [Мої бронювання](#) [Вихід](#)

Все обладнання

Камери

Об'єктиви

Штативи

Звукове обладнання

Аксесуари

Тип

- Камери
- Об'єктиви
- Штативи
- Звукове обладнання
- Аксесуари

Назва

- ATEM
- Blackmagic
- Cartoni
- Cooke
- DWR-Ro2D/33
- DaVinci
- Fairlight

Blackmagic URSA Cine 12K

[Детальніше](#)

Blackmagic URSA Broadcast G2

[Детальніше](#)

Art 14mm F1.4 DG DN

Sigma 14mm F1.4 DG DN | Art

[Детальніше](#)

Рис. 4.3. Сторінка зі списком обладнання

June 2024						
Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

## Тип

- Камери
- Об'єктиви
- Штативи
- Звукове обладнання
- Аксесуари

Рис. 4.4. Детальний огляд календаря та фільтрації за типом

Користувач може переглянути детальну інформацію про обладнання перед здійсненням бронювання. Така сторінка містить назву обладнання, фото та технічні характеристики. На цій сторінці користувач може переглянути дати, у які обладнання доступне для бронювання. Даний функціонал реалізовано за допомогою календаря. Сторінку з детальною інформацією про обладнання представлено на рис. 4.5.

[Назад до списку обладнання](#)

[Панель адміністратора](#) [Мої бронювання](#) [Вихід](#)



## Blackmagic URSA Cine 12K

Тип: Камери

Blackmagic URSA Cine 12K — це новітня цифрова кінокамера, яка має великоформатний RGBW-сенсор 36 x 24 мм із крупнішими світлочувливими елементами, котрі забезпечують 16 ступенів динамічного діапазону. Вона також оснащена змінними байонетами, спеціальним допоміжним блоком із 5-дюймовим сенсорним HDR-екраном, вбудованими ND-фільтрами та стандартними роз'ємами Lemo й Fischer. Крім того, камера має величезний знімний накопичувач ємністю 8 ТБ для швидкісної передачі даних через 10G Ethernet або Wi-Fi та вивантаження й синхронізації медіаконтенту із сервісом Blackmagic Cloud.

Рис. 4.5. Сторінка обладнання

Після перегляду детальної інформації обладнання можна забронювати, обравши період, у який обладнання є доступним. На сторінці створення бронювання реалізовано функціональність у вигляді віртуального кошика, аналогічного тому, що зустрічається в Інтернет-магазинах. Тут користувач може додавати до бронювання різне обладнання. У разі конфлікту дат, коли якась позиція обладнання не буде доступна протягом потрібного періоду, система надсилає повідомлення з інформацією про це користувачеві. Таким чином, користувач може швидко та зручно скласти бронювання, отримуючи повідомлення про доступність обладнання у вибраній період.

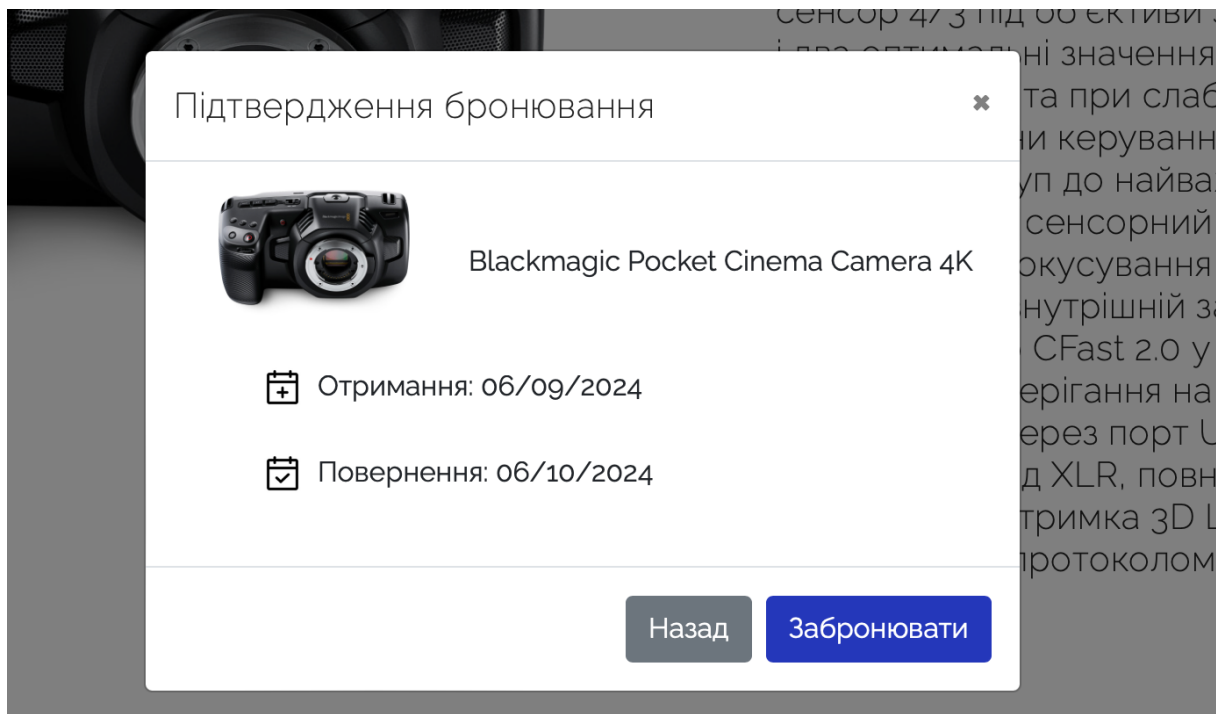


Рис. 4.6. Сторінка створення бронювання

Після успішного створення бронювання, воно автоматично додається до списку всіх зроблених бронювань користувача. В цьому списку доступні опції для управління бронюванням, включаючи можливість видалення або редагування. Таким чином, користувач має повний контроль над своїми бронюваннями і може легко виконувати будь-які необхідні дії з ними,

зручно керуючи своїм розкладом та обладнанням. Приклад такого списку показано на рис. 4.7.

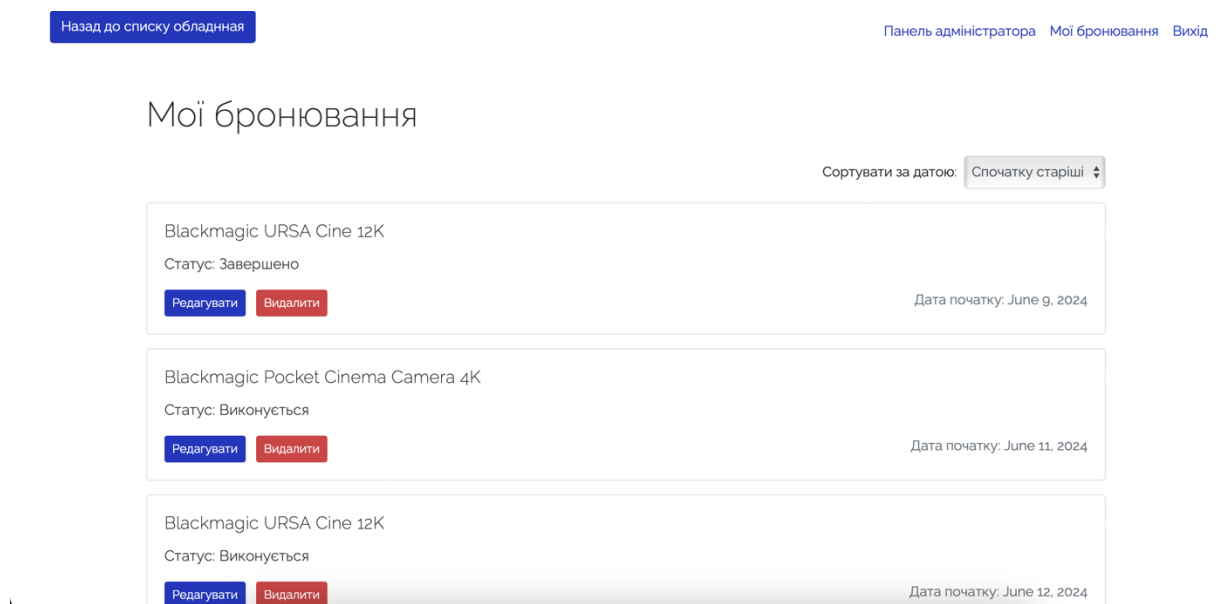


Рис. 4.7. Сторінка з бронюваннями користувача

### 4.3. Тестування вебзастосунку

Проведено смок-тестування розробленого програмного забезпечення для перевірки коректної роботи основних функцій. Тест-кейси наведено в таблиці 4.2.

Таблиця 4.2

Тест-кейси для смок-тестування вебзастосунку

Код тест-кейсу	Дії	Очікуваний результат
T-1	1) Відкрити браузер. 2) Ввести посилання на вебзастосунок.	Завантаження сторінки авторизації.
T-2	1) Відкрити сторінку авторизації вебзастосунку. 2) Ввести невірний логін. 3) Ввести невірний пароль. 4) Натиснути кнопку "Вхід".	Відображення повідомлення про неправильно введені дані.

Т-3	<ol style="list-style-type: none"> <li>1) Увійти до системи з будь-якою роллю.</li> <li>2) Перейти на сторінку з переліком обладнання.</li> <li>3) Вибрати фільтр "Відеокамери".</li> </ol>	Відображення списку обладнання, яке є відеокамерами.
Т-4	<ol style="list-style-type: none"> <li>1) Відкрити сторінку з переліком обладнання.</li> <li>2) Вибрати картку будь-якого обладнання.</li> <li>3) Натиснути на назву або опис обладнання.</li> </ol>	Відображення сторінки з детальною інформацією про обране обладнання
Т-5	<ol style="list-style-type: none"> <li>1) Відкрити сторінку з переліком обладнання.</li> <li>2) У календарі вибрати певну дату.</li> </ol>	Відображення обладнання, яке доступне в обрану дату
Т-6	<ol style="list-style-type: none"> <li>1) Відкрити сторінку авторизації вебзастосунку.</li> <li>2) Ввести логін та пароль користувача з роллю "Технічний асистент".</li> <li>3) Натиснути кнопку "Вхід".</li> </ol>	Користувач має можливість управляти сутностями і налаштовувати доступні дані у панелі технічного асистента.
Т-7	<ol style="list-style-type: none"> <li>1) Відкрити сторінку авторизації вебзастосунку.</li> <li>2) Ввести логін та пароль користувача з роллю "Адміністратор".</li> <li>3) Натиснути кнопку "Вхід".</li> </ol>	Користувач має можливість управляти сутностями і налаштовувати доступні дані у панелі адміністратора.
Т-8	<ol style="list-style-type: none"> <li>1) Обрати обладнання зі списку доступного обладнання.</li> <li>2) Обрати час бронювання.</li> <li>3) Натиснути кнопку "Забронювати".</li> </ol>	<ol style="list-style-type: none"> <li>1) Відображення бронювання у списку бронювань користувача.</li> <li>2) Обладнання стає недоступним у обраний період.</li> </ol>

Т-9	<ol style="list-style-type: none"> <li>1) Обрати обладнання зі списку обладнання.</li> <li>2) Ввести опис несправності.</li> <li>3) Натиснути кнопку "Створити запис".</li> </ol>	<ol style="list-style-type: none"> <li>1) Відображення запису про ремонт у списку про ремонт обладнання.</li> <li>2) Обладнання стає недоступним у обраний період.</li> </ol>
Т-10	<ol style="list-style-type: none"> <li>1) Обрати бронювання, яке хоче відредагувати.</li> <li>2) Натиснути кнопку "Редагувати".</li> <li>3) Змінити час бронювання або видалити обладнання з бронювання.</li> <li>4) Натиснути кнопку "Зберегти".</li> </ol>	<ol style="list-style-type: none"> <li>1) Бронювання оновлюється згідно з внесеними змінами.</li> <li>2) Обладнання стає доступним або недоступним для бронювання, залежно від змін.</li> </ol>
Т-11	<ol style="list-style-type: none"> <li>1) Відкрити сторінку профілю користувача.</li> <li>2) Натиснути кнопку «Вихід».</li> </ol>	<ol style="list-style-type: none"> <li>1) Користувач виходить з системи.</li> <li>2) З'являється сторінка входу.</li> </ol>

Результати проведення smoke-тестування показали, що основні функції вебзастосунку для обліку обладнання телеканалу для виїзних зйомок працюють коректно. Під час тестування вдалося підтвердити, що система здатна обробляти запити користувачів, здійснювати бронювання обладнання та надавати інформацію про його наявність без збоїв. Виявлені незначні помилки були оперативно виправлені, що дозволило підвищити загальну стабільність та надійність додатку. Таким чином димове тестування підтвердило готовність системи до подальшого використання та більш глибоких тестувань.

#### **4.4. Рекомендації щодо подальшого вдосконалення**

Під час створення застосунку з обліку обладнання телеканалу для виїзних зйомок було реалізовано багато корисних функцій. Однак, для

подальшого покращення функціональності, надійності та користувацького досвіду рекомендується звернути увагу на наступні аспекти:

- 1) додати можливість додавати обладнання в «Обране»;
- 2) додати можливість формування PDF-документу з вмістом бронювання, опису запису про ремонт обладнання для ведення звітності;
- 3) додати функцію надсилання сповіщення про готовність, зібраного технічним асистентом, бронювання;
- 4) додати функцію надсилання нагадування за добу про бронювання;
- 5) додати можливість додавати інших користувачів до бронювання, щоб воно відображалось у списках бронювань декількох користувачів;
- 6) додати інтеграцію з Google Calendar для зручного перегляду періоду бронювання та уникання конфлікту запланованих подій.

#### **4.5. Висновки до розділу 4**

У четвертій частині детально описано процес розроблення програмного забезпечення вебзастосунку з фокусом на реалізації серверної та клієнтської частин. Особлива увага була приділена аналізу використаних фреймворків та бібліотек для серверної частини, які визначили основні можливості та ефективність. Також розглядалися різні аспекти, включаючи обробку запитів, зберігання даних та забезпечення безпеки даних.

Подробиці процесу розробки клієнтської частини вебдодатку були також детально описані. Висвітлено використані фреймворки для створення користувацького інтерфейсу та його функціональності. Розглянуто маршрутизацію, яка визначає, які компоненти мають відображатися на певних сторінках. Також надано опис графічного інтерфейсу розроблених вебсторінок.

План тестування розробленого програмного забезпечення ретельно написаний і реалізований, що дозволило перевірити основну

функціональність і стабільність. Для цього було використано димове тестування, яке дозволило оцінити роботу програмного забезпечення в реальних умовах та виявити потенційні проблеми.

У цьому розділі також надано рекомендації щодо подальшого вдосконалення вебзастосунку, оптимізацію продуктивності та покращення безпеки. Ці рекомендації допоможуть підвищити якість та зручність користування вебдодатком, враховуючи потенційні потреби користувачів у майбутньому.

## ВИСНОВКИ

У рамках даного дипломного проєкту розроблено вебзастосунок для обліку обладнання телеканалу для виїзних зйомок, що має на меті підвищити ефективність управління технічними ресурсами під час виїзних зйомок, знизити ризик конфліктів інтересів та оптимізувати робочі процеси.

Виявлено ряд проблем, що виникають серед потенційних користувачів додатку, проаналізовано існуючі програмні рішення, визначено їхні переваги та недоліки, а також сформульовано вимоги до нової системи. З урахуванням результатів порівняльного аналізу визначено, що жоден з існуючих сервісів не вирішує всі проблеми користувачів та не відповідає їхнім потребам повною мірою, тому розроблення вебдодатку для обліку обладнання телеканалу для виїзних зйомок є актуальною.

За результатами аналізу інструментів для розроблення програмного забезпечення обрано мову програмування Python з фреймворком Django та СКБД PostgreSQL. Вебдодаток реалізовано за принципами клієнт-серверної архітектури з використанням патерну проєктування Model -Template-View. У рамках дипломного проєкту реалізовано користувацький інтерфейс, алгоритм перевірки доступності обладнання на певний період часу та спроектовано базу даних. Тестування розробленого програмного забезпечення проведено згідно з процедурою димового тестування.

Використання вебзастосунку для обліку обладнання телеканалу для виїзних зйомок допоможе забезпечити ефективне керування та контроль над усіма аспектами стану обладнання та його використанням. Користувачі матимуть зручний доступ до інформації про наявні ресурси, можливість швидко здійснювати бронювання та керувати власними запитами. Завдяки цьому, процес планування та координації виїзних зйомок стане більш організованим та продуктивним. Також, цей додаток допоможе зменшити можливі ризики пов'язані з недостатнім управлінням обладнанням та

забезпечить збереження важливої інформації про ресурси та їх використання для майбутніх використань.

## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

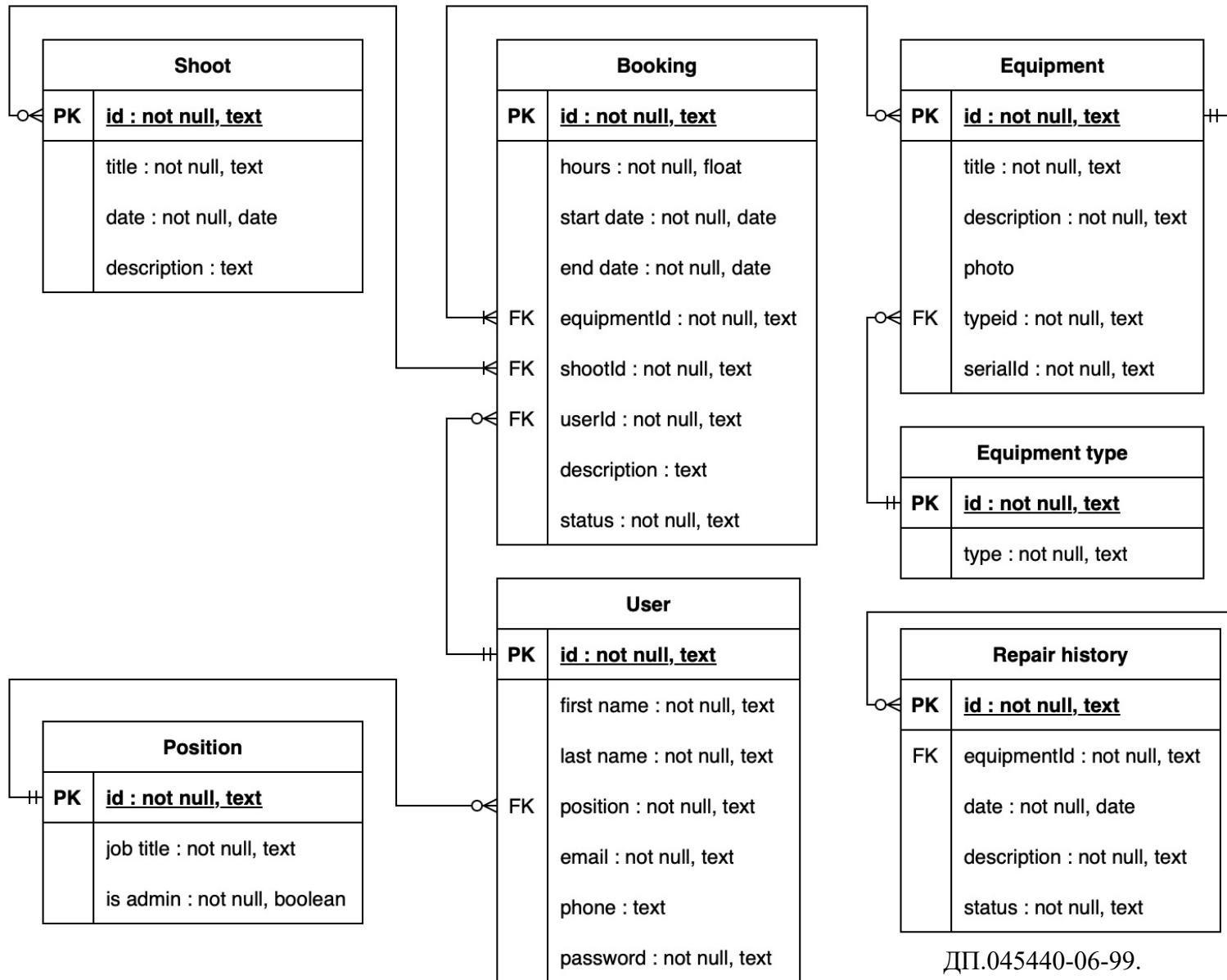
1. Frame Rental [Електронний ресурс]. — Режим доступу: <https://rental.com.ua>
2. Google Calendar [Електронний ресурс]. — Режим доступу: <https://calendar.google.com/calendar/u/0/r?pli=1>
3. Microsoft Excel [Електронний ресурс]. — Режим доступу: <https://www.microsoft.com/uk-ua/microsoft-365/excel>
4. What is XML? [Електронний ресурс]. — Режим доступу: [https://aws.amazon.com/what-is/xml/?nc1=h\\_ls](https://aws.amazon.com/what-is/xml/?nc1=h_ls)
5. Що таке Python? [Електронний ресурс]. — Режим доступу: <https://acode.com.ua/intro-python/>
6. Безпека веб-додатків: найкращі практики та вразливості [Електронний ресурс]. — Режим доступу: <https://itproger.com/ua/news/bezopasnost-veb-prilozheniy-luchshie-praktiki-i-uyazvimosti>
7. JavaScript [Електронний ресурс]. — Режим доступу: <https://astwellsoft.com/blog/tehnology/javascript.html>
8. JavaScript's Dominance in 2024 [Електронний ресурс]. — Режим доступу: <https://www.sencha.com/blog/how-javascript-has-become-the-most-utilitarian-programming-language/>
9. Огляд фреймворків JavaScript. Що, для чого і коли використовувати [Електронний ресурс]. — Режим доступу: <https://dou.ua/forums/topic/34739/>
10. The Best CSS Frameworks to Use in Your Projects for 2024 and Beyond [Електронний ресурс]. — Режим доступу: <https://www.freecodecamp.org/news/best-css-frameworks-for-frontend-devs/>
11. Модульна архітектура для розширеного React-застосунку [Електронний ресурс]. — Режим доступу: <https://www.gen.tech/post/modules-for-react-app>

12. Vue.js — навіщо він нам? [Електронний ресурс]. — Режим доступу: <https://medium.com/@jstify.community/vue-js-навіщо-він-нам-981b5e17af39>
13. Angular vs Vue.js: що обрати ? [Електронний ресурс]. — Режим доступу: <https://dou.ua/lenta/columns/angular-vs-vuejs/>
14. What is Javascript (JS)? [Електронний ресурс]. — Режим доступу: [https://aws.amazon.com/what-is/javascript/?nc1=h\\_ls](https://aws.amazon.com/what-is/javascript/?nc1=h_ls)
15. Top CSS Frameworks for Web Development [Електронний ресурс]. — Режим доступу: <https://www.browserstack.com/guide/top-css-frameworks>
16. Реляційна база даних [Електронний ресурс]. — Режим доступу: <https://ua5.org/database/189-reljaccjjna-baza-danikh.html>
17. What is NoSQL? [Електронний ресурс]. — Режим доступу: <https://www.mongodb.com/resources/basics/databases/nosql-explained>
18. What is PostgreSQL? [Електронний ресурс]. — Режим доступу: <https://www.postgresql.org/about/>
19. What is MySQL and what is it used for? [Електронний ресурс]. — Режим доступу: <https://planetScale.com/learn/articles/what-is-mysql>
20. Django — Documentation URL dispatcher [Електронний ресурс]. — Режим доступу: <https://docs.djangoproject.com/en/5.0/topics/http/urls/>
21. Django — Documentation Making queries [Електронний ресурс]. — Режим доступу: <https://docs.djangoproject.com/en/5.0/topics/db/queries/>
22. Django.fun — Documentation Templates [Електронний ресурс]. — Режим доступу: <https://django.fun/docs/django/5.0/topics/templates/>
23. Хабара Д. О. Методи та засоби пошуку даних в різноманітних веб-ресурсах для спеціалізованого телеграм-канала : магістерська дисертація / Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського». Київ, 2023. 86 с.

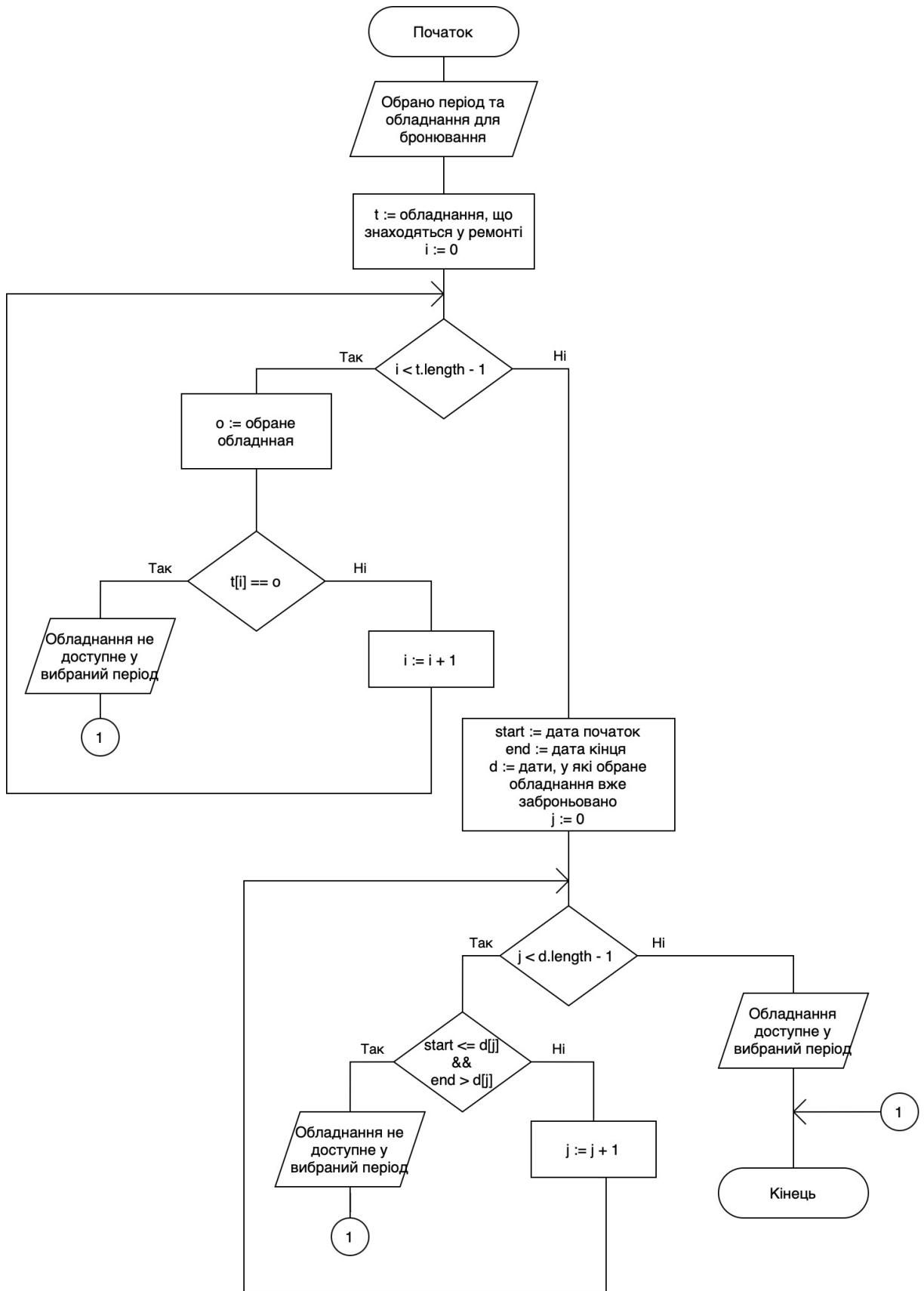
24. Website Ergonomics: How to Make Your Web Resource User-Friendly [Електронний ресурс]. — Режим доступу: <https://coi.ua/en/blog/DesignCo/website-ergonomics-how-to-make-your-web-resource-user-friendly/>

## **ДОДАТКИ**

**Додаток 1**  
**Копії графічних матеріалів**

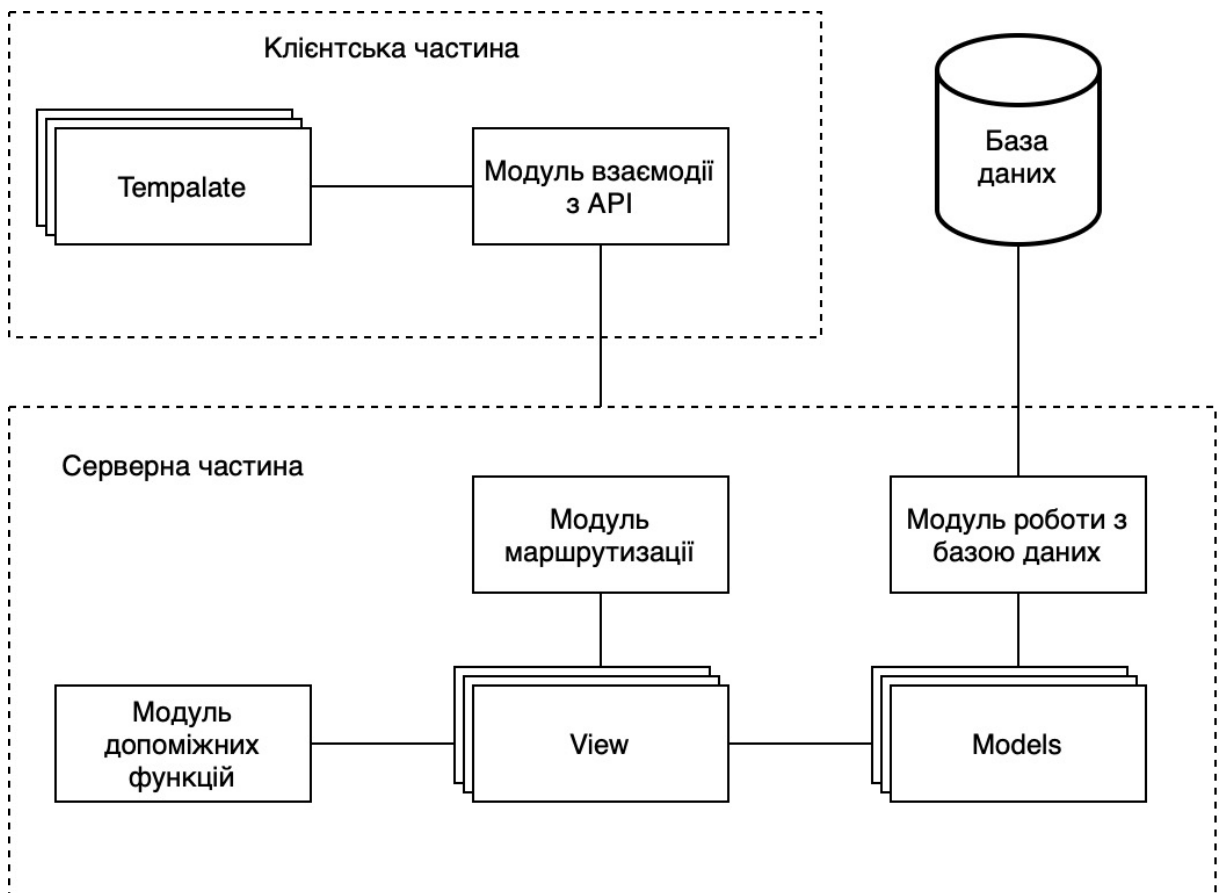


ДП.045440-06-99.  
 Вебзастосунок для обліку обладнання  
 для виїзних зйомок. Структура бази  
 даних. ERD діаграма

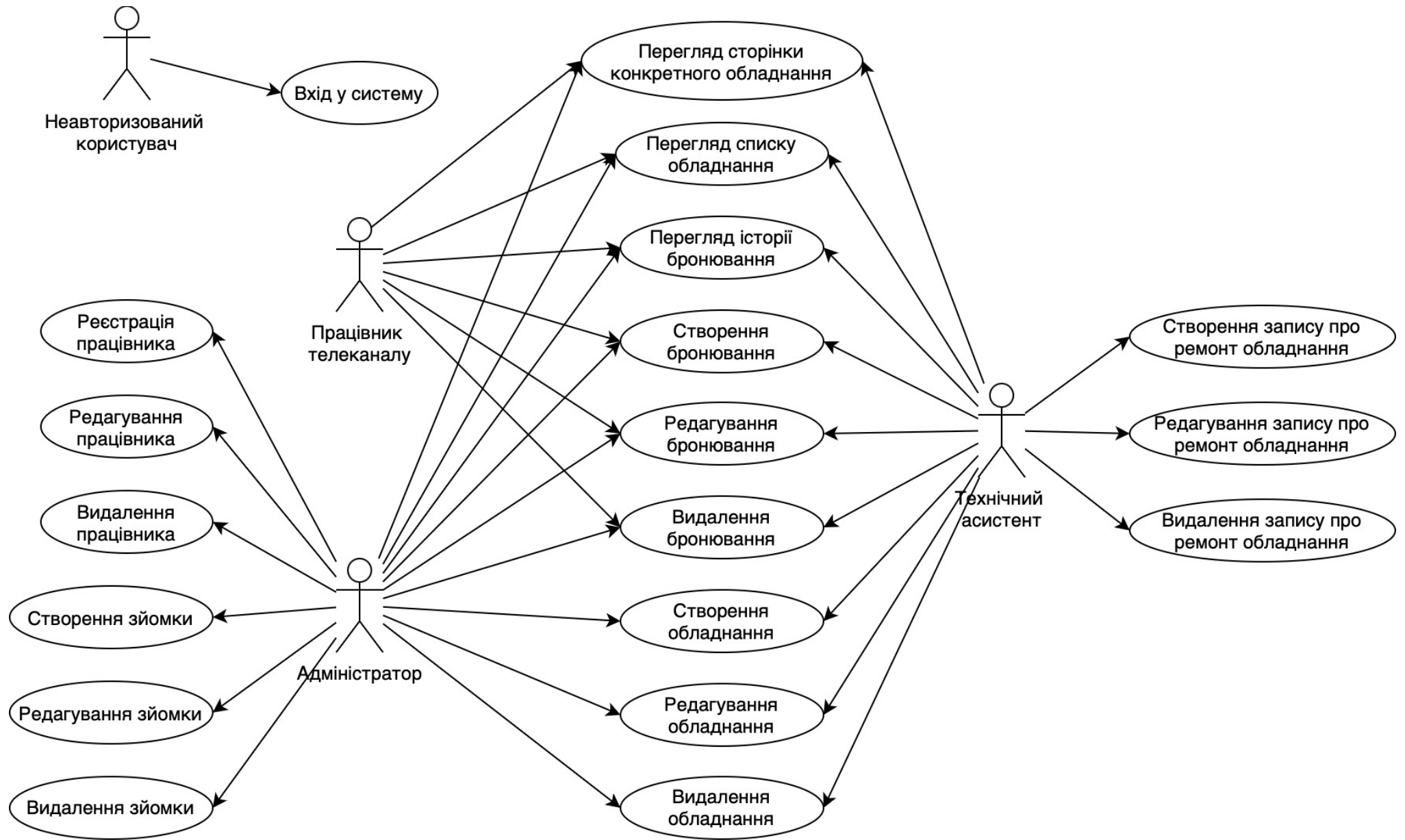


ДП.045440-07-99.

Вебзастосунок для обліку обладнання для виїзних зйомок. зйомок. Алгоритм перевірки доступності обладнання на певний період часу. Блок-схема



Загальна архітектура застосунку  
Гудзіцька Д.С., група КП-02



UML-діаграма варіантів використання  
Гудзіцька Д.С., група КП-02

**Додаток 2**  
**Лістинг програми**

```

from django.urls import path, include

from django.views.generic.base import RedirectView

from django.contrib.auth.views import LoginView

urlpatterns = [

    path('', include('user.urls')),

    path('', include('equipment.urls')),

    path('login/', LoginView.as_view(template_name='user/login.html',
redirect_authenticated_user=True), name='login'),

    path('', RedirectView.as_view(url='login/', permanent=False)),

]

from django.db import models

from django.contrib.auth.models import AbstractBaseUser, BaseUserManager,
PermissionsMixin

class Position(models.Model):

    job_title = models.TextField()

    is_admin = models.BooleanField(default=False)

class CustomUserManager(BaseUserManager):

    def create_user(self, email, first_name, last_name, password=None,
**extra_fields):

        if not email:

            raise ValueError('The Email field must be set')

        email = self.normalize_email(email)

        user = self.model(email=email, first_name=first_name, last_name=last_name,
**extra_fields)

        user.set_password(password)

        user.save(using=self._db)

        return user

    def create_superuser(self, email, first_name, last_name, password=None,
**extra_fields):

        extra_fields.setdefault('is_staff', True)

        extra_fields.setdefault('is_superuser', True)

```

```

        if extra_fields.get('is_staff') is not True:
            raise ValueError('Superuser must have is_staff=True.')

        if extra_fields.get('is_superuser') is not True:
            raise ValueError('Superuser must have is_superuser=True.')

    return self.create_user(email, first_name, last_name, password,
**extra_fields)

class Users(AbstractBaseUser, PermissionsMixin):

    first_name = models.TextField()

    last_name = models.TextField()

    position = models.ForeignKey(Position, on_delete=models.CASCADE, null=True,
blank=True)

    email = models.EmailField(unique=True)

    phone = models.TextField(blank=True, null=True)

    password = models.TextField()

    is_active = models.BooleanField(default=True)

    is_staff = models.BooleanField(default=False)

    objects = CustomUserManager()

    USERNAME_FIELD = 'email'

    REQUIRED_FIELDS = ['first_name', 'last_name']

    def __str__(self):
        return self.email

#добавить hash

    def save(self, *args, **kwargs):

        if not self.id:

            self.password = self.password

            super().save(*args, **kwargs)

from django.contrib.auth.decorators import login_required

from django.shortcuts import render, redirect

```

```
from django.contrib.auth import login, logout

from .authentication import AuthBackend

from django.shortcuts import render

from .models import Users

@login_required

def user_list(request):

    users = Users.objects.all()

    return render(request, 'user/user_list.html', {'users': users})

@login_required

def redirect_to_equipment(request):

    return redirect('equipment:equipment_list')

def login_view(request):

    if request.method == 'POST':

        email = request.POST.get('email')

        password = request.POST.get('password')

        user = AuthBackend.authenticate(request, email=email, password=password)

        if user is not None:

            login(request, user)

            return redirect('equipment:equipment_list')

        else:

            return render(request, 'user/login.html', {"error": 'Невірно введений логін або пароль! Повторіть спробу'})

    else:

        return render(request, 'user/login.html')

@login_required

def logout_view(request):

    logout(request)

    return redirect('login')

from django.contrib.auth.models import User
```

```
from django.db import models

from django.conf import settings

from django.db.models import Q

class EquipmentType(models.Model):

    type = models.TextField()

    icons = models.TextField(default='')

class Equipment(models.Model):

    title = models.TextField()

    description = models.TextField()

    photo = models.TextField(blank=True, null=True)

    type = models.ForeignKey(EquipmentType, on_delete=models.CASCADE)

    serial_id = models.TextField()

class Repair(models.Model):

    STATUS_CHOICES = [

        ('in_progress', 'Виконується'),

        ('completed', 'Завершено'),

    ]

    equipment = models.ForeignKey(Equipment, on_delete=models.CASCADE)

    date = models.DateField()

    description = models.TextField(null=True)

    status = models.CharField(max_length=20, choices=STATUS_CHOICES,
default='in_progress')

class Booking(models.Model):

    STATUS_CHOICES = [

        ('created', 'Створено'),

        ('in_progress', 'Виконується'),
```

```

        ('completed', 'Завершено'),
    ]

    hours = models.TextField()

    start_date = models.DateField()

    end_date = models.DateField()

    equipment = models.ManyToManyField(Equipment)

    user = models.ForeignKey(settings.AUTH_USER_MODEL, on_delete=models.CASCADE)

    status = models.CharField(max_length=20, choices=STATUS_CHOICES,
                              default='created')

from django.contrib.auth.decorators import login_required

from django.shortcuts import render, get_object_or_404

from django.views.decorators.csrf import csrf_exempt

from django.views.decorators.http import require_POST

from django.views.generic import DetailView

from .models import *

from django.http import JsonResponse

from .models import Booking, Repair

from datetime import datetime

from django.db.models import Q

@login_required

def equipment_list(request):

    equipments = Equipment.objects.all()

    equipments_type = EquipmentType.objects.all()

    eq_t = [equipment.title.split()[0] for equipment in equipments]

    unique_words = sorted(set(eq_t))

    context = {

```

```

        'equipments': equipments,

        'equipments_type': equipments_type,

        'unique_words': unique_words,

    }

    return render(request, 'equipment/equipment-list.html', context)

class EquipmentDetailView(DetailView):

    model = Equipment

    template_name = 'equipment/equipment_detail.html'

    context_object_name = 'equipment'

    def check_booking_availability(request):

        if request.META.get('HTTP_X_REQUESTED_WITH') == 'XMLHttpRequest':

            start_date_str = request.POST.get('start_date')

            end_date_str = request.POST.get('end_date')

            equipment_id = request.POST.get('equipment_id')

            try:

                # Преобразуем строки дат в объекты datetime

                start_date = datetime.strptime(start_date_str, '%m/%d/%Y')

                end_date = datetime.strptime(end_date_str, '%m/%d/%Y')

                print("Start date:", start_date)

                print("End date:", end_date)

                overlapping_repairs = Repair.objects.filter(

                    equipment__id__in=equipment_id,

                    date__range=[start_date, end_date],

                    status='in_progress'

                ).exists()

                print(equipment_id)

                existing_bookings = Booking.objects.filter(

                    Q(start_date__lte=end_date) & Q(end_date__gte=start_date),

```

```

        equipment__id=equipment_id
    ).exists()

    if overlapping_repairs or existing_bookings:
        available = False
    else:
        available = True

    return JsonResponse({'available': available})
except ValueError:
    return JsonResponse({'error': 'Invalid date format'}, status=400)
return JsonResponse({'error': 'Invalid request'}, status=400)

```

```
@csrf_exempt
```

```
@login_required
```

```
def submit_booking(request):
```

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

```
        equipment_id = request.POST.getlist('equipment_id')
```

```
        start_date = request.POST.get('start_date')
```

```
        end_date = request.POST.get('end_date')
```

```
        user = request.user
```

```
        booking = Booking.objects.create(
```

```
            hours=24.0,
```

```
            start_date=start_date,
```

```
            end_date=end_date,
```

```
            user=user,
```

```
            status='created'
```

```
        )
```

```
        equipment = Equipment.objects.filter(id__in=equipment_id)
```

```

        booking.equipment.set(equipment)

        return JsonResponse({'success': True})

        return JsonResponse({'success': False, 'error': 'Invalid request'},
            status=400)

$(document).ready(function () {

    $('#datepicker').datepicker({

        format: 'mm-dd-yyyy',

        autoclose: true,

        todayHighlight: true,

    });

    var bookingStartDate = "";

    var bookingEndDate = "";

    $('#checkAvailability').click(function () {

        var equipmentId = $(this).data('id');

        bookingStartDate = $('#startDate').val();

        bookingEndDate = $('#endDate').val();

        $.ajax({

            url: '/check_booking_availability/',

            method: 'POST',

            data: {

                equipment_id: equipmentId,

                start_date: moment(bookingStartDate, 'MM-DD-YYYY').format('MM/DD/YYYY'),

                end_date: moment(bookingEndDate, 'MM-DD-YYYY').format('MM/DD/YYYY'),

                csrfmiddlewaretoken:

                $('#input[name=csrfmiddlewaretoken]').val()

            },

```

```
        success: function (response) {  
            if (response.available) {  
                $('#availabilityMessage').text('Бронювання  
доступно').addClass('alert-success').removeClass('alert-danger');  
            } else {  
                $('#availabilityMessage').text('Бронювання недоступно.  
Оберіть іншу дату.').addClass('alert-danger').removeClass('alert-success');  
            }  
        },  
        error: function (xhr, status, error) {  
            alert('Виникла помилка, повторите спробу!');  
        }  
    });  
});
```

```
$('#.book-equipment').click(function () {  
    var equipmentId = $(this).data('id');  
    var equipmentName = $(this).data('title');  
    bookingStartDate = $('#startDate').val();  
    bookingEndDate = $('#endDate').val();  
  
    $('#equipmentId').val(equipmentId);  
    $('#equipmentName').text(equipmentName);  
    $('#startDateModal').text(bookingStartDate);  
    $('#endDateModal').text(bookingEndDate);  
});
```

```
$('#submitBooking').click(function () {  
    $.ajax({
```

```
url: '/submit_booking/',

method: 'POST',

data: {

    equipment_id: $('#equipmentId').val(),

    start_date: moment(bookingStartDate, 'MM-DD-
YYYY').format('YYYY-MM-DD'),

    end_date: moment(bookingEndDate, 'MM-DD-YYYY').format('YYYY-
MM-DD'),

    csrfmiddlewaretoken:
$('#input[name=csrfmiddlewaretoken]').val()

},

success: function (response) {

    if (response.success) {

        alert('Бронирование успешно создано!');

        $('#bookingModal').modal('hide');

    } else {

        alert('Помилка');

    }

},

error: function (xhr, status, error) {

    alert('Произошла ошибка при создании бронирования. Повторите
попытку.');
```

**Додаток 3**  
**Копія презентації**

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО”



ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ

**ВЕБЗАСТОСУНОК ДЛЯ БРОНЮВАННЯ  
ОБЛАДНАННЯ ДЛЯ ВИЇЗНИХ ЗЙОМОК**

Виконала: Гудзіцька Дарина Сергіївна

Керівник: Доц. кафедри ПЗКС, к.т.н., доц., Заболотня Тетяна Миколаївна

Київ – 2024

1/16

# АКТУАЛЬНІСТЬ



- У сьогоденнішніх реаліях телеканали мають велику кількість запитів на виїзні сюжети, тому часто виникають проблеми щодо наявності обладнання на потрібний час.
- Наразі не існує єдиного автоматизованого рішення для обліку обладнання підприємства.

# ПОСТАНОВКА ЗАДАЧІ



**Мета проєкту:** створення програмного забезпечення для ефективного використання обладнання телеканалу та уникнення конфліктів у бронюванні обладнання.

## **Завдання:**

1. Здійснити аналіз ринку та виявити проблеми, що виникають при обліку технічного обладнання.
2. Обрати засоби реалізації програмного забезпечення.
3. Визначити вимоги до розроблювального застосунку.
4. Розробити та протестувати вебдодаток для бронювання обладнання для виїзних зйомок.
5. Визначити перспективи розвитку проєкту.

# ІСНУЮЧІ АНАЛОГИ



Frame Rental



Google Calendar



Microsoft Excel

# ОБРАНІ ЗАСОБИ РЕАЛІЗАЦІЇ



Серверна частина

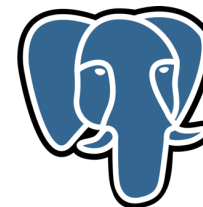


Мова програмування  
Python



Фреймворк Django

СУБД



Система управління  
базами даних

Клієнтська частина



Мова програмування



Інструментарій для  
створення інтерфейсів

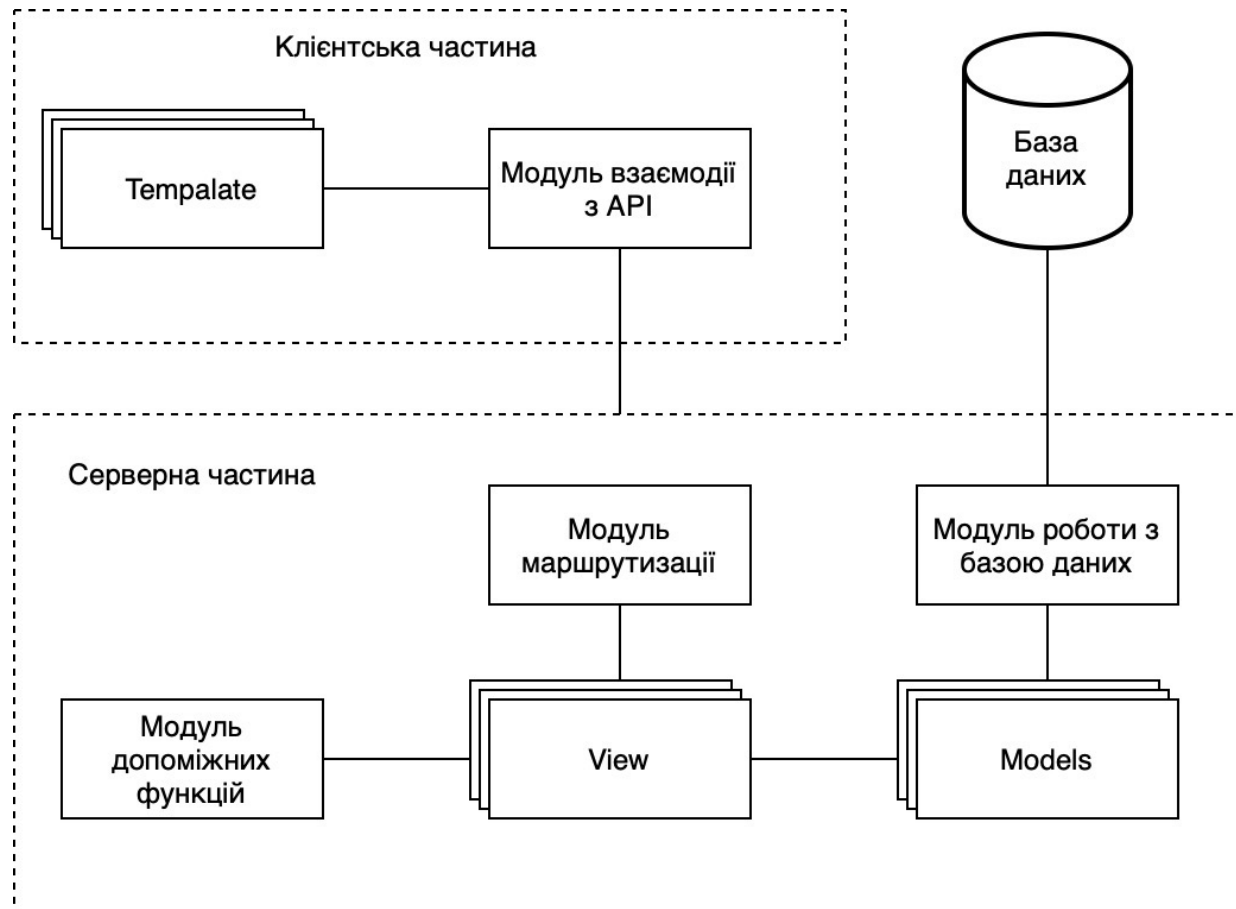
# ФУНКЦІОНАЛЬНІ ВИМОГИ ДО РОЗРОБЛЮВАНОВОГО ПЗ



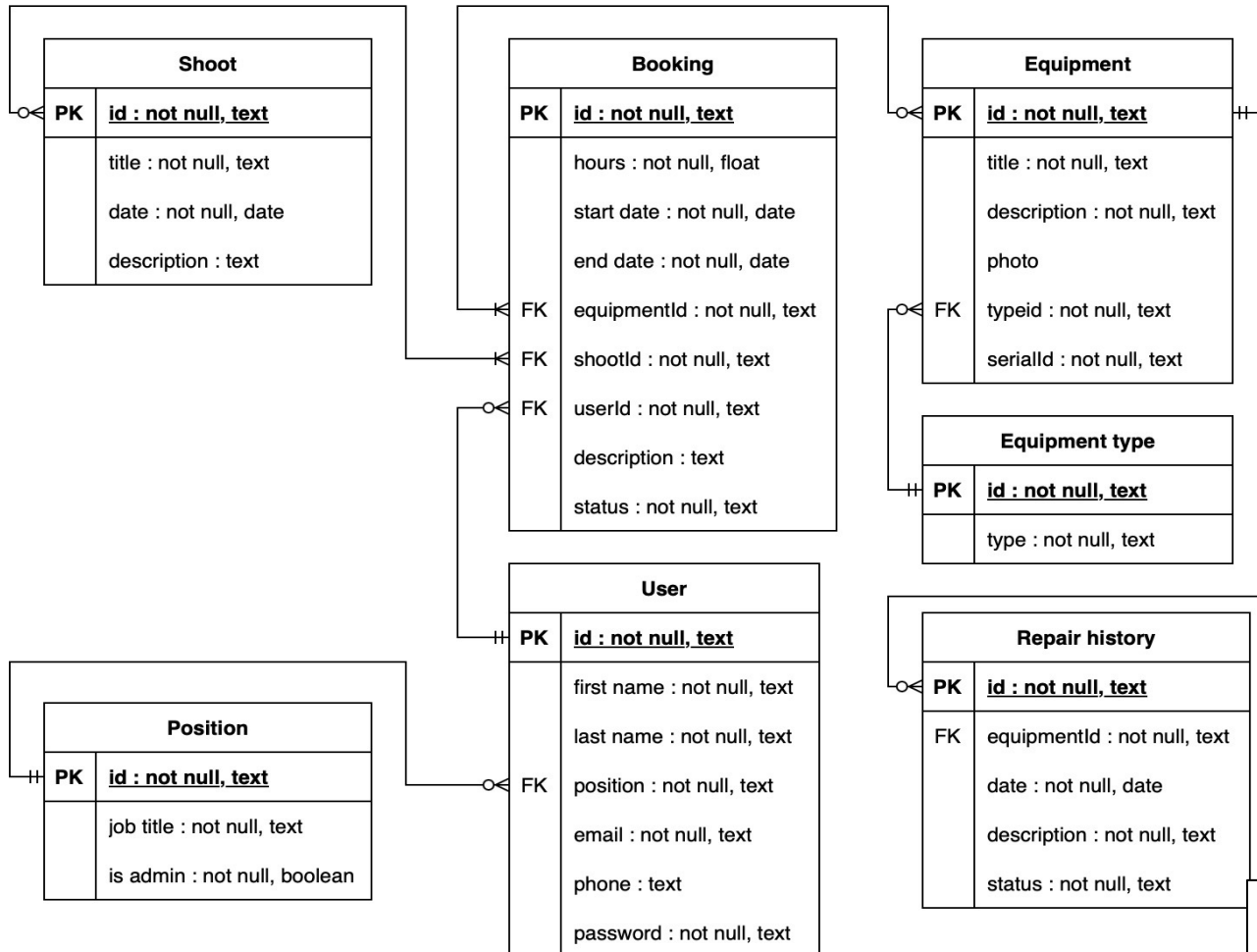
Розроблюваний вебзастосунок для бронювання обладнання для виїзних зйомок повинен надавати можливість:

- аутентифікуватись у застосунку;
- переглядати інформацію про наявне обладнання;
- створювати, редагувати та видаляти записи про бронювання;
- створювати профілі нових користувачів;
- надсилати повідомлення про реєстрацію у застосунку на електронну пошту;
- створювати записи про нове обладнання;
- створювати та редагувати дані про стан обладнання.

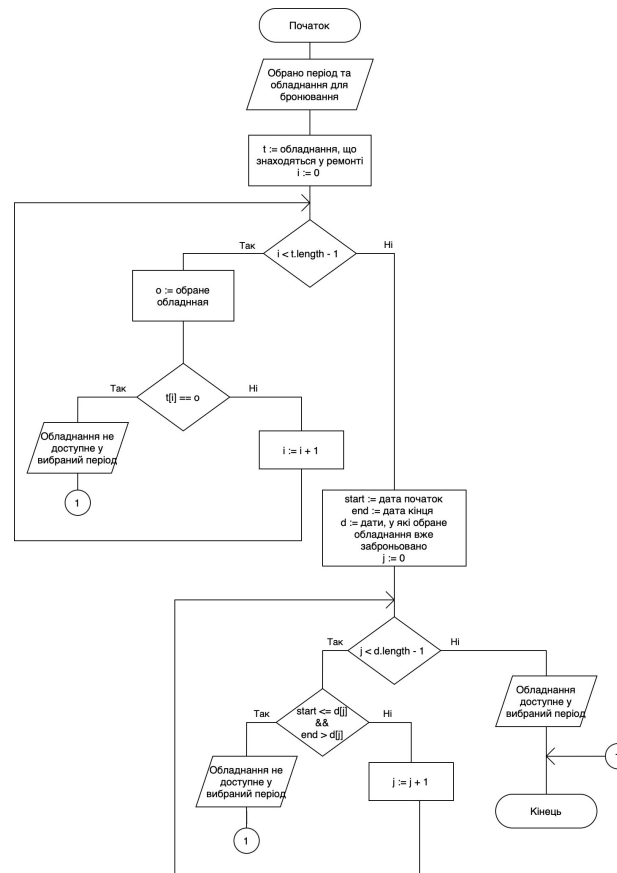
# АРХІТЕКТУРА РОЗРОБЛЕНОЇ СИСТЕМИ



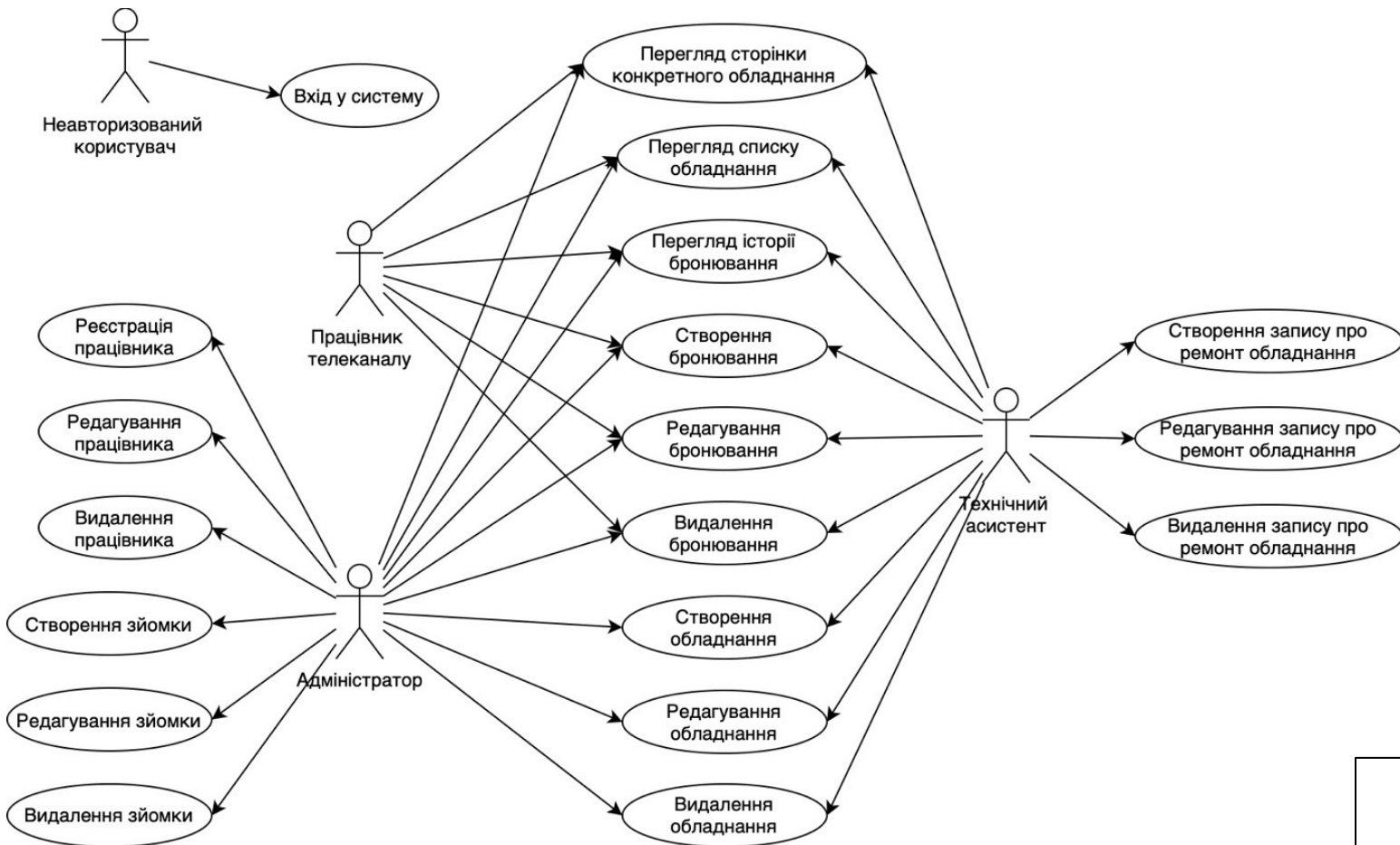
# СТРУКТУРА БАЗИ ДАНИХ



# АЛГОРИТМ ПЕРЕВІРКИ ДОСТУПНОСТІ ОБЛАДНАННЯ НА ПЕВНИЙ ПЕРІОД ЧАСУ



# ДІАГРАМА ВИКОРИСТАННЯ



# ІНТЕРФЕЙС ГОЛОВНОЇ СТОРІНКИ




Пошук

Пошук


[Панель адміністратора](#) [Мої бронювання](#) [Вихід](#)

Все обладнання

 Камери

 Об'єктиви

 Штативи

 Звукове  
обладнання

 Аксесуари

Тип

- Камери
- Об'єктиви
- Штативи
- Звукове обладнання
- Аксесуари

Назва

- ATEM
- Blackmagic
- Cartoni
- Cooke
- DWR-R02D/33
- DaVinci
- Fairlight



Blackmagic URSA Cine  
12K

Детальніше



Blackmagic URSA  
Broadcast G2

Детальніше



 Art  
14mm F1.4 DG DN

Sigma 14mm F1.4 DG DN |  
Art

Детальніше

# ІНТЕРФЕЙС СТОРІНКИ З БРОНЮВАННЯМИ КОРИСТУВАЧА



[Назад до списку обладнання](#)

[Панель адміністратора](#) [Мої бронювання](#) [Вихід](#)

## Мої бронювання

Сортувати за датою: Спочатку старіші ▾

Blackmagic URSA Cine 12K

Статус: Завершено

[Редагувати](#)

[Видалити](#)

Дата початку: June 9, 2024

Blackmagic Pocket Cinema Camera 4K

Статус: Виконується

[Редагувати](#)

[Видалити](#)

Дата початку: June 11, 2024

Blackmagic URSA Cine 12K

Статус: Виконується

[Редагувати](#)

[Видалити](#)

Дата початку: June 12, 2024

12/16

# ПЕРСПЕКТИВИ РОЗВИТКУ ПРОДУКТУ



- Можливість формування PDF-документу з вмістом бронювання, опису запису про ремонт обладнання для ведення звітності.
- Можливість надсилання сповіщення про готовність бронювання, зібраного технічним асистентом.
- Інтеграція з Google Calendar для зручного перегляду періоду бронювання та уникання конфлікту запланованих подій.



# АВТОРСЬКЕ СВДОЦТВО НА РОЗРОБКУ

Було сформовано і подано пакет документів на отримання авторського свідчення на розроблене програмне забезпечення.

# ВИСНОВКИ



- Проведено аналіз існуючого програмного забезпечення для бронювання обладнання для виїзних зйомок.
- Обрано технічний стек для розроблення вебдодатка.
- Визначено та описано вимоги до вебдодатка.
- Розроблено вебдодаток для бронювання обладнання для виїзних зйомок.
- Розглянуто архітектуру системи.
- Наведено інформацію про базу даних та збережені у ній таблиці.
- Описано алгоритм перевірки доступності обладнання на певний період.
- Виконане тестування вебдодатка, у результаті якого виправлені помилки відображення інтерфейсу.
- Визначено перспективи розвитку проєкту.



**ДЯКУЮ ЗА УВАГУ**

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

**“ЗАТВЕРДЖЕНО”**

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

“ \_\_\_ ” \_\_\_\_\_ 2023 р.

**ВЕБЗАСТОСУНОК ДЛЯ ОБЛІКУ ОБЛАДНАННЯ ТЕЛЕКАНАЛУ  
ДЛЯ ВИЇЗНИХ ЗЙОМОК**

**Програма та методика тестування**

ДП.045440-04-51

**“ПОГОДЖЕНО”**

Керівник проєкту:

\_\_\_\_\_ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Дарина ГУДЗІЦЬКА

## ЗМІСТ

1. Об'єкт випробувань.....	3
2. Мета тестування.....	3
3. Методи тестування.....	3
4. Засоби та порядок тестування.....	4

## **1. ОБ'ЄКТ ВИПРОБУВАНЬ**

Вебдодаток для обліку обладнання телеканалу для виїзних зйомок вебсайт, створений на платформі Django з використанням технології JavaScript та Bootstrap.

## **2. МЕТА ТЕСТУВАННЯ**

У процесі тестування має бути перевірено наступне:

- 1) функціональна працездатність елементів сторінок вебзастосунку;
- 2) наявність доступу до бази дистанційних з бронюваннями;
- 3) можливість отримання повідомлень на електронну пошту;
- 4) забезпечення належного рівня безпеки даних;
- 5) зручність роботи з вебсайтом;
- 6) відповідність дизайну вимогам Технічного завдання.

## **3. МЕТОДИ ТЕСТУВАННЯ**

Тестування виконується методом Gray Box Testing. Перевіряється як код, так і безпосередньо програмний продукт на відповідність функціональним вимогам. Тестування відбувається на рівні «системного тестування».

Використовуються наступні методи:

- 1) функціональне тестування, зокрема на рівні Critical path test (базове тестування);
- 2) тестування продуктивності програмного забезпечення, зокрема Stability testing (тестування стабільності) та Load testing (навантажувальне тестування);
- 3) тестування інтерфейсу.

#### 4. ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Працездатність вебзастосунку перевіряється шляхом:

- 1) динамічного ручного тестування – введенням граничних та недопустимих значень в поля, які можна редагувати;
- 2) динамічного ручного тестування на відповідність функціональним вимогам;
- 3) статичного тестування коду;
- 4) тестування вебзастосунку в різних веббраузерах;
- 5) тестування при максимальному навантаженні;
- 6) тестування стабільності роботи при різних умовах;
- 7) тестування зручності використання;
- 8) тестування інтерфейсу.

**Факультет прикладної математики**  
**Кафедра програмного забезпечення комп'ютерних систем**

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

\_\_\_\_\_ Євгенія СУЛЕМА

“ \_\_\_ ” \_\_\_\_\_ 2024 р.

**ВЕБЗАСТОСУНОК ДЛЯ ОБЛІКУ ОБЛАДНАННЯ ТЕЛЕКАНАЛУ  
ДЛЯ ВИЇЗНИХ ЗЙОМОК**  
**Керівництво користувача**

ДП.045440-05-34

“ПОГОДЖЕНО”

Керівник проєкту:

\_\_\_\_\_ Тетяна ЗАБОЛОТНЯ

Нормоконтроль:

\_\_\_\_\_ Микола ОНАЙ

Виконавець:

\_\_\_\_\_ Дарина ГУДЗІЦЬКА

## ЗМІСТ

1. Опис структури вебзастосунку .....	3
2. Процедура авторизації користувача .....	4
3. Сторінка зі списком наявного обладнання .....	5
4. Сторінка для створення бронювання.....	7
5. Сторінка для управління записами про ремонт.....	9
6. Сторінка для управління записами про ремонт.....	10
7. Сторінка для управління записами про користувачів.....	11
8. Сторінка для управління записами про користувачів.....	12

## **1. Опис структури вебзастосунку**

Вебзастосунок для бронювання обладнання для виїзних зйомок реалізовано у вигляді односторінкового застосунку. Він містить у собі наступні складові:

- 1) сторінку реєстрації;
- 2) головну сторінку зі списком наявного обладнання;
- 3) сторінка з детальною інформацією про обладнання;
- 4) сторінку для створення бронювання;
- 5) сторінку для управління записами про ремонт;
- 6) сторінку для управління обладнанням;
- 7) сторінку для управління користувачами;
- 8) сторінку для управління зйомками.

Останні 2 сторінок, згаданих у переліку існуючих, відносяться до адміністративної панелі вебзастосунку, доступної лише адміністратору системи. Сторінки з пунктів 7 та 8 доступні технічному асистенту та адміністратору.

## 2. Процедура авторизації користувача

При відвідуванні вебзастосунку для бронювання обладнання для виїзних зйомок користувач потрапляє на сторінку для авторизації. Для подальшої взаємодії з функціоналом системи, користувач обов'язково має пройти процедуру авторизації.

Авторизація

Електронна пошта

Пароль


   

Рис. 1. Форма для реєстрації користувача

### 3. Сторінка зі списком наявного обладнання

Оразу після успішної авторизації користувач переходить на головну сторінку. Основний елемент сторінки – список обладнання наявного у телеканалу. Список представлений у вигляді сітки карток з фото та назвою обладнання. Також на картці присутня кнопка «Забронювати», для швидкого оформлення бронювання без перегляду детальної інформації на сторінці з описом обладнання.

Згори на сторінці присутній футер, де розташовані кнопка «Мої бронювання» та іконка профілю користувача. Натиснувши на кнопку «Мої бронювання», користувач перейде до списку з його бронюваннями, де він може відредагувати та видалити існуючі записи.

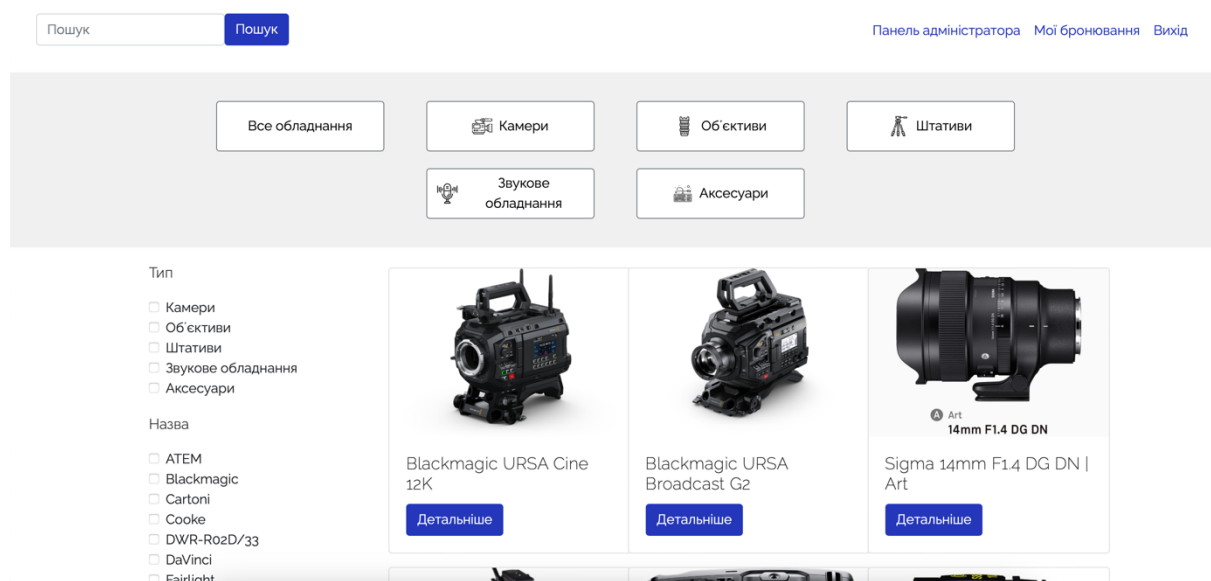


Рис. 2. Сторінка зі списком обладнання

Щоб обрати конкретний тип обладнання, користувач може скористатися фільтрацією. Щоб обрати обладнання, яке доступне на у конкретну дату та/або період, користувач може скористатися календарем.

June 2024						
Su	Mo	Tu	We	Th	Fr	Sa
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30						

Тип

- Камери
- Об'єктиви
- Штативи
- Звукове обладнання
- Аксесуари

Рис. 3. Детальний огляд календаря та фільтрації за типом

#### 4. Сторінка для створення бронювання

На сторінці створення бронювання реалізовано функціональність у вигляді віртуального кошика, аналогічного тому, що зустрічається в Інтернет-магазинах. Тут користувач може додавати до бронювання різне обладнання або забронювати лише 1. Після успішного бронювання на екрані з'являється підтвердження цього.

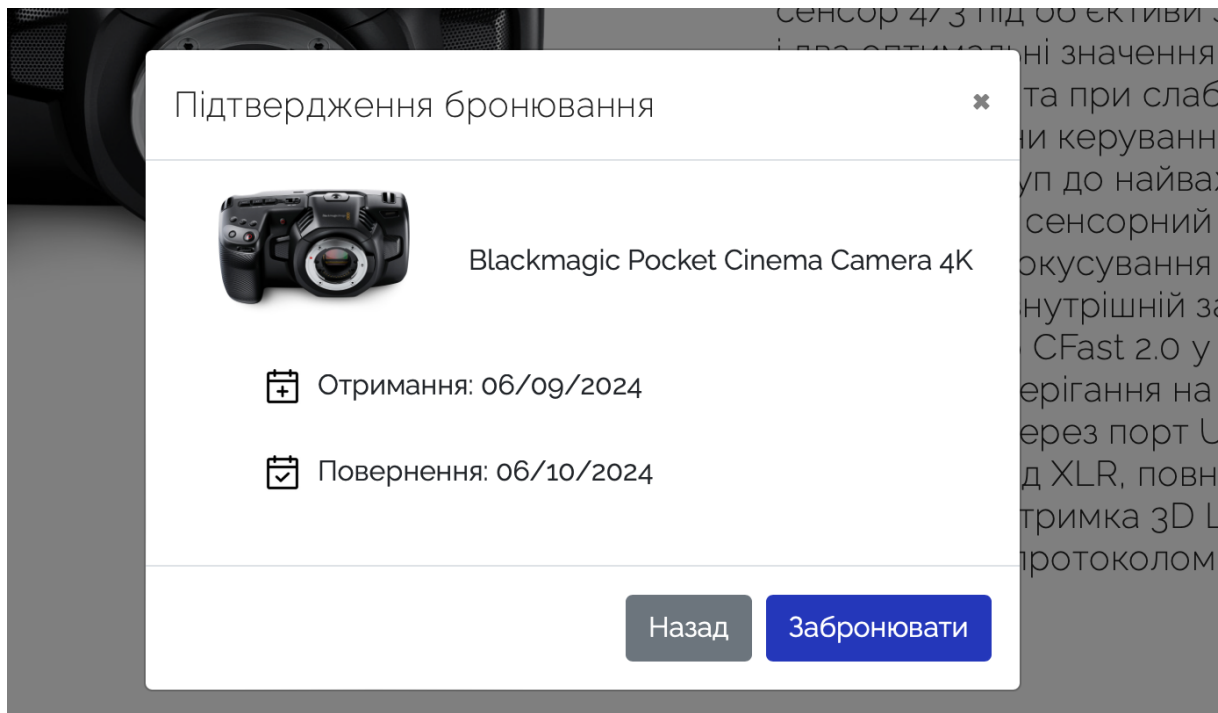


Рис. 4. Підтвердження створення бронювання

Далі користувач переходить на сторінку зі списком його бронювань. Тут він може відсортувати їх за датою створення. А також відредагувати ще невидане бронювання. Або ж видалити бронювання не залежно від його статусу.

## Мої бронювання

Сортувати за датою: Спочатку старіші ▾

Blackmagic URSA Cine 12K Статус: Завершено <a href="#">Редагувати</a> <a href="#">Видалити</a> Дата початку: June 9, 2024
Blackmagic Pocket Cinema Camera 4K Статус: Виконується <a href="#">Редагувати</a> <a href="#">Видалити</a> Дата початку: June 11, 2024
Blackmagic URSA Cine 12K Статус: Виконується <a href="#">Редагувати</a> <a href="#">Видалити</a> Дата початку: June 12, 2024

Рис. 5. Сторінка зі списком бронювань користувача

## 5. Сторінка для управління записами про ремонт

Сторінка управління записами про ремонт доступна лише користувачам з ролями «Технічний асистент» та «Адміністратор». На ній представлена панель адміністратора для управління всіма сутностями програми. Користувач може переглядати усі записи про ремонт та створювати нові. Для створення нового запису, користувач має заповнити відповідну форму, у якій вказати назву обладнання та описати причину несправності. Запис збережеться зі статусом «Виконується», надалі його можна змінити на сторінці зі списком записів про ремонт.

The screenshot shows the user interface for an administrator. At the top left, the user is identified as 'Адміністратор'. At the top right, there is a link for 'Мої бронювання' with a user icon. A sidebar on the left contains a list of navigation items: 'Обладнання', 'Записи про бронювання', 'Записи про обладнання', 'Записи про зйомку', 'Записи про користувачів', and 'Записи про ремонт'. Under 'Записи про ремонт', there are two sub-items: 'Новий запис' (highlighted in blue) and 'Список всіх записів'. The main content area is titled 'Додати новий запис про ремонт'. It features a dropdown menu for 'Обладнання', a text input field for 'Description', and two buttons at the bottom: 'Зберегти' (Save) and 'Скасувати' (Cancel).

Рис. 5. Форма реєстрації нового запису про ремонт

## 6. Сторінка для управління записами про ремонт

Сторінка управління записами про обладнання також доступна лише користувачам з ролями «Технічний асистент» та «Адміністратор». Для створення нового запису, користувач має заповнити відповідну форму, у якій вказати назву нового обладнання, обрати його тип, серійний номер та детально описати обладнання. Запис збережеться та почне відображатись у списку обладнання у панелі адміністратора та на головній сторінці додатку.

The screenshot shows the administrator interface for adding new equipment. On the left is a sidebar menu with the user role 'Адміністратор' and a 'Мої бронювання' link. The main content area is titled 'Додати нове обладнання'. It contains four input fields: 'Назва', 'Тип обладнання', 'Description', and 'Серійний номер'. At the bottom are two buttons: 'Зберегти' (Save) and 'Скасувати' (Cancel).

Адміністратор Мої бронювання

Обладнання

Записи про бронювання

Записи про обладнання

Новий запис

Список всіх записів

Записи про зйомку

Записи про користувачів

Записи про ремонт

### Додати нове обладнання

Назва

Тип обладнання

Description

Серійний номер

Рис. 6. Форма реєстрації нового обладнання

## 7. Сторінка для управління записами про користувачів

Сторінка управління записами про обладнання доступна лише користувачам з роллю «Адміністратор». Для створення нового запису, користувач має заповнити відповідну форму, у якій вказати персональні дані нового користувача та його посаду на підприємстві. Запис збережеться та новому користувачу надійде пароль на вказану пошту.

The screenshot shows a web interface for an administrator. On the left is a sidebar menu with the following items: 'Обладнання', 'Записи про бронювання', 'Записи про обладнання', 'Записи про користувачів' (with sub-items 'Новий запис' and 'Список всіх записів'), 'Записи про ремонт', and 'Записи про зйомки'. The 'Новий запис' item is highlighted in blue. At the top right of the main area, it says 'Мої бронювання' with a user icon. The main heading is 'Додати нове обладнання'. The form contains the following fields: 'Прізвище' and 'Ім'я' (two input boxes), 'Посада' (one input box), 'Електронна пошта' (one input box), and 'Номер телефону' (one input box). At the bottom of the form are two buttons: 'Зберегти' (Save) and 'Скасувати' (Cancel).

Рис. 7. Форма реєстрації нового користувача

## 8. Сторінка для управління записами про користувачів

Сторінка управління записами про обладнання також доступна лише користувачам з роллю «Адміністратор». Для створення нового запису, користувач має заповнити відповідну форму, у якій вказати назву, дату та опціонально опис.

The screenshot shows a web interface for an administrator. On the left is a vertical sidebar with a user profile icon and the name 'Адміністратор'. Below the name are several menu items with dropdown arrows: 'Обладнання', 'Записи про бронювання', 'Записи про обладнання', 'Записи про користувачів', 'Записи про ремонт', and 'Записи про зйомки'. The 'Новий запис' (New record) item is highlighted in blue, and 'Список всіх записів' (List of all records) is below it. In the top right corner, there is a link 'Мої бронювання' (My reservations) with a user icon. The main content area is titled 'Додати нове обладнання' (Add new equipment). It contains three input fields: 'Назва' (Name), 'Дата' (Date), and 'Опис' (Description). At the bottom of the form are two buttons: 'Зберегти' (Save) in blue and 'Скасувати' (Cancel) in white.

Рис. 8. Форма реєстрації нової зйомки