

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

**Факультет інформатики та обчислювальної техніки**

**Кафедра інформаційних систем та технологій**

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо– професійною програмою «Інтегровані інформаційні  
системи»**

**спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Вебзастосунок для оптимізації організаційного процесу в офісі»**

Виконав (– ла):

студент (– ка) IV курсу, групи ІА– 93

Візняк Максим Богданович \_\_\_\_\_

Керівник:

Старший викладач,

Проскура Світлана Леонідівна \_\_\_\_\_

Рецензент:

доцент кафедри ІІІ, к.т.н., доцент

Лісовиченко Олег Іванович \_\_\_\_\_

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

Студент (– ка) \_\_\_\_\_

Київ – 2023 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**  
**Факультет інформатики та обчислювальної техніки**  
**Кафедра інформаційних систем та технологій**

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо– професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Олександр РОЛІК

«\_\_» \_\_\_\_\_ 20

\_\_\_\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

Візняку Максиму Богдановичу

1. Тема проєкту «Вебзастосунок для оптимізації організаційного процесу в офісі», керівник проєкту Проскура Світлана Леонідівна, старший викладач кафедри ІСТ, затверджені наказом по університету від «31» травня 2023 р. № 2101– с
2. Термін подання студентом проєкту: 12 червня 2023 року
3. Вихідні дані до проєкту: клієнтський інтерфейс
4. Зміст пояснювальної записки: перелік скорочень, умовних позначень, термінів, вступ, аналіз предметної області, обґрунтування вибору засобів розробки вебзастосунку, опис структури вебзастосунку, власна реалізація, висновки, перелік використаних джерел.
5. Перелік графічного матеріалу: діаграма використання, схема бізнес процесів, діаграма діяльності, діаграма послідовності.
6. Дата видачі завдання 2 травня 2023 року

## Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Огляд та аналіз існуючих рішень та технологій	21.04.2023	Виконано
2	Вибір архітектури	25.04.2023	Виконано
3	Створення діаграм та схем	30.04.2023	Виконано
4	Створення бази даних	15.05.2023	Виконано
5	Створення клієнтської частини	25.05.2023	Виконано
6	Створення серверної частини	30.05.2023	Виконано
7	Тестування вебзастосунку	02.06.2023	Виконано

Студент

Максим ВІЗНЯК

Керівник

Світлана ПРОСКУРА

## АНОТАЦІЯ

Візняк М.Б. Вебзастосунок для оптимізації організаційного процесу в офісі. КІП ім. Ігоря Сікорського, Київ, 2023.

Проект містить 60 с. тексту, 14 рисунків, 12 таблиць, покликання на 29 літературні джерела та 4 конструкторські документи.

Ключові слова: вебзастосунок, ІТ, офіс, робочий процес.

Об'єктом розробки є створення вебзастосунку для оптимізації організаційних процесів в офісі.

Мета розробки – створення вебзастосунку, що допоможе працівникам зменшити витрати власного часу на організацію робочого місця.

У результаті цього проекту було проаналізовано існуючі рішення та розроблено зручну та гнучкий інструмент, що дозволяє користувачам підняти ефективність власної роботи, скоротивши витрати власного часу на такі організаційні процеси, як резервування робочих місць та бронювання додаткового обладнання. Також було розроблено зручний конструктор для створення власних планів офісу.

Даний проект має потенціал стати універсальним інструментом, що дозволить охопити більшу область, де застосунок може бути корисним. У майбутньому можна створити додатковий функціонал, який дозволить інтегруватись з сервісами та спростити процес роботи людей у великих групах.

## SUMMARY

Vizniak M. B. Web application for optimization of the organizational process in the office. Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, 2023.

The project contains 60 pages of text, 14 figures, 12 tables, references to 29 literary sources and 4 design documents.

Keywords: web application, IT, office, workflow.

The object of development is to create a web application for optimizing organizational processes in the office.

The purpose of the development is to create a web application that will help employees reduce their time spent on organizing their workplace.

As a result of this project, we analyzed existing solutions and developed a convenient and flexible tool that allows users to increase their work efficiency by reducing their time spent on organizational processes such as workplace reservations and booking additional equipment. A convenient designer was also developed to create custom office plans.

This project has the potential to become a universal tool that will cover a wider area where the app can be useful. In the future, additional functionality can be created that will allow integration with services and simplify the process of people working in large groups.

Номер рядка	Формат	Позначення	Найменування	Кільк. аркушів	Номер екзем.	Примітка
1			<u>Документація загальна</u>			
2						
3			Знову розроблена			
4						
5	A4	IA93.070БАК.004 ПЗ	Вебзастосунок	60		
6			для оптимізації			
7			організаційного процесу в			
8			офісі. Пояснювальна записка			
9						
10	A3	IA93.070БАК.004 Д1	Вебзастосунок	1		
11			для оптимізації			
12			організаційного процесу в			
13			офісі. Діаграма використання			
14						
15	A3	IA93.070БАК.004 Д2	Вебзастосунок	1		
16			для оптимізації			
17			організаційного процесу в			
18			офісі. Схема бізнес процесів			
19						
20	A3	IA93.070БАК.004 Д3	Вебзастосунок	1		
21			для оптимізації			
22			організаційного процесу в			
23			офісі. Діаграма діяльності			
24						
25	A3	IA93.070БАК.004 Д4	Вебзастосунок	1		
26			для оптимізації			
27			організаційного процесу в			
28			офісі. Діаграма послідовності			
<b>IA93.070БАК.004 ТП</b>						
Зм.	Аркуш	№ докум.	Підпис	Дата		
Розроб.		Візник М.Б.			Літ.	Аркуш
Керівн.		Проскура С.Л.			Т	Аркушів
						1
						1
Затв.					КПІ ім. Ігоря Сікорського Група ІА-93	
				Вебзастосунок для оптимізації організаційного процесу в офісі. Відомість проекту		

**Пояснювальна записка**  
**до дипломного проєкту**  
**на тему: «Вебзастосунок для оптимізації**  
**організаційного процесу в офісі»**

Київ – 2023 року

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	4
ВСТУП .....	5
1 АНАЛІЗ ПРЕДМЕНОЇ ОБЛАСТІ .....	7
1.2 Опис процесу діяльності.....	8
1.3 Огляд наявних аналогів.....	9
1.3.1 Robin .....	9
1.3.2 DeskFlex.....	12
1.3.3 Teem.....	14
1.3.4 Condeco.....	16
1.4 Порівняльний аналіз.....	18
1.5 Постановка задачі .....	19
Висновки до розділу 1 .....	20
2 ОГЛЯД ТЕХНОЛОГІЙ РОЗРОБКИ.....	21
2.1 Вибір архітектури вебзастосунку.....	21
2.1.1 Multi–Page Application .....	21
2.1.2 Single–Page Application .....	22
2.1.3 SSR (Server–Side Rendering).....	24
2.1.4 Підсумок вибору архітектури .....	26
2.2 Вибір технологій розробки користувацької частини.....	27
2.3 Вибір технологій розробки серверної частини.....	29
2.4 Вибір Бази даних.....	30
2.5 Вибір середовища розробки .....	31

					<b>IA93.070БАК.004 ПЗ</b>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Візник М.Б.			Вебзастосунок для оптимізації організаційного процесу в офісі. Пояснювальна записка	Літ.	Арк.	Акрюшів
Перевір.		Проскура С. Л.				2	60	
						КПІ ім. Ігоря Сікорського		
Затверд.						Група ІА-93		

Висновки до розділу 2.....	32
<b>3 РОЗРОБКА ТА ПРОЄКТУВАННЯ .....</b>	<b>34</b>
3.1 Моделювання системи.....	34
3.2 Опис сутностей системи.....	35
3.3 Серверна частина.....	38
3.3.1 Архітектура REST API.....	39
3.3.2 API ендпоінти .....	41
3.3.3 Архітектура MVC.....	43
3.4 Клієнтська частина .....	45
3.4.1 JWT .....	46
3.4.2 Redux toolkit .....	48
3.4.3 React–konva .....	49
3.4.4 Структура клієнтської частини.....	51
3.5 Схема бізнес процесів .....	53
3.6 Діаграма діяльності .....	53
3.7 Діаграма послідовності .....	53
Висновки до розділу 3.....	54
<b>4 ОГЛЯД ВЕБЗАСТОСУНКУ .....</b>	<b>56</b>
Висновки до розділу 4.....	57
<b>ВИСНОВКИ.....</b>	<b>58</b>
<b>СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....</b>	<b>59</b>
<b>ДОДАТОК А.....</b>	<b>62</b>

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface  
SPA – Single Page Application  
CSR – Client–side Rendering  
SSR – Server–side Rendering  
SSG – Server–side Generation  
ISR –Incremental Static Regeneration  
MPA Multi Page Application  
MVC – Model–View–Controller  
HTML – HyperText Markup Language  
CSS – Cascading Style Sheets  
HMR – Hot Module Replacement  
NPM – Node Package Manager  
NoSQL – Not Only Structured Query Language  
MQL – MongoDB Query Language  
VS Code – Visual Studio Code  
JWT– JSON Web Tokens  
AWS – Amazon Web Services

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		4

## ВСТУП

Сучасний офісний простір є складним середовищем, де ефективно управління та оптимізація організаційних процесів є необхідними для успіху будь-якої організації. Швидкі технологічні зміни та зростання кількості співробітників створюють потребу в нових підходах до організації робочих місць, розміщенні периферійних пристроїв та управлінні резервуванням робочих місць. Традиційні методи планування офісного простору, які базуються на приблизних оцінках та статистичних даних, часто неефективні, адже не враховують індивідуальних потреб співробітників та навантаження на робочі місця.

У цьому контексті розробка вебзастосунку для оптимізації організаційних процесів в офісі стає надзвичайно актуальною. Метою даного проекту є розробка та імплементація вебзастосунку, який надасть можливість створювати власні плани офісних приміщень, точно розміщувати робочі місця та додаткову периферію. Користувачі зможуть візуалізувати офісний простір, розташовувати робочі місця, обладнання та меблі відповідно до своїх потреб. Додатково, вебзастосунок буде мати функцію резервування робочих місць та необхідного обладнання в режимі реального часу, що забезпечить гнучкість та ефективну координацію використання ресурсів.

Цей вебзастосунок буде допомагати організаціям максимально використовувати свої ресурси, забезпечувати зручні умови для співробітників та покращувати продуктивність роботи. Оптимальне розміщення робочих місць та периферійних пристроїв дозволить зменшити простір обладнання та звільнити місце для більш ефективного використання. Співробітники зможуть легко знаходити вільні робочі місця та резервувати їх в зручний для них спосіб, що зменшить конфлікти та забезпечить раціональне використання ресурсів. Також вебзастосунок надасть можливість спостерігати за навантаженням робочих місць, аналізувати дані та виявляти патерни, що

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		5

дозволить планувати розширення офісного простору та внесення змін у його організацію.

Одним з головних викликів, з якими стикаються організації, є ефективне використання офісного простору. Традиційні підходи до розміщення робочих місць можуть бути неефективними та неспроможними задовольнити зростаючі потреби компаній. Вебзастосунок для оптимізації організаційних процесів в офісі вирішує цю проблему, надаючи можливість створювати власні плани офісних приміщень, точно розміщувати робочі місця та периферійні пристрої, що дозволить забезпечити оптимальне використання доступного простору та покращити робочий процес. Застосування такого вебзастосунку сприятиме ефективному використанню офісного простору, зменшенню затрат на оренду приміщень та забезпеченню гнучкості в розміщенні робочих зон.

Розробка вебзастосунку для оптимізації організаційних процесів в офісі не тільки забезпечить краще використання ресурсів та покращить умови для співробітників, але й сприятиме підвищенню продуктивності офісного колективу. Забезпечення ефективного використання офісного простору та ресурсів впливає на швидкість роботи, забезпечує комфортні умови для кожного працівника та стимулює співпрацю. Завдяки точному розміщенню робочих місць та периферійних пристроїв, співробітники зможуть легше спілкуватися, обмінюватися ідеями та здійснювати спільні проекти. Також можливість бронювання робочих місць та обладнання сприятиме ефективному використанню часу та забезпечить безперервність робочих процесів.

Загалом, розробка вебзастосунку для оптимізації організаційних процесів в офісі відповідає потребам сучасних компаній у ефективному управлінні ресурсами та підвищенні продуктивності. Цей інструмент допоможе організаціям створити оптимальне робоче середовище, забезпечити комфорт та сприятливі умови для співробітників, а також зменшити зайві витрати та підвищити ефективність використання ресурсів.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		6

# 1 АНАЛІЗ ПРЕДМЕНОЇ ОБЛАСТІ

## 1.1 Опис предметного середовища

Існуючі технології та інструменти в офісному управлінні часто не забезпечують ефективну організацію та використання ресурсів, що призводить до зайвих витрат часу, енергії та грошей. Однак, розробка вебзастосунок, спрямованого на оптимізацію організаційних процесів, має безліч переваг.

Перш за все, вебзастосунок для резервування робочих місць дозволить покращити ефективність використання робочого простору в офісі. Кожен працівник зможе зарезервувати робоче місце заздалегідь, уникнувши непотрібних затримок та нестабільності при пошуку вільного простору. Це дозволить зосередитися на роботі та покращить продуктивність всього колективу.

Також такий вебзастосунок сприятиме покращенню комунікації та співпраці між працівниками. Завдяки системі резервування, колеги зможуть планувати спільні зустрічі та спілкування, співпрацювати над проєктами та ділитися ідеями. Це підвищить командний дух, стимулюватиме обмін знаннями та покращить загальну ефективність роботи колективу.

Крім того, вебзастосунок для оптимізації організаційних процесів в офісі дозволить забезпечити зручний доступ до необхідних ресурсів. Проєкти, документи, контакти - все це може бути легко доступним з одного місця. Працівники зможуть швидко знаходити необхідну інформацію та взаємодіяти з колегами без зайвих зусиль.

Враховуючи всі ці переваги, вебзастосунок для оптимізації організаційного процесу в офісі стає необхідним інструментом для сучасних компаній, які прагнуть підвищити ефективність, забезпечити зручні умови роботи для своїх працівників та покращити загальну організацію робочого процесу. Використання вебзастосунку дозволить зменшити зайві витрати часу

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		7

та ресурсів, покращить комунікацію та співпрацю, а також забезпечить зручний доступ до необхідних ресурсів, що призведе до підвищення ефективності та успішності організації.

## 1.2 Опис процесу діяльності

Для того ,щоб користувач міг вільно користуватися функціоналом даного вебзастосунку, йому необхідно авторизуватися у системі. Для авторизації користувач повинен ввести електронну пошту, яка є логіном та пароль. У випадку якщо користувач не має акаунту, то він повинен зареєструватися вказавши власне ім'я, прізвище, електронну пошту та пароль.

Після входу до системи, користувач має можливість переглянути усі доступні йому кімнати, створити власну кімнату, та доєднатися до існуючої за допомогою коду доступу, переглянути власний профіль.

При перегляді кімнати, до якої користувач вже отримав доступ, він може зарезервувати за собою робоче місце та після цього вже забронювати додаткову периферію, якщо місце вже зайняте, користувач може переглянути інформацію про людину, яка зайняла це місце. Після бронювання вільного місця, користувач має можливість скасувати його резервацію.

Якщо користувач є власником кімнати, він має можливість переглянути код доступу, отримати список усіх учасників, які вже мають доступ до даної кімнати, та видаляти користувачів з цього списку.

При створенні кімнати ко, користувач може створити власний план офісу за допомогою зручного конструктора, розставити на ньому робочі місця, вказати перелік додаткової периферії.

Діаграму використання наведено в ІА93.070БАК.004 Д1

					ІА93.070БАК.004 ПЗ	Арк.
						8
Зм.	Лист	№ докум.	Підпис	Дата		

### 1.3 Огляд наявних аналогів

На сьогодні існує декілька різних застосунків, які дуже сильно допомагають у оптимізації організаційних процесів для офісів. Усі вони мають власні переваги та недоліки, але водночас виконують головну функцію – бронювання робочих місць працівників. Ці засоби дозволяють співробітникам легко знаходити вільні робочі місця та резервувати їх на певний час. Вони надають зручні інтерфейси, що дозволяють користувачам швидко переглядати доступні місця та обирати ті, які відповідають їхнім потребам.

Одним з найпоширеніших типом таких застосунків є електронна система бронювання робочих місць. Цей інструмент дозволяє співробітникам легко резервувати вільні місця через онлайн-інтерфейс. Вони можуть переглядати наявність робочих місць, вибрати певну зону або кабінет, а також вказувати тривалість резервації. Ця система дозволяє ефективно використовувати офісний простір, уникати конфліктів між співробітниками та забезпечувати комфортні умови для праці.

Інший популярний тип застосунку – мобільний додаток для бронювання робочих місць. Цей інструмент надає співробітникам можливість бронювати робоче місце за допомогою свого смартфона або планшета. Вони можуть швидко переглядати вільні місця, обирати зручний час та тривалість резервації, а також отримувати сповіщення про підтвердження бронювання.

#### 1.3.1 Robin

Robin [1] – це інноваційний додаток, розроблений для оптимізації організаційних процесів в офісі. Його основна функція – бронювання робочих місць для співробітників. Додаток надає зручну та просту у використанні платформу для резервації робочих місць в режимі реального часу.

					ІА93.070БАК.004 ПЗ	Арк.
						9
Зм.	Лист	№ докум.	Підпис	Дата		

Завдяки додатку Robin, співробітники можуть швидко знайти вільне робоче місце та забронювати його всього кількома клацаннями. Додаток пропонує інтерактивну мапу офісного простору (рисунок 1.1), де співробітники можуть переглядати наявність вільних місць та їх розташування. Вони можуть обирати місця в залежності від їхніх потреб, таких як тиша, комунікація з колегами або доступ до певних пристроїв.

Окрім бронювання робочих місць, Robin дозволяє резервувати периферійні пристрої, такі як принтери, сканери або проектори. Це забезпечує додаткову зручність для співробітників, які можуть планувати свою роботу заздалегідь та мати впевненість у наявності необхідного обладнання.

Додаток Robin також має функції аналітики та статистики, що дозволяє організації отримувати дані про використання офісного простору та ресурсів. Це допомагає виявляти ефективність використання простору, виявляти надлишки або нестачу ресурсів, а також планувати подальші зміни в організації робочих місць.

Robin працює на різних платформах, включаючи веб-версію та мобільні додатки для iOS та Android. Це робить його доступним для користувачів з різних пристроїв та сприяє зручності використання.

Загалом, додаток Robin є потужним інструментом для оптимізації організаційних процесів в офісі, забезпечуючи простий та ефективний спосіб бронювання робочих місць та периферійних пристроїв. Він допомагає організаціям максимально використовувати свої ресурси, покращувати умови для співробітників та підвищувати продуктивність роботи.

					ІА93.070БАК.004 ПЗ	Арк.
						10
Зм.	Лист	№ докум.	Підпис	Дата		

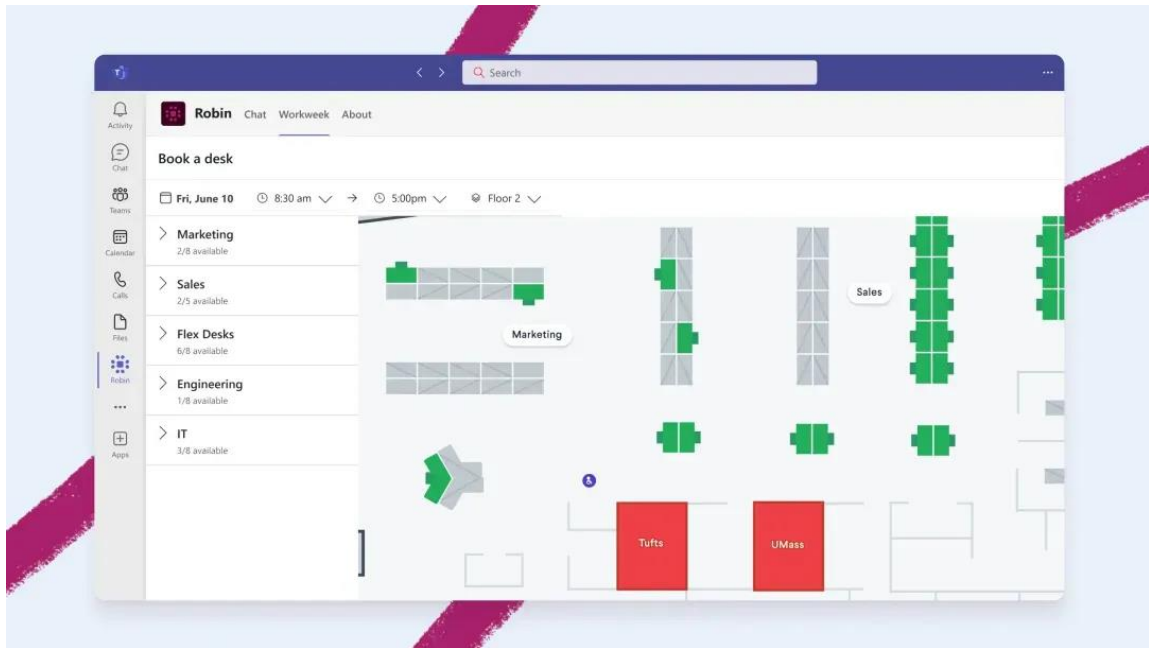


Рисунок 1.1 – Конструктор планів офісу додатку Robin [2]

#### Переваги:

- Robin має інтуїтивний і зручний інтерфейс, що дозволяє просто та швидко робити бронювання робочих місць;
- додаток є гнучким, він дозволяє налаштовувати різні параметри бронювання, такі як тривалість, доступність ресурсів та правила використання;
- застосунок підтримує інтеграцію з популярними календарними додатками, такими як Google Календар, Outlook та інші, що дозволяє працівникам зручно планувати свій час;
- Robin надає детальну аналітичну інформацію про використання робочих місць, що дозволяє організації зрозуміти, як ефективно використовуються їхні ресурси та виявити можливі покращення.
- присутній мобільний додаток, що позитивно впливає на доступність та зручність використання сервісу.

#### Недоліки:

- додаток є платним, це може створити незручності для невеликих компаній;

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		11

– для повного функціонування Robin може вимагати певної інфраструктури, такої як розміщення датчиків або підтримка спеціального обладнання, що може бути вартісним та складним у впровадженні;

– використання нового додатку вимагає часу та зусиль для навчання співробітників та їх прийняття нової системи. Деякі користувачі можуть мати труднощі з адаптацією до нового інструменту.

### 1.3.2 DeskFlex

DeskFlex [3] – це сучасний і високофункціональний додаток для організаційного управління робочим простором в офісі. Цей застосунок дозволяє ефективно розподіляти та бронювати робочі місця для співробітників, забезпечуючи оптимальне використання ресурсів.

Однією з ключових особливостей DeskFlex є його гнучкість та адаптивність до різних потреб організацій. За допомогою цього додатку можна налаштувати різні типи робочих просторів, включаючи відкриті офіси, кабінети або спеціалізовані зони для колективної роботи. Кожен співробітник може легко знайти вільне робоче місце за допомогою інтерактивної (рисунок 1.2) мапи та забронювати його зручним для нього способом.

DeskFlex також надає можливість бронювання зустрічних кімнат та конференц-залів. Це сприяє організації ефективних зустрічей та нарад, де співробітники можуть використовувати необхідне обладнання та зручність приміщень для проведення своїх зустрічей.

Одним з важливих аспектів DeskFlex є його простота в управлінні та інтеграція з іншими системами. Додаток може бути легко налаштований та інтегрований з існуючими організаційними системами, що дозволяє забезпечити безперебійну роботу та ефективне використання ресурсів.

DeskFlex також забезпечує аналітику та звітність про використання робочого простору, що дозволяє організаціям отримувати цінні дані для

					ІА93.070БАК.004 ПЗ	Арк.
						12
Зм.	Лист	№ докум.	Підпис	Дата		

прийняття рішень щодо планування та оптимізації простору. За допомогою цих аналітичних функцій організації можуть знайти ефективні шляхи покращення використання ресурсів та забезпечення комфортних умов для своїх співробітників.

DeskFlex має мобільний застосунок, який доступний для скачування на пристрої з операційною системою iOS та Android. Він дозволяє співробітникам зручно керувати робочим простором та використовувати функціонал DeskFlex на своєму мобільному пристрої.

Завдяки мобільному застосунку DeskFlex, користувачі можуть бронювати робочі місця в офісі, переглядати та змінювати свої резервації, отримувати сповіщення про зміни в розкладі та інші повідомлення відносно організаційних процесів.

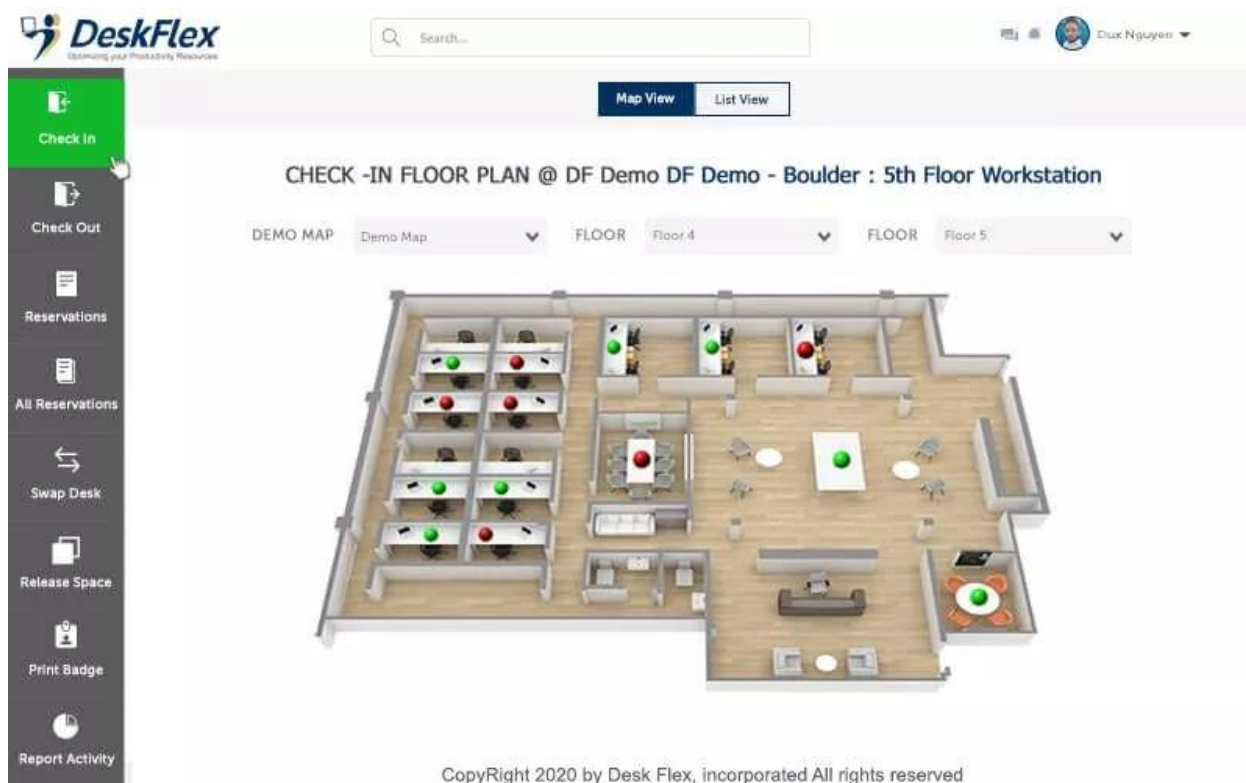


Рисунок 1.2 – Конструктор планів офісу додатку DeskFlex [4]

#### Переваги:

– DeskFlex надає користувачам гнучкі можливості бронювання робочих місць залежно від їх потреб. Він дозволяє налаштувати розташування, тип та зручності робочого місця відповідно до вимог кожного користувача;

– DeskFlex може легко інтегруватись з календарними додатками та системами управління, що дозволяє автоматично синхронізувати бронювання та нагадування, спрощуючи процес планування та організації робочого простору;

– DeskFlex допомагає збільшити продуктивність співробітників шляхом оптимізації робочих умов. Користувачі можуть вибирати місця з відповідним оснащенням та зручностями, що сприяє покращенню комфорту та концентрації на роботі;

– DeskFlex дозволяє організаціям ефективно використовувати ресурси, такі як офісні приміщення та обладнання. Він допомагає уникнути зайвих витрат на невикористані робочі місця та забезпечує оптимальне використання доступних ресурсів.

#### Недоліки:

– впровадження DeskFlex може вимагати певного зусилля та часу для налагодження інтеграції з існуючими системами управління та календарями. Це може бути складним завданням для організацій зі складною інфраструктурою та багатьма відділами.

### 1.3.3 Teem

Teem [5] є сучасним додатком, спрямованим на оптимізацію організаційних процесів у робочому середовищі. Цей застосунок надає широкі можливості для керування робочим простором та покращення продуктивності співробітників.

					ІА93.070БАК.004 ПЗ	Арк.
						14
Зм.	Лист	№ докум.	Підпис	Дата		

Однією з ключових особливостей Teem є його функція бронювання робочих місць та переговорних кімнат (рисунок 1.3). Завдяки цьому, користувачі можуть швидко знайти та зарезервувати необхідні приміщення для зустрічей, конференцій або праці в команді.

Teem також надає зручний календарний інтерфейс, що дозволяє планувати та організовувати зустрічі та події. Користувачі можуть легко створювати розклади роботи, додавати учасників та надсилати запрошення, що сприяє ефективній координації робочих процесів.

Teem забезпечує різні додаткові функції, такі як контроль доступу до приміщень, візуалізація розташування працівників на плані офісу та статистика використання ресурсів. Ці функції дозволяють організаціям краще розуміти та оптимізувати використання робочого простору.

Teem є потужним інструментом для покращення організації робочого середовища і забезпечення ефективності робочих процесів. Він дозволяє компаніям краще використовувати свої ресурси та забезпечує зручні умови для співробітників, сприяючи підвищенню продуктивності та співпраці.

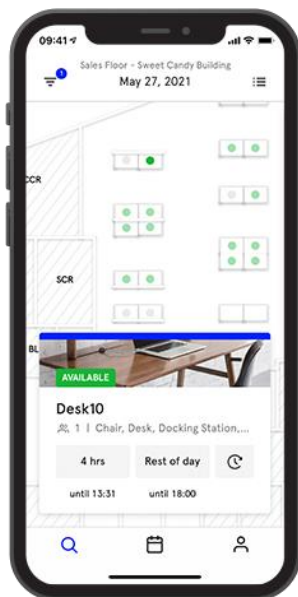


Рисунок 1.3 – Конструктор планів офісу додатку Teem [6]

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		15

#### Переваги:

- Teem надає зручний та інтуїтивно зрозумілий інтерфейс, який дозволяє легко резервувати простори та ресурси;
- застосунок надає функції аналітики та звітності, що дозволяють організаціям отримувати цінні дані про використання простору та ресурсів, що допомагає вдосконалювати розстановку та поліпшувати робоче середовище;
- додаток має мобільну версію, що дозволяє користувачам мати доступ до сервісу у своєму смартфоні;
- сервіс має базові інтеграції з календарями Google Календар та Microsoft Outlook.

#### Недоліки:

- використання Teem є платним, що негативно впливає на малі організації, оскільки вимагає певних витрат на ліцензію та підтримку;
- додаток не є гнучким у питанні інтеграції зі сторонніми сервісами.

#### 1.3.4 Condeco

Condeco [7] – це додаток для керування резервуванням конференц-залів та робочих просторів. Він є досить популярним застосунком для управління робочими місцями та організаційними процесами в офісному середовищі. Він спрощує та автоматизує процеси бронювання робочих місць, забезпечує ефективне використання офісного простору та ресурсів, а також підвищує комунікацію та співпрацю між співробітниками.

Однією з основних переваг додатку Condeco є його інтуїтивно зрозумілий та легкий у використанні інтерфейс. Користувачі можуть швидко та зручно бронювати робочі місця за допомогою веб-порталу (рисунок 1.4) або мобільного додатку, вибираючи зручний час та місце для роботи. Бронювання може бути зроблено заздалегідь або в режимі реального часу, що дає більшу гнучкість користувачам.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		16

Condeco також надає розширені можливості аналітики та звітності. Він збирає та аналізує дані про використання офісного простору, що дозволяє керівникам отримувати цінну інформацію про використання ресурсів, популярність певних зон або приміщень, а також прогнозувати потреби в майбутньому. Це допомагає приймати обґрунтовані рішення щодо розміщення робочих місць та оптимізації офісного простору.

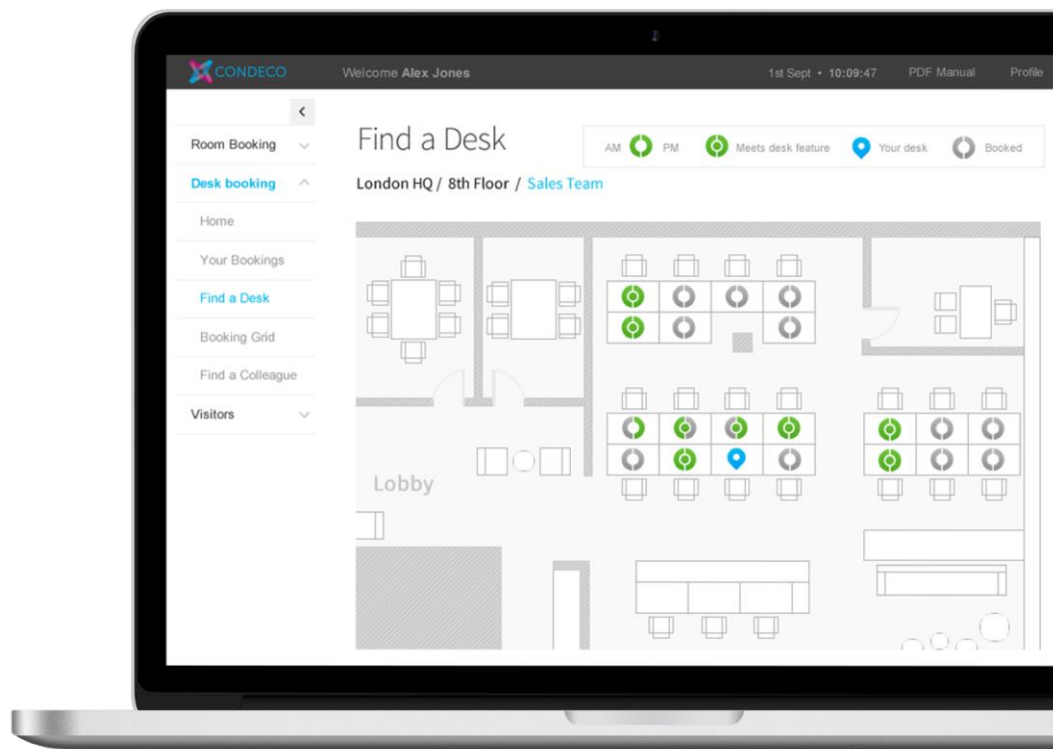


Рисунок 1.4– Конструктор планів офісу додатку Condeco [8]

#### Переваги:

- гнучкість, що дозволяє організаціям адаптувати систему під свої потреби;
- інтуїтивний та зручний інтерфейс для користувачів, що спрощує процес бронювання та керування просторами;
- можливість інтеграції з календарними сервісами та іншими корпоративними системами, що дозволяє автоматично синхронізувати дані та спрощує процес планування;

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		17

– оптимізація використання ресурсів та забезпечення ефективного використання простору;

– забезпечення точної та актуальної інформації про доступність просторів та зустрічей;

– сервіс має мобільний додаток, що дозволяє користувачам зручно бронювати простори та керувати своїми зустрічами і подіями безпосередньо зі своїх мобільних пристроїв.

Недоліки:

– Condeco є комерційним продуктом, і вартість його використання може бути високою, особливо для менших організацій або стартапів з обмеженим бюджетом;

– процес впровадження та налаштування може вимагати додаткових зусиль та ресурсів.

#### 1.4 Порівняльний аналіз

Під час пошуку аналогів до програмного забезпечення, було проаналізовано декілька рішень та побудовано порівняльну таблицю 1.1. Тому можемо оцінити за основними параметрами переваги та недоліки додатків розглянутих вище.

Таблиця 1.1 – Порівняльний аналіз

Функціонал/ Додаток	Robin	DeskFlex	Teem	Condeco
Резервування робочих місць	Так	Так	Так	Так

Резервування обладнання	Так	Так	Ні	Ні
Мобільний додаток	Так	Так	Так	Так
Інтеграція з календарями	Так	Так	Так	Так
Зручний інтерфейс	Так	Так	Так	Так
Складність використання	Середня	Середня	Проста	Середня
Вартість	Платно	Платно	Платно	Платно

### 1.5 Постановка задачі

Розробка вебзастосунку для оптимізації організаційного процесу в офісі шляхом керування резервуваннями робочих місць. Застосунок повинен забезпечувати можливість бронювання робочих місць працівниками через вебсайт.

Додаток повинен також надавати зручний інтерфейс для користувачів, мати зручний та гнучкий конструктор для створення планів та робочих місць.

Основною метою є покращення організації робочих процесів, уникнення зайнятості робочих місць та підвищення продуктивності працівників.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		19

Основні функціональні вимоги до застосунку:

- реєстрація та автентифікація користувачів;
- можливість створення та управління лобі, включаючи додавання, редагування та видалення інформації про лобі;
- планування робочих місць у лобі з можливістю додавання та видалення робочих місць;
- управління обладнанням, включаючи додавання, редагування та видалення інформації про обладнання;
- відображення деталей про кожне робоче місце та обладнання, такі як стан, власник, номер тощо;
- забезпечення безпеки та конфіденційності даних, використовуючи JWT token для автентифікації та авторизації користувачів.

#### Висновки до розділу 1

Після проведення аналізу існуючих рішень на світовому ринку, було виявлено, що на даний момент застосунків, які можуть вирішувати дану проблему мало, також вони більш орієнтовані на великі компанії.

Враховуючи вищевказане, для реалізації вебзастосунку для оптимізації організаційного процесу необхідно врахувати потреби користувачів, які можуть варіюватись від зручного інтерфейсу до можливості створення детальних планів приміщень.

У майбутньому даний застосунок можна розширити, а саме інтегрувати сторонні календарі та розробити повноцінний мобільну версію та додати додатковий функціонал конструктору.

В даному розділі виконана постановка задачі та визначені основні положення дипломного проєкту. Також було проведено аналіз існуючих схожих рішень. Проведений аналіз показав переваги та недоліки наявних рішень, які будуть враховані у нашому проєкті.

					ІА93.070БАК.004 ПЗ	Арк.
						20
Зм.	Лист	№ докум.	Підпис	Дата		

## 2 ОГЛЯД ТЕХНОЛОГІЙ РОЗРОБКИ

### 2.1 Вибір архітектури вебзастосунку

Вибір правильної архітектури дозволяє ефективно вирішувати задачі проекту та забезпечує багато переваг. Вона допомагає розподілити функціональність додатку на окремі модулі, що спрощує розробку та підтримку коду. Архітектура також забезпечує гнучкість та масштабованість, дозволяючи легко додавати нові функціональність і розширювати систему у майбутньому.

#### 2.1.1 Multi-Page Application

Архітектура МРА (Multi-page Application) є однією з популярних архітектур для веб-додатків, де кожна сторінка повністю перезавантажується при кожному запиті. У цій архітектурі кожна сторінка представляє окремий HTML документ зі своїм власним набором стилів, скриптів та контентом.

МРА базується на традиційній веб-моделі, де кожна сторінка є окремим файлом HTML, який відправляється з сервера до клієнта при кожному запиті. Коли користувач переходить з однієї сторінки на іншу, весь контент зберігається на сервері, і нова сторінка повністю завантажується.

Однією з основних переваг МРА є простота розробки та підтримки. Кожна сторінка є самодостатньою ізольованою одиницею, що спрощує розробку, тестування і модифікацію окремих частин додатку. Також МРА дозволяє використовувати традиційний підхід до створення веб-сторінок і не вимагає великих знань JavaScript або фронтенд-фреймворків.

Однак, МРА має кілька недоліків. Оскільки кожна сторінка повністю перезавантажується, це може призводити до деякого затримання відображення контенту для користувача. Також це може спричинити більше навантаження на сервер, оскільки при кожному переході весь контент потрібно відправляти повторно.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		21

Також, використання МРА може бути обмеженим для створення більш складних та інтерактивних інтерфейсів. Оскільки кожна сторінка повністю перезавантажується, важко створити багатоелементну взаємодію, де зміни в одній частині сторінки не впливають на інші без повного оновлення.

Загалом, архітектура МРА є простою та зрозумілою, і вона підходить для багатьох типів веб-додатків. Однак, вона може бути менш підходящою для складних та інтерактивних додатків, які вимагають більш плавної та динамічної взаємодії з користувачем.

На рисунку 2.1 наведено діаграму життєвого циклу МРА додатку

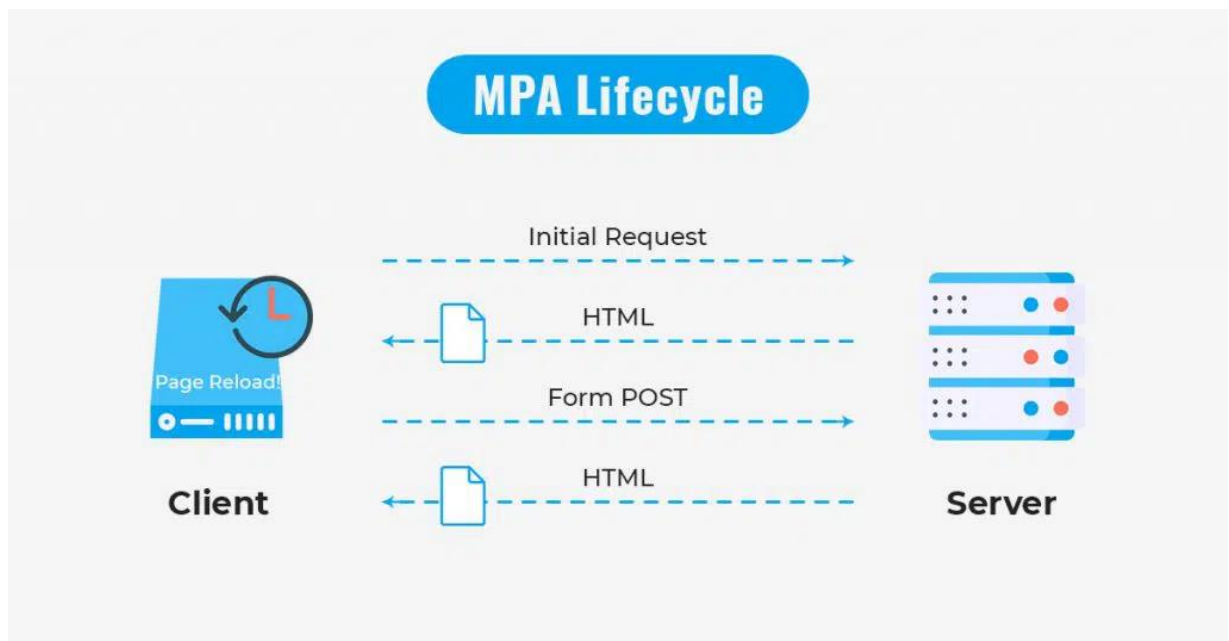


Рисунок 2.1 – Життєвий цикл МРА додатку [10]

### 2.1.2 Single-Page Application

Архітектура SPA (Single Page Application) є підходом до розробки веб-додатків, де весь контент завантажується одноразово при завантаженні сторінки, і подальша навігація та взаємодія з додатком відбуваються без перезавантаження сторінки (рисунок 2.2). Це відмінно від традиційного підходу, коли кожна дія користувача вимагає перезавантаження сторінки і запиту на сервер.

Однією з основних переваг SPA є збільшена швидкість та плавність взаємодії з додатком. Коли користувач взаємодіє з SPA, відбувається асинхронна взаємодія з сервером, завантажуються лише необхідні дані, а не вся сторінка цілком. Це дозволяє зменшити час відгуку та покращити загальний досвід користувача.

SPA забезпечує більш динамічну та інтерактивну взаємодію з додатком. За допомогою JavaScript-фреймворків та бібліотек, таких як React, Angular або Vue.js, можна побудувати складні інтерфейси з багатьма компонентами, які можна оновлювати асинхронно, без необхідності перезавантаження сторінки.

Також SPA дозволяє розробляти кросплатформені додатки. Оскільки весь фронтендовий код виконується в браузері, додатки можуть працювати на різних платформах, включаючи комп'ютери, планшети та смартфони, без потреби у розробці окремих версій для кожної платформи.

Звичайно, SPA також має свої недоліки. Оскільки весь код виконується в браузері, стартове завантаження додатку може займати більше часу, особливо якщо додаток має великий обсяг або вимагає багато залежностей. Також SPA може бути менш підходящим для SEO (пошукової оптимізації), оскільки сторінки генеруються динамічно за допомогою JavaScript і не завжди легко індексуються пошуковими системами.

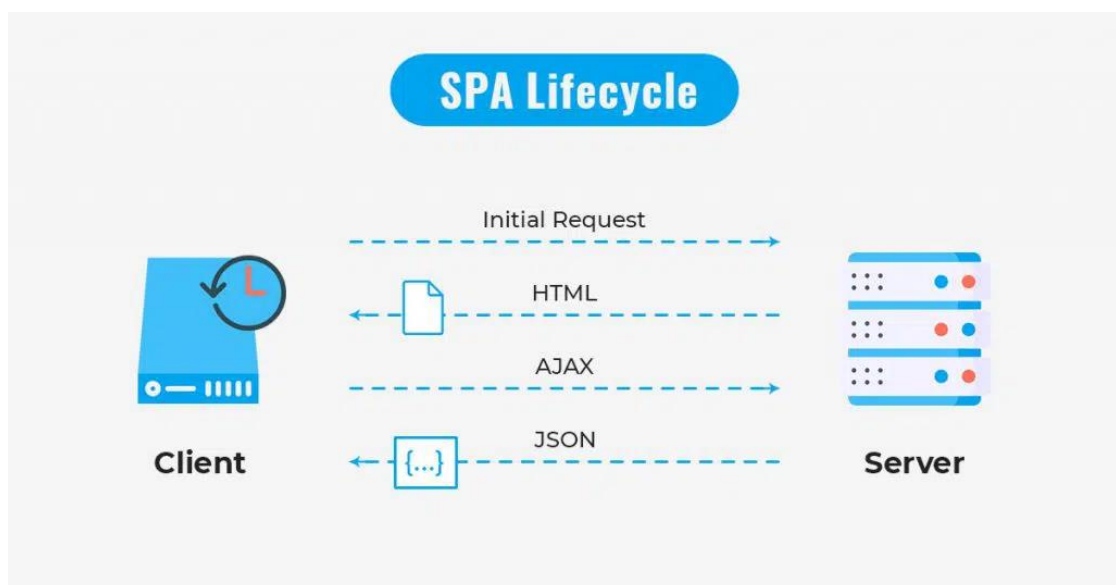


Рисунок 2.2 – Життєвий цикл SPA додатку [11]

### 2.1.3 SSR (Server–Side Rendering)

SSR (Server–Side Rendering) – це підхід, при якому генерація сторінок відбувається на сервері перед їх надсиланням на клієнтську частину. За допомогою SSR додатки генерують повний HTML–код сторінки на сервері, включаючи всі необхідні дані та запити. При отриманні запиту від користувача, сервер виконує генерацію сторінки та надсилає клієнтові готовий HTML, який може бути відразу відображений.

Основна перевага SSR полягає в тому, що користувачі отримують повністю згенеровані сторінки з вмістом, що динамічно генерується на сервері. Це поліпшує швидкість завантаження сторінок та забезпечує більш зручний досвід для користувачів. Також з точки зору SEO, розрендерені сторінки мають перевагу, оскільки пошукові системи можуть легше індексувати їх і враховувати вміст.

У сучасній архітектурі SSR, сервер виконує роль рендерера сторінок. Коли користувач робить запит до сервера, він отримує розрендерену сторінку, яка містить не тільки статичний контент, але й динамічні дані, отримані з бази даних або зовнішніх джерел. Цей процес забезпечує змінну та актуальну інформацію для кожного користувача.

SSR дозволяє краще керувати станом додатків та забезпечує високу безпеку. Серверний рендеринг дозволяє ініціювати та обробляти стан додатків на сервері, забезпечуючи його стійкість та незалежність від клієнтського пристрою. Це також дозволяє контролювати доступ до ресурсів сервера та виконувати необхідні перевірки, щоб забезпечити безпеку даних.

Хоча розробка за допомогою сучасної архітектури SSR може бути складнішою, ніж традиційний підхід до SPA (Single–Page Applications), вона пропонує багато переваг, зокрема поліпшену швидкість завантаження, покращений SEO та кращий контроль стану та безпеки.

					ІА93.070БАК.004 ПЗ	Арк.
						24
Зм.	Лист	№ докум.	Підпис	Дата		

SSR можна уявити як об'єднання двох попередніх архітектур SPA та MPA, оскільки дане рішення містить у собі одразу переваги обох архітектур, а саме SEO та користувацький досвід. Додатками, реалізованих за допомогою SSR, набагато зручніше користуватися та розроблювати.

У порівнянні з SPA, де рендеринг відбувається на стороні клієнта за допомогою JavaScript, SSR дозволяє забезпечити швидкодію і коректне відображення контенту відразу при завантаженні сторінки.

Користувачі відносно швидко можуть завантажити сторінку вже з базовим функціоналом, як до прикладу навігація, або статичні дані, а усі інші більш затратні запити можна реалізувати за допомогою CSR.

Життєвий цикл SSR запиту складається з кількох етапів, які описують процес обробки та відправки запиту на сервер, генерації сторінки та повернення результату на клієнтську частину (рисунок 2.3).

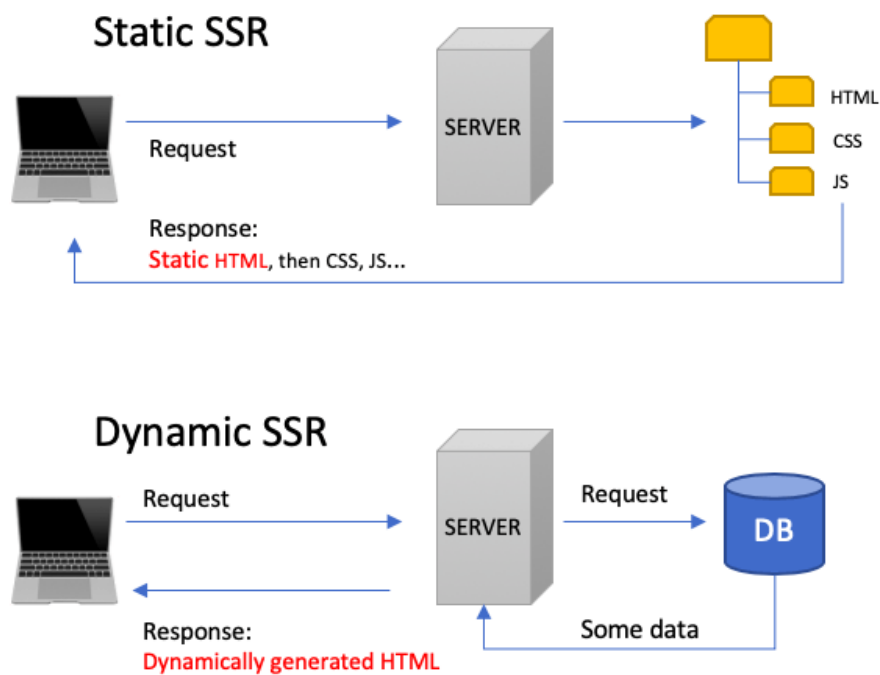


Рисунок 2.3 – Життєвий цикл SSR додатку [13]

Основні етапи циклу SSR запиту включають наступне:

- клієнтська сторона (браузер) ініціює запит до сервера, передаючи необхідні параметри та шлях до ресурсу;
- сервер отримує запит і виконує маршрутизацію, визначаючи, яка сторінка або компонент потребує рендерингу;
- якщо необхідно, сервер може звернутися до бази даних або інших зовнішніх джерел для отримання динамічних даних, які будуть включені у сторінку;
- на основі отриманих даних і шаблонів, сервер генерує повний HTML-код сторінки, включаючи статичний контент та динамічні дані;
- сервер надсилає згенерований HTML-код сторінки назад до клієнтської сторони, яка отримує його у відповіді на свій початковий запит.

#### 2.1.4 Підсумок вибору архітектури

Проаналізувавши найбільш популярні рішення архітектури для вебзастосунку, було створено порівняльну таблицю 2.1

Таблиця 2.1 – Порівняння архітектур

Критерій	SPA	MPA	SSR
Перше завантаження	Довго	Помірно	Помірно
CSR	Можливо	Неможливо	Можливо
Навігація	Без перезавантаження	Необхідне перезавантаження	Можливе перезавантаження.
SEO	Часткова	Так	Так

Навантаження на сервер	Низьке	Середнє, залежить від сторінки	Середнє, залежить від сторінки
Навантаження на клієнта	Високе	Низьке	Середнє
Складність розробки	Середня	Низька	Середня
Користувацький досвід	Середній	Низький	Високий

Під час розгляду основних архітектур веб-додатків, таких як MPA, SPA і SSR, була проаналізована їх особливість та переваги. У результаті цього аналізу обрано архітектуру SSR для даного проекту.

Основною причиною цього вибору є поліпшення швидкості завантаження сторінок та забезпечення кращого досвіду для користувачів. SSR дозволяє повністю генерувати сторінки з динамічним вмістом на сервері. Навіть якщо SEO не є пріоритетом для даного застосунку, SSR все ще пропонує переваги щодо кешування, оптимізації бекенду та безпеки даних.

Використання SSR надає можливість кращого керування станом додатків та забезпечує високу безпеку. Також дана архітектура дозволяє ефективно керувати доступом до ресурсів сервера та виконувати необхідні перевірки, що забезпечує безпеку збережених даних.

## 2.2 Вибір технологій розробки користувацької частини

Вибір технологій для розробки користувацької частини проекту є важливим етапом, що визначає подальші можливості та продуктивність додатку. Оскільки даний проєкт використовує архітектуру SSR, однією з найкращих варіантів для розробки клієнтської частини є використання фреймворку Next.js.

					IA93.070BAK.004 ПЗ	Арк.
						27
Зм.	Лист	№ докум.	Підпис	Дата		

Next.js надає повний набір інструментів для створення динамічних та ефективних користувацьких інтерфейсів. Цей фреймворк дозволяє розробникам працювати з React у поєднанні з серверним рендерінгом, що забезпечує швидке та передбачуване завантаження сторінок.

Next.js [15] – це розширення фреймворка React, яке надає можливість розробляти універсальні веб-додатки з використанням серверного рендерінгу (Server-Side Rendering, SSR). Цей фреймворк стає все більш популярним серед веб-розробників завдяки своїм потужним функціональностям та зручному способу роботи з React.

Однією з ключових особливостей Next.js є підтримка серверного рендерінгу, що дозволяє генерувати HTML на сервері та відправляти його до клієнтського браузера. Це дозволяє отримати передбачуваний та швидкий перший рендерінг сторінок, що сприяє покращенню SEO та користувацького досвіду.

Next.js також має вбудовану систему маршрутизації, яка спрощує навігацію в додатку. Ви можете використовувати стандартні роутери або створювати власні маршрути, що дає вам повний контроль над роутингом.

Ще однією корисною особливістю Next.js є автоматичне кодогенерація. Він автоматично оптимізує завантаження коду, розділяючи його на набір невеликих частин, які завантажуються за потребою. Це допомагає зменшити час завантаження сторінок та покращити продуктивність додатку.

Також Next.js надає велику кількість вбудованих функцій, таких як підтримка стилів CSS-модулів, гаряче перезавантаження (hot reloading), підтримка інкрементальної перебудови (incremental static regeneration) та багато іншого. Ви можете легко інтегрувати зовнішні бібліотеки та розширити функціональність свого додатку.

Загалом, Next.js є потужним і гнучким фреймворком для розробки веб-додатків з використанням SSR. Він надає зручність роботи з React, покращений SEO, швидкість завантаження сторінок та багато інших корисних

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		28

функцій. Це робить його одним з популярних виборів для створення сучасних і продуктивних додатків.

### 2.3 Вибір технологій розробки серверної частини

При виборі технологій для розробки серверної частини, були розглянуті два підходи – реалізація на Node.js та використання сервера від Next.js.

Node.js [16] є платформою для розробки серверних додатків з використанням JavaScript. Вона надає гнучкість та можливість побудови власної серверної архітектури. За допомогою Node.js, розробники можуть створювати власні сервери з використанням різних фреймворків та бібліотек.

Однак, порівняно з реалізацією сервера від Next.js, використання сервера з коробки має свої переваги. Next.js надає вбудовану підтримку серверного рендерінгу, що дозволяє ефективно розробляти SSR застосунки. Це спрощує процес розробки і дозволяє зосередитися на функціональності додатку, не витрачаючи час на настройку сервера.

Однією з ключових переваг використання сервера з коробки в Next.js є спрощення роботи з маршрутизацією. Next.js має вбудований маршрутизатор, який дозволяє легко визначити шляхи (routes) для різних сторінок та обробляти їх запити. Це зменшує необхідність вручну налаштовувати маршрутизацію і спрощує організацію структури додатку.

Крім того, Next.js надає готові рішення і структуру для обробки запитів і відповідей. Це означає, що ви можете зосередитися на розробці функціональності додатку, не витрачаючи багато часу на налаштування серверних аспектів. Готові рішення включають в себе обробку форм, кешування, обробку файлів та інші пов'язані задачі..

Ще однією перевагою використання сервера з коробки є підтримка серверного рендерінгу (Server– Side Rendering, SSR).

					ІА93.070БАК.004 ПЗ	Арк.
						29
Зм.	Лист	№ докум.	Підпис	Дата		

## 2.4 Вибір Бази даних

Вибір відповідної бази даних для нашого застосунку є критичним етапом у процесі розробки. Добре обрана база даних забезпечить ефективне зберігання та управління даними, а також вплине на продуктивність та масштабованість нашого застосунку. У наступному розділі ми розглянемо кілька можливих варіантів реалізації бази даних і порівняємо їх основні характеристики.

**Реляційні бази даних (RDBMS):** Цей тип баз даних, такий як MySQL, PostgreSQL або SQLite, використовує таблиці зі структурованими даними, зв'язками між ними та SQL-запитами для зберігання та операцій з даними. Реляційні бази даних надають сильну консистентність, надійність та можливості для складних запитів. Вони підходять для додатків з високими вимогами до даних та складними структурами.

**Нереляційні бази даних (NoSQL):** Цей тип баз даних, такий як MongoDB, Cassandra або Redis, використовує різні моделі для зберігання даних, такі як документи, ключ-значення або колонки. Нереляційні бази даних надають гнучкість у роботі зі змінюваними та непрогнозованими даними, а також високу швидкодію доступу до даних. Вони підходять для швидкого зберігання та обробки великого обсягу даних.

MongoDB [17] є нереляційною базою даних, яка використовує схему документів для зберігання даних. Вона розроблена для масштабованих та гнучких додатків, які працюють з незалежними структурами даних. Особливість MongoDB полягає у тому, що вона не вимагає жорсткої схеми даних, що дозволяє додавати, змінювати та видаляти поля з документів без необхідності модифікації всіх інших документів у колекції.

MongoDB базується на моделі ключ-значення, де кожен документ має унікальний ідентифікатор (ключ) та пов'язані з ним значення. Документи зберігаються у колекціях, які можуть бути логічно пов'язані між собою. MongoDB

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		30

надає широкі можливості для запитів та фільтрації даних за допомогою різних умов та індексів.

Однією з основних переваг MongoDB є горизонтальне масштабування. Вона дозволяє розподіляти дані на кілька серверів (шардів), що дозволяє обробляти великі обсяги даних та високі навантаження. MongoDB також надає можливості реплікації, що забезпечує високу доступність та надійність даних.

Додатково, MongoDB пропонує гнучкий механізм для зберігання різних типів даних, включаючи документи, масиви, вкладені документи та інші. Вона також підтримує географічні запити та індексацію просторових даних, що може бути корисним для розробки географічно орієнтованих застосунків.

MongoDB була обрана для даного проекту через її особливості та переваги. Нереляційна модель бази даних дозволяє гнучко працювати з даними, забезпечуючи швидкість розробки та адаптивність. Горизонтальне масштабування та можливості реплікації гарантують високу доступність та надійність системи.

Можливості зберігання різних типів даних та географічні запити розширюють можливості роботи з різноманітними даними та розробки функціоналу, пов'язаного з геолокацією. Усі ці фактори роблять MongoDB ідеальним вибором для даного проекту.

## 2.5 Вибір середовища розробки

При виборі середовища розробки важливо враховувати зручність інтерфейсу, розширення, які можуть автоматизувати або пришвидшити розробку.

Visual Studio Code [18] – це відкритий та розширюваний редактор, який підтримує багато мов програмування, включаючи JavaScript, Python, C#, Java та багато інших. Він працює на різних операційних системах, включаючи Windows, macOS і Linux.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		31

Однією з головних переваг VS Code є його легкість використання та інтуїтивний інтерфейс. Він надає зручне розташування редактора коду, панелей інструментів та вікон, що дозволяє розробникам ефективно працювати зі своїми проектами. Також VS Code підтримує велику кількість мов програмування та надає розширення для роботи з різними фреймворками та інструментами.

Інтегрована система керування версіями, така як Git, дозволяє розробникам легко відстежувати та керувати змінами в коді. VS Code також має потужні можливості для роботи з віддаленими серверами та контейнерами, що полегшує розробку та тестування додатків.

Ще одна важлива особливість VS Code – це його екосистема розширень. Розробники можуть встановлювати розширення для покращення функціональності редактора, додавання підтримки для конкретних мов програмування, фреймворків та інших інструментів. Це дозволяє налаштувати середовище розробки під свої потреби та підвищити продуктивність.

VS Code підтримує вбудоване налагодження коду, автодоповнення, форматування та багато інших корисних функцій, які спрощують розробку та поліпшують якість коду.

Отже, середовищем розробки було обрано Visual Studio Code, оскільки він є потужним та універсальним, надає розробникам зручні інструменти, має багатий набір функцій та можливість налаштування для підтримки різних мов програмування та фреймворків.

## Висновки до розділу 2

Під час процесу вибору архітектури додатку, технологій для користувацької та серверної частин, бази даних та середовища розробки, було здійснено ретельний аналіз та порівняння різних варіантів. З метою

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		32

досягнення поставлених цілей та задоволення вимог проекту, було прийнято наступні рішення.

Щодо архітектури додатку, визначилися з використанням серверного рендерингу (SSR), що дозволить поліпшити швидкість завантаження сторінок та забезпечити кращий досвід для користувачів.

У виборі технологій для користувацької частини, було вирішено скористатися сучасним фреймворком Next.js, що дозволяє легко реалізувати SSR, забезпечити розширюваність та підтримку багатьох функціональних можливостей.

У виборі технологій для серверної частини, було обрано використання серверу, який надає фреймворк Next.js, що дозволить швидко налагодити серверну інфраструктуру та забезпечити ефективну обробку запитів.

Щодо бази даних, було вирішено використовувати MongoDB, оскільки вона надає гнучкість, швидкість та масштабованість для зберігання та обробки даних застосунку.

					ІА93.070БАК.004 ПЗ	Арк.
						33
Зм.	Лист	№ докум.	Підпис	Дата		

## 3 РОЗРОБКА ТА ПРОЄКТУВАННЯ

### 3.1 Моделювання системи

Схема бізнес-процесів є незамінним інструментом в моделюванні системи, оскільки вона дозволяє візуалізувати послідовність та взаємодію різних етапів та компонентів бізнес-процесу. Вона надає уніфікований огляд процесу, його активностей та залежностей між ними, спрощуючи розуміння та аналіз діяльності системи.

Застосування схеми бізнес-процесів допомагає уточнити вимоги до системи та виявити потенційні проблеми. Вона дозволяє здійснити детальний аналіз кожного етапу процесу, виявити можливі ризики та прогнозувати їх вплив на систему. Крім того, схема бізнес-процесів допомагає знайти шляхи оптимізації та покращення ефективності роботи системи.

Використання схеми бізнес-процесів є особливо важливим на етапі проектування та впровадження системи. Вона дозволяє усвідомити всі складові процесу та їх взаємозв'язки, що сприяє розробці оптимальних рішень та вибору найбільш ефективних стратегій.

Крім того, схема бізнес-процесів стає важливим інструментом для комунікації між різними зацікавленими сторонами проекту. Вона дозволяє зрозуміло та конкретно відобразити всі етапи та дії, що відбуваються в системі, спрощуючи спілкування та забезпечуючи однозначне розуміння всіх учасників проекту.

Отже, використання схеми бізнес-процесів є важливим етапом в розробці та вдосконаленні системи, що допомагає покращити її ефективність, виявити та усунути потенційні проблеми та сприяє зрозумінню та спілкуванню між усіма зацікавленими сторонами.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		34

### 3.2 Опис сутностей системи

Після проведення аналізу діаграми діяльностей та схеми бізнес–процесів було виділено декілька сутностей (рисунок 3.1), які є важливими елементами нашої системи. Ці сутності включають користувачів, лобі, план приміщення, робоче місце та обладнання.

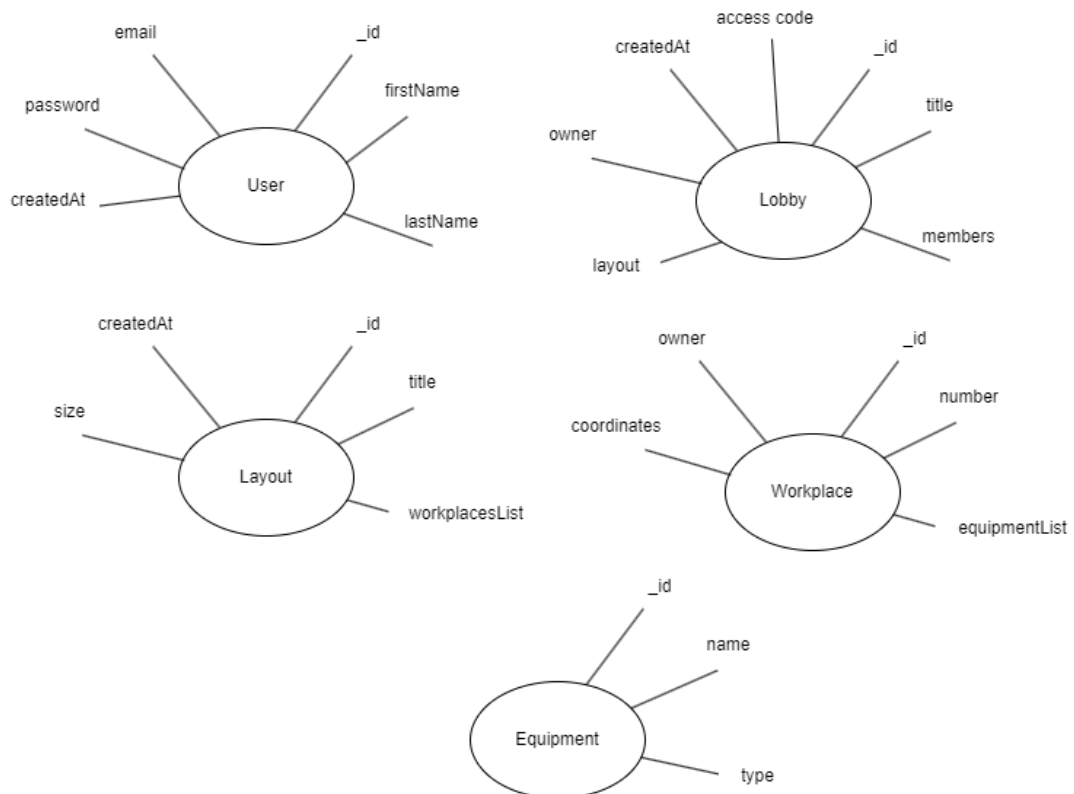


Рисунок 3.1– Схема сутностей

Сутність користувача має атрибути (Таблиця 3.1):

- `_id`, це унікальний ідентифікатор, що використовується для однозначної ідентифікації кожного запису в таблиці;
- `firstName`, це поле містить ім'я користувача;
- `lastName`, це поле містить прізвище користувача;
- `email`, це поле містить електронну адресу користувача, яка водночас є його логіном;

- password, це поле містить захешований пароль користувача;
- createdAt, це поле вказує на дату створення запису користувача.

Таблиця 3.1 – Опис моделі User

Назва	Атрибут	Тип
Первинний ключ	_id	objectId
Ім'я	firstName	string
Прізвище	lastName	string
Електронна пошта	email	string
Пароль	password	string
Дата створення	createdAt	date

Сутність кімнати має атрибути (Таблиця 3.2):

- \_id, це унікальний ідентифікатор, що використовується для однозначної ідентифікації кожного запису в таблиці;
- title, поле містить назву лобі;
- owner, поле містить інформацію про власника лобі;
- layout, поле містить інформацію про план приміщення, що є у цьому лобі;
- members, поле містить список учасників лобі;
- equipment, поле містить список обладнання, яке доступне в лобі;
- accessCode, поле містить автоматично згенерований код, який використовується для отримання доступу до лобі;
- createdAt, це поле вказує на дату створення кімнати.

Таблиця 3.2 – Опис моделі Lobby

Назва	Атрибут	Тип
Первинний ключ	_id	objectId
Назва	title	string
Власник	owner	string
План	layout	string

Дата створення	createdAt	date
Учасники	members	array
Список обладнання	equipment	array
Код доступу	accessCode	string

Сутність плану має атрибути (Таблиця 3.3):

- `_id`, це унікальний ідентифікатор, що використовується для однозначної ідентифікації кожного запису в таблиці;
- `size`, це поле містить інформацію про власника лобі;
- `workplacesList`, Це поле містить інформацію про усі робочі місця, які розташовані на плані;
- `members`, це поле містить список учасників лобі;
- `equipment`, це поле містить список обладнання, яке доступне в лобі;
- `createdAt`: Це поле вказує на дату створення плану.

Таблиця 3.3 – Опис типу Layout

Назва	Атрибут	Тип
Первинний ключ	<code>_id</code>	objectId
Розміри	<code>size</code>	object
Дата створення	<code>createdAt</code>	date
Робочі місця	<code>workplacesList</code>	array

Сутність робоче місце має атрибути (Таблиця 3.4):

- `_id`, це унікальний ідентифікатор, що використовується для однозначної ідентифікації кожного запису в таблиці;
- `owner`, це поле містить інформацію про власника лобі;
- `workplacesList`, атрибут містить інформацію про усі робочі місця, які розташовані на плані;

– coordinates, атрибут містить координати розташування робочого місця у приміщенні;

– reservedEquipment, це поле містить список обладнання, яке заброньоване за робочим місцем.

Таблиця 3.4 – Опис моделі Workplace

Назва	Атрибут	Тип
Первинний ключ	_id	objectId
Власник	owner	string
Номер	number	number
Координати	coordinates	object
Зарезервоване обладнання	reservedEquipment	array

Сутність обладнання має атрибути (Таблиця 3.5):

– \_id, це унікальний ідентифікатор, що використовується для однозначної ідентифікації кожного запису в таблиці;

– title, атрибут вказує на назву обладнання;

– type, атрибут вказує на тип обладнання.

Таблиця 3.5 – Опис моделі Equipment

Назва	Атрибут	Тип
Первинний ключ	_id	objectId
Назва	title	string
Тип	type	string

### 3.3 Серверна частина

Вибір правильної реалізації серверної частини є надзвичайно важливим етапом у розробці веб-застосунків. Ця складова відіграє ключову роль у забезпеченні ефективності, масштабованості та безпеки системи.

По-перше, серверна частина виконує обробку запитів від клієнтів. Це означає, що вона приймає дані, отримані від користувача, і проводить їх обробку відповідно до визначеної бізнес-логіки. Якщо серверна частина погано оптимізована або неспроможна швидко обробляти запити, це може призвести до погіршення продуктивності та незадовільного користувацького досвіду.

По-друге, серверна частина забезпечує доступ до бази даних, де зберігаються всі необхідні дані. Це означає, що вона повинна вміти ефективно взаємодіяти з базою даних, виконувати операції читання та запису даних. Вибір неадекватної технології або неправильна настройка серверної частини можуть призвести до проблем зі швидкістю доступу до даних або навіть до втрати інформації.

По-третє, серверна частина забезпечує безпеку даних. Вона повинна мати механізми для автентифікації та авторизації користувачів, захисту від атак та неправомірного доступу до інформації. Якщо серверна частина не належним чином захищена, це може призвести до компрометації конфіденційної інформації та порушення прав доступу.

Тому, правильний вибір реалізації серверної частини є критичним для успішної розробки веб-застосунків. Потрібно ретельно оцінювати вимоги проекту, вибирати ефективні технології та архітектурні рішення, які найкраще відповідають потребам системи. Це дозволить забезпечити швидку обробку запитів, ефективне використання ресурсів та високий рівень безпеки даних.

### 3.3.1 Архітектура REST API

Архітектура REST (Representational State Transfer) є одним із найпоширеніших підходів до проектування та розробки веб-сервісів. REST API (Application Programming Interface) використовує цю архітектуру для побудови розподіленої системи, яка забезпечує обмін даними між клієнтами і

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		39

сервером. REST API розробляється з урахуванням принципів REST, що забезпечують простоту, гнучкість та масштабованість системи.

Основна ідея REST полягає у тому, щоб розглядати ресурси (наприклад, об'єкти, колекції, послуги) як основну складову веб- додатку. Кожен ресурс має свій унікальний ідентифікатор (URL), який можна використовувати для доступу до нього. Клієнти можуть виконувати різні операції з ресурсами, такі як отримання, створення, оновлення та видалення, за допомогою HTTP- запитів.

Одним із ключових принципів REST є безстановність. Це означає, що кожен запит до сервера містить усю необхідну інформацію для його обробки, а сервер не зберігає жодної інформації про попередні запити клієнта. Це дозволяє покращити масштабованість та надійність системи.

Створення REST API включає в себе визначення ресурсів, використання HTTP- методів (GET, POST, PUT, DELETE) для взаємодії з ресурсами, встановлення правильної структури URL- адрес та використання формату даних, такого як JSON або XML, для обміну даними.

На рисунку 3.8 продемонстровано приклад використання моделі REST API.

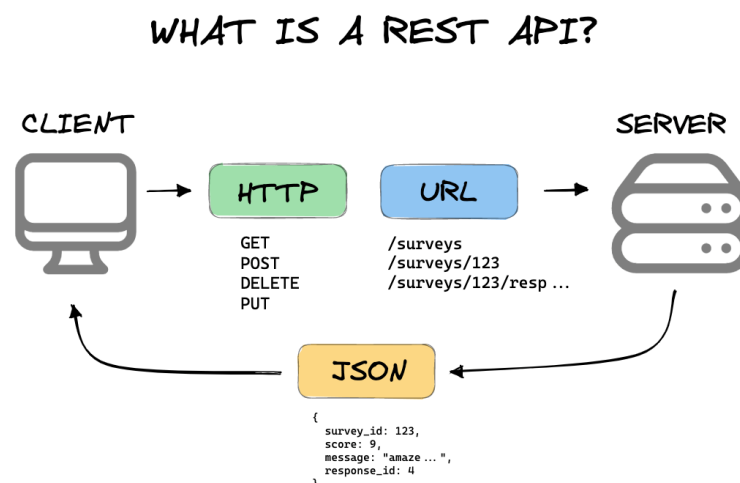


Рисунок 3.2 – Схема роботи REST API моделі [20]

При розробці REST API важливо правильно визначити ресурси, які будуть доступні клієнтам. Ресурси повинні бути логічно структуровані і відповідати функціональним можливостям системи. Наприклад, у веб-додатку можуть бути ресурси, що представляють користувачів, продукти, замовлення тощо.

Кожен ресурс має свої унікальні URL-адреси, які використовуються для доступу до них.

Управління даними в REST API зазвичай здійснюється за допомогою CRUD-операцій (створення, читання, оновлення, видалення). Клієнти можуть надсилати запити на створення нових ресурсів, отримання інформації про ресурси, оновлення даних ресурсів або їх видалення. Операції CRUD виконуються залежно від типу HTTP-методу, який вказується в запиті.

Також REST API може підтримувати автентифікацію та авторизацію, що дозволяє обмежувати доступ до ресурсів лише авторизованим користувачам або користувачам з певними правами доступу. Це забезпечує безпеку і контроль над використанням API.

### 3.3.2 API ендпоінти

Ендпоінти у API є специфічними URL-адресами або кінцевими точками вебсервера, до яких можна звертатися для виконання певних операцій або отримання даних з серверної частини програмного продукту. Ендпоінти в API можна класифікувати за категоріями, враховуючи тип даних, з якими вони працюють.

Для даного застосунку було створено необхідні ендпоінти, що наведені у таблицях (3.6 – 3.10)

					IA93.070BAK.004 ПЗ	Арк.
						41
Зм.	Лист	№ докум.	Підпис	Дата		

Таблиця 3.6 – Ендпоінти для роботи з користувачем

Ендпоінт	Тип	Опис
api/users/login	POST	Авторизація
api/users/register	POST	Реєстрація
api/users/:id	GET	Отримати дані користувача
api/users/delete/:id	DELETE	Видалення користувача

Таблиця 3.7 – Ендпоінти для кімнати

Ендпоінт	Тип	Опис
api/lobbies	GET	Отримати список кімнат
api/lobbies/create	POST	Створити кімнату
api/lobbies/:id	GET	Отримати дані про конкретну кімнату
api/lobbies/connect/:code	POST	Підключитись до кімнати

Таблиця 3.8 – Ендпоінти для плану офіса

Ендпоінт	Тип	Опис
api/layout/create	POST	Створити план офісу

Таблиця 3.9 – Ендпоінти для робочого місця

Ендпоінт	Тип	Опис
api/workplace/:id	GET	Отримати дані про робоче місце
api/workplace/create	POST	Створити робоче місце

api/workplace/edit/:id	PUT	Оновити дані робочого місця
------------------------	-----	--------------------------------

Таблиця 3.10 – Ендпоінти для додаткового обладнання

Ендпоінт	Тип	Опис
api/equipment/create	POST	Створення додаткового обладнання
api/equipment	GET	Отримати перелік обладнання
api/equipment/:id	GET	Отримати дані певного обладнання
api/equipment/delete/:id	DELETE	Видалити обладнання

### 3.3.3 Архітектура MVC

Для серверної частини даного додатку використовується архітектурний шаблон Model–View–Controller (рисунок 3.9) подібний підхід, який є характерним для Next.js. Цей шаблон розбиває програму на три основні компоненти: моделі (Models), представлення (Views) і контролери (Controllers). Використання подібного підходу допомагає досягти розділення відповідальності та полегшує розробку.

Опис структури та функціональності кожного компонента.

Моделі представляють бізнес–логіку та дані, що використовуються на сервері. Вони можуть включати схеми бази даних, моделі даних, взаємодію з зовнішніми сервісами тощо. Моделі відповідають за збереження та обробку даних, а також виконання бізнес–логіки. Вони можуть включати методи для

створення, зчитування, оновлення та видалення даних.

Представлення відповідають за генерацію відповідей сервера, які надсилаються клієнту. У Next.js використовуються сторінки (pages) для відображення різних шаблонів та компонентів React, які формують вміст сторінок. Представлення можуть отримувати дані від моделей та передавати їх до клієнта. Вони також можуть включати логіку маршрутизації та обробки запитів, наприклад, визначення шляхів та виклик відповідних контролерів для обробки запитів.

Контролери виконують логіку обробки запитів та взаємодіють з моделями для виконання операцій або отримання даних. Вони приймають запити від представлень або маршрутизатора, викликають відповідні методи моделей для отримання чи оновлення даних та передають результати назад до представлень. Контролери також можуть включати логіку валідації даних, обробки помилок та взаємодії з іншими сервісами.

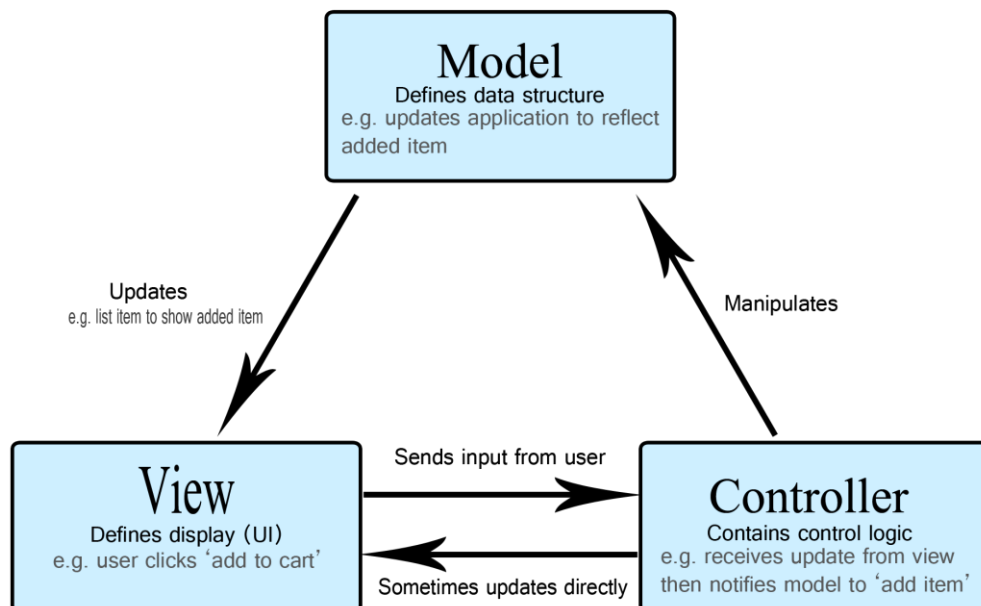


Рисунок 3.3 – Структура архітектури MVC[22]

Перевагами використання архітектури Model–View–Controller (MVC) в Next.js є:

- MVC допомагає розділити логіку програми на окремі компоненти, що полегшує розробку та забезпечує більшу організованість;
- за допомогою розділення на моделі, представлення та контролери, програма стає більш масштабованою, оскільки кожен компонент виконує свою визначену функцію;
- розділення коду на окремі компоненти дозволяє розробникам працювати паралельно над різними частинами програми, полегшуючи розробку та збільшуючи швидкість розробки;
- моделі, представлення та контролери можуть бути протестовані окремо, що спрощує процес тестування та налагодження програм;
- за допомогою MVC можна створювати різні компоненти перегляду для моделі, що дозволяє уникнути дублювання коду та розділити дані від бізнес–логіки.

Таким чином, архітектура Model–View–Controller є ефективним підходом до розробки серверної частини додатків на Next.js, який допомагає забезпечити розділення відповідальності, масштабованість та швидкість розробки.

### 3.4 Клієнтська частина

Вибір правильної реалізації клієнтської частини вебзастосунка є важливим кроком у процесі розробки. У даному проєкті було обрано фреймворк Next.js з метою забезпечити ефективну роботу з користувацьким інтерфейсом та взаємодією з користувачем.

Next.js є одним з найпопулярніших фреймворків для розробки інтерфейсів користувача в сучасному веб-розробці. Він пропонує широкий набір функцій та можливостей, які роблять розробку клієнтської частини

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		45

більш ефективною та продуктивною. Next.js підтримує серверний рендерінг та статичне генерування, що дозволяє забезпечити швидку відповідь і відмінну продуктивність вебзастосунка.

Крім того, Next.js надає зручність у роботі з роутингом, станом та стилізацією. Він має вбудовану підтримку для роботи зі станом за допомогою Redux Toolkit, що спрощує управління станом додатку та забезпечує зручну інтеграцію з React. Також Next.js надає можливість використовувати CSS модулі, CSS-in-JS бібліотеки або будь-який інший підхід до стилізації компонентів.

Один з головних факторів, який призвів до вибору Next.js для цього проєкту, є його висока продуктивність та швидкодія. Застосунок має потенціал для масштабування та обробки великого обсягу запитів від багатьох користувачів, і Next.js забезпечує оптимальну продуктивність навіть при високих навантаженнях.

Крім того, вибір Next.js дозволяє забезпечити безпеку застосунка шляхом використання JWT Token для авторизації. Цей механізм забезпечує захист від несанкціонованого доступу та зберігання даних користувачів в зашифрованому вигляді.

У підсумку, вибір реалізації клієнтської частини вебзастосунка є важливим кроком, який визначає продуктивність, ефективність та безпеку застосунка. Використання фреймворку Next.js дозволяє забезпечити потужний та швидкодіючий інтерфейс користувача, а також зручне управління станом та стилізацією. Крім того, використання JWT Token дозволяє забезпечити безпеку та авторизацію користувачів.

### 3.4.1 JWT

JWT (JSON Web Token) є стандартом для створення токенів, які використовуються для автентифікації та авторизації в веб– додатках. Він

					IA93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		46

базується на форматі JSON і має компактну структуру, що дозволяє передавати цифрові підписи та певні дані в зашифрованому вигляді.

JWT токени зазвичай використовуються для ідентифікації користувачів та дозволу доступу до ресурсів в захищеному API. Токен складається з трьох основних частин: заголовка, заявки і підпису.

У заголовку JWT вказується тип токена (JWT) і алгоритм шифрування, який використовується для підпису токена. Заявка містить корисну інформацію, таку як ідентифікатор користувача, його ролі, термін дії токена тощо. Ці дані можуть бути зашифровані або лише закодовані в форматі JSON.

Підпис JWT гарантує цілісність та автентичність токена. Він створюється за допомогою приватного ключа, який відомий лише серверу, і додається до токена. При отриманні токена сервер може перевірити його підпис, використовуючи публічний ключ, і перевірити його автентичність.

JWT токени мають декілька переваг. Вони компактні, легкі для передачі через мережу і мають можливість інкапсулювати потрібну інформацію. Також вони дозволяють серверу не зберігати стан сеансу, оскільки всі необхідні дані знаходяться в токені. Клієнтська сторона може зберігати токен в локальному сховищі і використовувати його для автентифікації в кожному запиті до сервера.

Однак, варто пам'ятати про безпеку JWT. Призначенням токена є передача інформації між сторонами, тому необхідно бути обережними щодо конфіденційної інформації, яку містить токен. Також важливо правильно налаштувати підпис та перевірку токена, щоб запобігти підробці та атакам.

В цьому проекті використання JWT токенів для автентифікації та авторизації користувачів є важливою складовою розробки клієнтської частини. Такі токени надають безпеку і зручність у взаємодії з серверним API, дозволяючи ідентифікувати та авторизувати користувачів у захищеному середовищі.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		47

### 3.4.2 Redux toolkit

Redux Toolkit - це набір інструментів, який надає зручний та продуктивний спосіб розробки клієнтської частини вебзастосунків з використанням Redux. Цей набір інструментів є популярним вибором серед розробників через свою здатність спростити та прискорити процес розробки, забезпечуючи одночасно гнучкість та надійність.

Однією з головних причин вибору Redux Toolkit є його простота в використанні. Він надає високорівневий API, який дозволяє розробникам працювати зі станом додатка та його змінами без необхідності вручну писати багато коду. Redux Toolkit включає в себе функції для автоматичного створення дій (actions), редюсерів (reducers) та допоміжних функцій для керування станом додатка, що значно спрощує процес розробки.

Другою важливою перевагою Redux Toolkit є його інтеграція зі зручними інструментами для відладки. Інструмент Redux DevTools дозволяє відстежувати та аналізувати зміни стану додатка, що робить процес відладки значно простішим та продуктивнішим.

Крім того, Redux Toolkit підтримує концепцію нормалізованої форми стану додатка. Це означає, що дані зберігаються у вигляді нормалізованого дерева, що полегшує роботу з даними та забезпечує ефективне управління станом додатка.

Однією з найважливіших переваг Redux Toolkit є його висока оптимізованість та ефективність, яка становить суттєву перевагу для розробки додатків. Це досягається завдяки вбудованій підтримці мемоізації селекторів (selectors), яка дозволяє ефективно керувати оновленнями даних та покращує швидкодію додатка.

Мемоізація селекторів забезпечує кешування результатів підрахунку, тому при зміні стану даних лише відповідні селектори будуть перераховані,

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		48

що дозволяє уникнути непотрібних обчислень та покращує продуктивність додатка.

Крім того, Redux Toolkit надає інші вбудовані оптимізації, такі як автоматична імутабельність даних та оптимізована обробка змін стану, що робить його потужним інструментом для створення швидких та ефективних додатків.

Узагальнюючи, вибір Redux Toolkit для розробки клієнтської частини вебзастосунків був зумовлений його простотою в використанні, інтеграцією з інструментами для відладки, підтримкою нормалізованої форми стану та високою продуктивністю. Використання Redux Toolkit дозволяє розробникам зосередитись на логіці додатка та зменшити кількість написаного коду, що робить його потужним інструментом для розробки клієнтської частини вебзастосунків.

### 3.4.3 React–konva

React Konva – це бібліотека, яка є обгорткою над бібліотекою Konva.js і дозволяє працювати з графікою та зображеннями в середовищі React. Вона надає розширені можливості для маніпулювання та відображення графічних елементів, таких як фігури, текст, зображення, шари, анімації та інші.

Основні причини використання React Konva:

– React Konva часто використовується для розробки конструкторів та редакторів, де користувачі можуть створювати та редагувати графічні об'єкти, такі як плани приміщень, схеми, діаграми та інші;

– бібліотека надає зручні інструменти для візуалізації даних у вигляді графіків, діаграм, карточок тощо. Вона дозволяє створювати інтерактивні та анімовані графічні елементи для зручного відображення даних;

– завдяки потужним можливостям візуалізації, React Konva може бути використана для створення простих ігор.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		49

### Переваги використання React Konva:

- React Konva нативно інтегрований з React. Це надає змогу легко створювати компоненти, керувати їх станом та оновлювати їх за допомогою React-методів, що спрощує розробку та підтримку коду;
- React Konva надає багатий набір функцій та можливостей для роботи з графічними елементами. Він надає можливість створювати фігури, лінії, текст, зображення та багато іншого, підтримує різноманітні операції, такі як масштабування, обертання, переміщення та зміна кольору. Це дозволяє створювати складні графічні ефекти та інтерактивні елементи;
- React Konva базується на вебграфічній бібліотеці Konva.js, яка володіє високою продуктивністю та оптимізацією. Вона використовує апаратне прискорення браузера для швидкої та ефективної візуалізації графіки. Це дозволяє створювати плавні та швидкі анімації, обробляти великі обсяги даних та забезпечувати відмінну реактивність і взаємодію з користувачем;
- React Konva дозволяє створювати власні компоненти та налаштовувати їх згідно потреб. Бібліотека дає можливість розширювати функціональність, додавати власні методи та властивості до компонентів, що дозволяє створювати повністю настроювані графічні інтерфейси.

### Недоліки використання React Konva:

- використання React Konva вимагає розуміння React-екосистеми та знання основ JavaScript;
- при створенні складних графічних інтерфейсів або великих обсягів даних, React Konva може вимагати значних ресурсів комп'ютера, що може впливати на продуктивність додатка;
- React Konva ґрунтується на бібліотеці Konva.js, тому для повноцінного використання бібліотеки потрібно мати розуміння основ Konva.js та його синтаксису.

Враховуючи різноманітні переваги та потенційні недоліки, можна зробити висновок, що React Konva є дуже потужною інструментальною

					ІА93.070БАК.004 ПЗ	Арк.
						50
Зм.	Лист	№ докум.	Підпис	Дата		

бібліотекою, яка надає широкі можливості для створення інтерактивних графічних елементів та візуалізації даних в середовищі React.

Завдяки своїй гнучкості та можливості маніпулювати графічними об'єктами, вона може бути особливо корисною для проєктів, які потребують створення складних конструкторів, ігор або візуалізації великого обсягу даних.

За допомогою React Konva розробники можуть забезпечити високу рівень інтерактивності та відповідності графічних елементів користувацьким діям, що поліпшує взаємодію з додатком та впливає на користувацький досвід.

Однак, при використанні React Konva слід враховувати можливість збільшення складності розробки, а також потенційні проблеми з продуктивністю, особливо при роботі з великим обсягом графічних об'єктів.

Тому, перед використанням React Konva, варто уважно оцінити специфіку проєкту та потреби користувачів для забезпечення оптимального вибору інструменту для візуалізації та роботи з графічними елементами у середовищі React.

#### 3.4.4 Структура клієнтської частини

Клієнтська частина додатка організована за такою структурою папок (рисунок 3.10):

- pages. У цій папці знаходяться компоненти, які представляють окремі сторінки додатка. Кожна сторінка має власний файл компонента, який містить логіку та відображення для цієї сторінки.
- assets. Тут розміщені ресурси, які використовуються в додатку, такі як стилі, зображення або шрифти.
- components. У цій папці знаходяться реактивні компоненти, які можуть бути використані на різних сторінках або в інших компонентах. Компоненти

можуть бути організовані в підпапки в залежності від їх функціональності або відповідальності.

– features. Тут знаходяться компоненти або модулі, які представляють функціональні частини додатка. Вони можуть включати компоненти, логіку та стан, які специфічні для конкретного функціоналу.

– slices. У цій папці розміщуються слайси для `redux-toolkit`, які представляють окремі частини стану додатка. Кожен слайс містить свій власний файл з логікою та діями.

– stores. Тут знаходиться основний файл для налаштування та створення `redux`-стору, який об'єднує всі слайси та управляє станом додатка.

Ця структура папок сприяє організації коду клієнтської частини додатка і робить його більш модульним та легким для розуміння та розширення. Вона допомагає уникнути дублювання коду, полегшує управління станом та забезпечує кращу організацію проєкту.

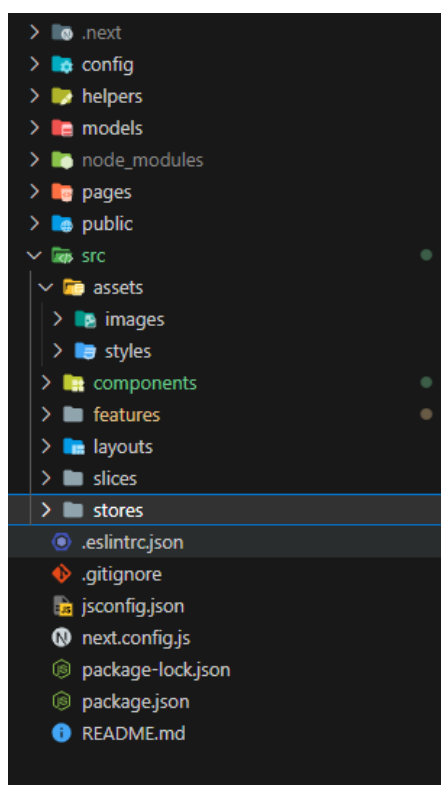


Рисунок 3.4 – Структура клієнтської частини

### 3.5 Схема бізнес процесів

Схему бізнес процесів наведено в графічному матеріалі, лист 2.

У діаграмі розписано 4 процеси:

- вхід у систему, процес авторизації та реєстрації користувачів;
- створення кімнати, процес створення кімнати для авторизованого користувача;
- бронювання робочого місця, процес, що виконується користувачем для резервування робочого місця;
- резервування обладнання, процес бронювання додаткової периферії користувачем.

### 3.6 Діаграма діяльності

Схему діяльності наведено в ІА93.070БАК.004 ДЗ.

Діаграма поділена для трьох можливих ролей:

- незареєстрований користувач має змогу лише зареєструватись;
- авторизований користувач, перегляд існуючих кімнат та підключення до них за допомогою коду;
- власник кімнати, процес створення кімнати та її редагування.

### 3.7 Діаграма послідовності

Діаграму послідовності наведено в ІА93.070БАК.004 Д4.

На діаграмі послідовності зображено взаємозв'язок Користувач – Frontend – Backend – MongoDB. Усі наведені процеси існують лише за умови, що користувач є зареєстрованим у базі даних.

Для авторизації користувач вводить логін та пароль, відправивши запит HTTP Request (“/login”, POST), сервер оброблює запит та звертається до бази

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		53

даних. В залежності від того чи існує даний користувач, чи правильно введені дані для авторизації буде повернено різну відповідь й повідомлено користувача на клієнтській частині.

Для створення кімнати відправляється запит HTTP Request (“/lobbies/create”, POST), після чого сервер зберігає дані у базі даних MongoDB та насилає відповідь користувачу про успішне створення кімнати.

### Висновки до розділу 3

Використання архітектури REST API дозволяє нам побудувати гнучку та масштабовану серверну частину додатку, забезпечуючи стандартизований спосіб взаємодії між клієнтом та сервером за допомогою HTTP методів. Це дозволяє нам ефективно передавати дані та забезпечувати гнучкість та зміну функціональності.

JWT token використовується для забезпечення безпеки та автентифікації в нашому додатку. Він дозволяє нам видавати та перевіряти токени, що містять інформацію про користувача, і забезпечувати безпечний доступ до ресурсів.

Redux Toolkit є потужним інструментом для керування станом додатку в React. Він надає нам зручність в роботі зі станом, зменшуючи кількість необхідного коду та спрощуючи процес управління станом додатку.

React Konva – це бібліотека, яка надає можливість легко робити маніпуляції з графікою та створювати розширені малювання на основі React. Вона надає нам потужні інструменти для роботи з візуалізацією та маніпуляцією графічних елементів.

Використання цих технологій дозволяє нам реалізувати потужний та гнучкий додаток з безпекою, зручним керуванням станом та розширеними можливостями візуалізації. Вони сприяють ефективному розробленню та підтримці додатку, забезпечуючи задоволення потреб користувачів. За допомогою архітектури REST API ми можемо створити гнучку та

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		54

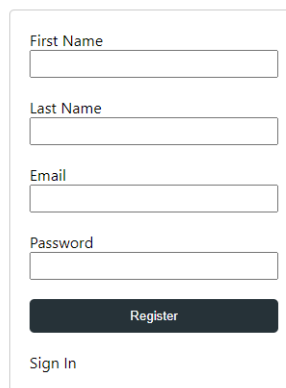
масштабовану серверну частину, яка забезпечує стандартизований спосіб взаємодії між клієнтом і сервером за допомогою HTTP методів. Це дозволяє ефективно передавати дані та забезпечувати гнучкість і зміну функціональності. JWT token використовується для забезпечення безпеки та автентифікації у нашому додатку. Він дозволяє видавати та перевіряти токени, які містять інформацію про користувача, тим самим забезпечуючи безпечний доступ до ресурсів. Redux Toolkit є потужним інструментом для керування станом додатку в React. Він спрощує управління станом та зменшує кількість необхідного коду. За допомогою React Konva ми можемо зручно маніпулювати графікою і створювати складні малюнки на основі React. Вона надає потужні інструменти для візуалізації та маніпуляції графічними елементами. Використання цих технологій дозволяє створити потужний і гнучкий додаток з безпекою, зручним управлінням станом та розширеними можливостями візуалізації. Вони сприяють ефективному розробленню та підтримці додатку, що задовольняє потреби користувачів.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		55

## 4 ОГЛЯД ВЕБЗАСТОСУНКУ

Для того щоб користуватися розробним застосунком необхідно зареєструватися, для цього потрібно натиснути на кнопку "Зареєструватися". Клацнувши на неї, користувач переходить на сторінку реєстрації, заповнює форму з обов'язковими полями, такими як ім'я, прізвище, електронна пошта та пароль (рисунок 4.1).

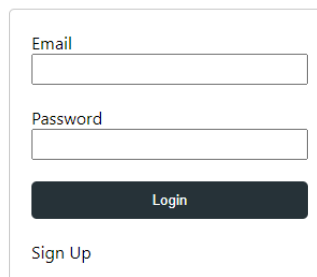
Після успішної реєстрації користувач може увійти до свого облікового запису. На сторінці авторизації вебзастосунку є форма для здійснення входу (рисунок 4.2). Заповнивши усі поля користувач успішно авторизований та перенаправляється на головну сторінку. Якщо дані введені не правильно, то з'явиться повідомлення з описом проблеми.



Registration form with the following fields and elements:

- First Name:
- Last Name:
- Email:
- Password:
- Register:
- Sign In: [Sign In](#)

Рисунок 4.1 – Форма реєстрації



Login form with the following fields and elements:

- Email:
- Password:
- Login:
- Sign Up: [Sign Up](#)

Рисунок 4.2 – Форма авторизації

Після авторизації користувач може переглянути список кімнат, до яких має доступ та кнопки "Створити кімнату" та під'єднатись. Клацнувши на цю опцію створення кімнати, користувач переходить на невеличку форму, де заповнює поля, в яких вказується інформація про кімнату. Після заповнення даних користувач може створити власний план офісу. Для створення планів використовується зручний конструктор, що дозволяє реалізувати усі потреби користувача.

Якщо користувач має код доступу, який він міг отримати від власника кімнати у такому разі він може під'єднатися до неї.

Увійшовши у кімнату користувач, може переглянути схему робочих місць. Обравши вільне місце користувач має змогу забронювати його, натиснувши на відповідну кнопку.

Після бронювання робочого місця користувач може звільнити його та зарезервувати додаткове обладнання.

#### Висновки до розділу 4

Було оглянуто використання застосунку, детально розібрано сценарії авторизації та реєстрації користувача, створення плану офісів, створення кімнати, підключення до кімнат за допомогою коду доступу, бронювання робочих місць та додаткової периферії.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		57

## ВИСНОВКИ

У процесі виконання дипломного проєкту було розроблено вебзастосунок для оптимізації організаційних процесів у офісі.

Було проаналізовано ряд існуючих додатків, які вирішували проблему організації робочого простору. Завдяки цьому було поставлено головні задачі, які повинен вирішувати даний застосунок. А саме:

- надавати змогу бронювати робочі місця;
- резервувати додаткове обладнання;
- мати зручний конструктор приміщень.

Було обрано технології розробки, які дозволяють вирішити поставлені цілі. Для зв'язку між сервером та клієнтом було обрано сучасну архітектуру API, клієнтська частина була реалізована за допомогою Next.js, також для забезпечення безпеки персональних даних було використано технологію JWT Token, яка дозволяє безпечно отримувати особисті дані користувача з сервера.

У результаті розробки застосунку було досягнуто мети та отриманий сервіс виконував усі вимоги для сучасного вебзастосунку, а саме має зручний інтерфейс та увесь необхідний функціонал для вирішення поставлених перед ним задач.

У підсумку, розроблений застосунок є потужним інструментом для покращення організаційних процесів у офісному середовищі. Він дозволяє зручно резервувати робочі місця та обладнання, а також використовувати інтуїтивно зрозумілий конструктор приміщень. Застосунок розроблений з використанням сучасних технологій, що забезпечують безпеку та ефективність. З його потенціалом для масштабування і розширення інфраструктури, він може стати незамінним інструментом для підвищення продуктивності та оптимізації робочих процесів у різних офісних середовищах.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		58

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. The Hybrid Workplace Experience Platform | Robin. URL: <https://robinpowered.com/>
2. Конструктор планів офісу додатку Robin URL: [https://assets-global.website-files.com/636bbf9c519296f08f480299/643d57ae5f7ab0bb5a0710a5\\_microsoft-teams-overview-p-1080.webp](https://assets-global.website-files.com/636bbf9c519296f08f480299/643d57ae5f7ab0bb5a0710a5_microsoft-teams-overview-p-1080.webp)
3. Office Hoteling Software | Desk Booking Software. URL: <https://www.deskflex.com/>
4. Конструктор планів офісу додатку DeskFlex URL: <https://cdn-dlpgi.nitrocdn.com/WtGQvvSZDKvnTdRJZYhTKQKpuplcDiTm/assets/images/optimized/rev-732c1dd/wp-content/uploads/2020/07/hover1.jpg>
5. Desk Booking Software Solutions | Teem by iOFFICE URL: <https://www.teem.com/solutions/space-booking-software/desk-booking-solutions>.
6. Конструктор планів офісу додатку Teem URL: <https://www.teem.com/wp-content/uploads/2022/03/desk-availability.png>
7. Desk Booking System Software & Apps – Condeco URL: <https://www.condecosoftware.com/> (дата звернення 05.05.2023).
8. Конструктор планів офісу додатку Teem URL: <https://www.teem.com/wp-content/uploads/2022/03/desk-availability.png>
9. Single Page Vs Multi-page Web Apps: What’s the Right Approach? URL: <https://www.hestabit.com/blog/single-page-vs-multi-page-web-apps/>.
10. Життєвий цикл МРА додатку URL: <https://www.hestabit.com/blog/wp-content/uploads/2021/08/MRA-Lifecycle-1024x536.jpg.webp>.
11. Життєвий цикл SPA додатку URL: <https://www.hestabit.com/blog/wp-content/uploads/2021/08/MRA-Lifecycle-1024x536.jpg.webp>.

					ІА93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		59

12. How Server side rendering (SSR) useful to improve lighthouse score?  
URL: <https://www.linearloop.io/blog/how-server-side-rendering-ssr-useful-to-improve-lighthouse-score/>.
13. ЖИТТЄВИЙ ЦИКЛ SSR додатку URL: <https://www.linearloop.io/wp-content/uploads/2021/05/How-it-works-1.png>
14. Single-page application vs. multiple-page application URL: <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>
15. Next.js by Vercel – The React Framework URL: <https://nextjs.org/learn/foundations/about-nextjs/what-is-nextjs>
16. Documentation – Node.js URL: <https://nodejs.org/en/docs>
17. MongoDB Documentation URL: <https://www.mongodb.com/docs/manual/>
18. Visual Studio Code URL: <https://code.visualstudio.com/>
19. REST APIs Explained – 4 Components URL: <https://mannhowie.com/rest-api>
20. Building a MVC Application with NextJs URL: <https://ianodad.medium.com/building-a-mvc-application-with-nextjs-and-prisma-a4be9276b9f2>
21. how/where to put models/controllers in next js app? URL: <https://stackoverflow.com/questions/60919884/how-where-to-put-models-controllers-in-next-js-app>
22. Структура архітектури MVC URL: <https://developer.mozilla.org/en-US/docs/Glossary/MVC/model-view-controller-light-blue.png>
23. Redux Toolkit URL: <https://redux-toolkit.js.org/>
24. Axios Docs URL: <https://axios-http.com/docs/intro>
25. React-konva URL: <https://konvajs.org/docs/react/Intro.html>
26. Docker: Accelerated, Containerized Application Development URL: <https://www.docker.com/>

					IA93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		60

27. Amazon AWC URL: <https://aws.amazon.com/>

28. Sequence Diagram Tutorial URL: <https://creately.com/guides/sequence-diagram-tutorial/>

29. NoSQL vs. SQL Databases URL: <https://www.mongodb.com/nosql-explained/nosql-vs-sql>

					IA93.070БАК.004 ПЗ	Арк.
Зм.	Лист	№ докум.	Підпис	Дата		61

## ДОДАТОК А

<https://github.com/Riptide-AI/OfficePlanManager>

Посилання на репозиторій з котом вебзастосунку

