

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

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Сергій СТИРЕНКО

(підпис)

“ \_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**Дипломний проєкт**

**на здобуття ступеня бакалавра**

**за освітньо-професійною програмою «Інженерія програмного  
забезпечення комп'ютерних систем»**

**спеціальності 121 «Інженерія програмного забезпечення»**

**на тему: «ВЕБ-додаток системи регулювання студентського дозвілля  
гуртожитку»**

Виконав:

студент IV курсу, групи ПІ-64

Зайченко Іван Володимирович

\_\_\_\_\_  
(підпис)

Керівник:

Асистент

Аленін Олег Ігорович

\_\_\_\_\_  
(підпис)

Консультант з нормоконтролю:

Професор, доктор технічних наук

Симоненко Валерій Павлович

\_\_\_\_\_  
(підпис)

Рецензент

\_\_\_\_\_  
(підпис)

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

Студент \_\_\_\_\_  
(підпис)

Київ – 2020 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ**  
**ІМ. ІГОРЯ СІКОРСЬКОГО»**

*Факультет інформатики та обчислювальної техніки*

*Кафедра обчислювальної техніки*

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Інженерія програмного забезпечення комп'ютерних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Сергій СТИПЕНКО

(підпис)

“    ” \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на дипломний проєкт студенту**

Зайченку Івану Володимировичу

1. Тема проєкту «ВЕБ-додаток системи регулювання студентського дозвілля гуртожитку»

керівник проєкту Аленін Олег Ігорович, асистент, затверджені наказом по університету від « 07 » травня \_\_\_\_\_ 2020р. № 1081-с

2. Термін здачі студентом закінченого роботи \_\_\_\_\_ 2020р.

3. Вихідні дані до проєкту технічне завдання, теоретичні дані.

4. Зміст пояснювальної записки: Дослідження існуючих рішень. Обґрунтування та вибір технологій розробки. Проектування архітектури програми. Проектування сховища даних. Реалізація логіки програми. Інструкція з користування. Ілюстрація розробленого додатку та шляхи вдосконалення.

5. Консультант роботи, з вказівкою розділів роботи, які до них вносяться

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	Симоненко В. П.		

6. Дата видачі завдання 01.09.2019 року

### КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів дипломного проекту (роботи)	Строк виконання етапів проекту(роботи)	Примітки
1.	<i>Затвердження теми роботи</i>	01.08.2019	
2.	<i>Вивчення та аналіз завдання</i>	15.09.2019-15.03.2020	
3.	<i>Розробка архітектури та загальної структури систем</i>	15.03.2020-25.03.2020	
4.	<i>Розробка структур окремих Підсистем</i>	25.03.2020-05.04.2020	
5.	<i>Програмна реалізація системи</i>	05.04.2020-15.04.2020	
6.	<i>Оформлення пояснювальної записки</i>	15.04.2020-20.05.2020	
7.	<i>Передзахист</i>	26.05.2020	
8.	<i>Захист</i>	25.06.2020	

Студент

Іван ЗАЙЧЕНКО \_\_\_\_\_

(підпис)

Керівник

Олег АЛЕНІН \_\_\_\_\_

(підпис)

## **Анотація**

Даний дипломний проект присвячений розробці веб-додатку системи регулювання студентського дозвілля в гуртожитку.

В бакалаврській роботі створено багатосторінковий веб-сайт для ефективного та зручного управління студентського активного відпочинку.

Програма дозволяє керувати резерваціями діяльності в зрозумілому для користувача інтерфейсі. Веб-додаток був розроблений за допомогою мови програмування Java та фреймворку Spring для серверної частини, а для клієнтської частини використовувалися JavaScript, HTML, CSS і Thymeleaf.

## **Annotation**

This diploma project is dedicated to the development of a web-based application of the student dormitory regulation system.

The bachelor's work has created a multi-page website for efficient and convenient management of student leisure activities.

The application allows you to manage activity bookings in a user-friendly interface. The web application was developed by using Java programming language and Spring framework for the server side, and JavaScript, HTML, CSS and Thymeleaf were used for the client side.

## ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проєкт	2	
2	A4	ІАЛЦ.467100.001 ВП	Відомість проєкту	1	
3	A4	ІАЛЦ.467100.002 ТЗ	Технічне завдання	4	
4	A4	ІАЛЦ.467100.003 ПЗ	Пояснювальна записка	70	
5	A3	ІАЛЦ.467100.004 Д1	Алгоритм веб-додатку	1	
6	A3	ІАЛЦ.467100.005 Д2	Функціональна схема, діаграма класів	1	
7	A3	ІАЛЦ.467100.006 Д3	Структурна схема клієнт-серверного веб-додатку	1	
8	A4	ІАЛЦ.467100.007 Д4	Текст програми	11	

					<b>ІАЛЦ.467100.001 ВП</b>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Зайченко І.В.</i>			<i>ВЕБ-додаток системи регулювання студентського дозвілля гуртожитку</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Аленін О.І.</i>					<i>5</i>	<i>1</i>
<i>Н. Контр.</i>		<i>Симоненко В.П.</i>			<i>Відомість дипломного проєкту</i>	<i>НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ, гр.ІІІ-64</i>		
<i>Затверд.</i>								

# **ТЕХНІЧНЕ ЗАВДАННЯ**

**до дипломної роботи  
освітньо-кваліфікаційного рівня бакалавр**

на тему: “ВЕБ-додаток системи регулювання студентського дозвілля  
гуртожитку”

Київ – 2020 року

## ЗМІСТ

1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ .....	2
2. ПІДСТАВИ ДЛЯ РОЗРОБКИ .....	2
3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ .....	2
4. ДЖЕРЕЛА РОЗРОБКИ .....	2
5. ТЕХНІЧНІ ВИМОГИ .....	3
5.1. Вимоги до продукту що розробляється.....	3
5.2. Вимоги до програмного забезпечення.....	3
5.3. Вимоги до апаратної частини .....	3
6. ЕТАПИ РОЗРОБКИ .....	4

					<i>ІАЛЦ.467100.002 ТЗ</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Зайченко І. В.</i>			<i>ВЕБ-додаток системи регулювання студентського дозвілля гуртожитку Технічне завдання дипломного проєкту</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Аленін О. І.</i>				7	4	
<i>Н. Контр.</i>		<i>Симоненко В. П.</i>				<i>НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ, гр.ІІІ-64</i>		
<i>Затверд.</i>								

## 1. НАЙМЕНУВАННЯ ТА ОБЛАСТЬ ЗАСТОСУВАННЯ

Дана технічна робота поширюється на розробку, впровадження та підтримку «Веб-додатку системи регулювання студентського дозвілля в гуртожитку». Область застосування: використання користувачами з студмістечку НТУУ КПІ для ефективного бронювання дозвілля в гуртожитку.

## 2. ПІДСТАВИ ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на виконання роботи кваліфікаційно-освітнього рівня «бакалавр програмної інженерії», затверджене кафедрою обчислювальної техніки Національного технічного Університету України «Київський Політехнічний інститут ім. Ігоря Сікорського».

## 3. МЕТА І ПРИЗНАЧЕННЯ РОЗРОБКИ

Метою даного проекту є розробка клієнт-серверної системи для зручного та ефективного регулювання студентського дозвілля в гуртожитку.

Розробка призначена для адміністрації гуртожитку та студмістечка, а також в першу чергу, для студентів, які планують вести активну діяльність після навчального процесу.

## 4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом розробки є науково-технічна література з даної тематики, програмування на мов Java, JavaScript та фреймворку Spring, публікації в періодичних виданнях та в Інтернеті з даних питань.

					ІАЛЦ.467100.002 ТЗ	Акл.
Змн.	Арк.	№ докум.	Підпис	Дата		2

## 5. ТЕХНІЧНІ ВИМОГИ

### 5.1. Вимоги до продукту, що розробляється

- Зручний та зрозумілий користувацький інтерфейс.
- Збереження резервації подій з прив'язкою до часу, користувача, кімнати та гуртожитку
- Регулювання, додавання, зміна та видалення існуючих власних бронювань.
- Адмінський контроль за подіями, студентами, кімнатами, та гуртожитками.

### 5.2. Вимоги до програмного забезпечення

- Операційна система MS Windows/Linux
- Java 8+
- PostgreSQL 12+
- IntelliJ Idea 2019

### 5.3. Вимоги до апаратної частини

- Процесор Intel або AMD.
- Оперативна пам'ять не менше 2Гб.
- Вільне місце на жорсткому диску не менше 30 Гб.

					ІАЛЦ.467100.002 ТЗ	Акл.
Змн.	Анк.	№ докум.	Підпис	Дата		3

## 6. ЕТАПИ РОЗРОБКИ

	Дата
Вивчення літератури	20.12.2019
Складання і узгодження технічного завдання	20.12.2019
Створення модулів системи, що розробляється	12.02.2020
Тестування окремих модулів системи	20.03.2020
Допрацювання, налагодження і виправлення помилок	02.04.2020
Оформлення документації дипломної роботи	17.04.2020

					ІАЛЦ.467100.002 ТЗ	Акл.
Змн.	Арк.	№ докум.	Підпис	Дата		4

**ПОЯСНЮВАЛЬНА ЗАПИСКА**  
до дипломного проєкту  
на тему: «ВЕБ-додаток системи регулювання  
студентського дозвілля гуртожитку»

Київ – 2020 року

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	3
ВСТУП.....	5
РОЗДІЛ 1 АНАЛІЗ СУЧАСНИХ ІСНУЮЧИХ РІШЕНЬ .....	8
1.1. Опис завдання веб-системи .....	8
1.2. Аналіз існуючих рішень .....	9
1.2.1. Менеджер Microsoft Outlook .....	10
1.2.2. Сервіс Google Calendar .....	15
1.2.3. Jorte Calendar .....	20
ВИСНОВКИ ДО РОЗДІЛУ 1 .....	25
РОЗДІЛ 2 АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ДЛЯ РІШЕННЯ ПРОБЛЕМИ.....	27
2.1. Завдання та технології .....	27
2.2. Вибір мови програмування.....	28
2.2.1. Мова програмування Java.....	28
2.2.2. Мова програмування Python.....	30
2.2.3. Мова програмування C#.....	32
2.3. Вибір системи управління бази даних .....	34
2.3.1. СУБД PostgreSQL .....	34
2.3.2. СУБД MySQL.....	37
2.3.3. СУБД SQLite .....	40
2.4. Вибір технології для клієнтської частини .....	42
2.4.1. Мова програмування JavaScript .....	42
2.4.2. Веб-фреймворк AngularJS .....	44
2.4.3. Веб-фреймворк Vue.js.....	46
ВИСНОВКИ ДО РОЗДІЛУ 2.....	48
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ.....	49
3.1. Функціональні вимоги до додатку.....	49

					<i>ІАЛЦ.467100.003 ПЗ</i>			
<i>Зм.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розроб.</i>		<i>Зайченко І. В.</i>			<i>ВЕБ-додаток системи регулювання студентського дозвілля гуртожитку Пояснювальна записка дипломного проекту</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевір.</i>		<i>Аленін О. І.</i>				12	70	
<i>Н. Контр.</i>		<i>Симоненко В. П.</i>			<i>НТУУ «КПІ ім. Ігоря Сікорського», ФІОТ, гр.ІІІ-64</i>			
<i>Затверд.</i>								

3.2.	Опис архітектури веб-додатку .....	50
3.3.	Серверна частина додатку .....	55
3.3.1.	Model архітектури MVC веб-застосунку .....	55
3.3.2.	Service, Repository, Controller веб-додатку .....	58
	ВИСНОВКИ ДО РОЗДІЛУ 3 .....	60
	РОЗДІЛ 4 АНАЛІЗ РЕЗУЛЬТАТІВ СТВОРЕНОГО ВЕБ-ЗАСТОСУНКУ ....	61
4.1.	Ілюстрація роботи програми (інструкція з користування) .....	61
4.2.	Тестування програмного забезпечення.....	67
4.3.	Порівняння з існуючими аналогами .....	68
4.4.	Рекомендації для подальшого вдосконалення та підтримки .....	69
	ВИСНОВКИ ДО РОЗДІЛУ 4.....	70
	ВИСНОВКИ .....	71
	СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	73
	ДОДАТКИ	

					ІАЛЦ.467100.003 ПЗ	Акл.
						2
<i>Змн.</i>	<i>Анк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Java - об'єктно-орієнтована мова програмування для створення, компіляції та запуску програм.

Spring - фреймворк для мови програмування Java для створення веб-додатків з використанням інверсії управління та іншими додатковими можливостями.

IntelliJ Idea – середовище розробки для мов програмування, зокрема Java.

PostgreSQL – об'єктно-реляційна система для керування базами даних.

Front end – частина програми взаємодії з користувачем.

Back end – сервісна частина для обробки логіки програми.

User interface – користувацький графічний інтерфейс, який зображується перед юзером для отримання запитів.

SQL - мова для взаємодії бази даних з програмою.

JVM – віртуальна машина Java – основна частина виконуючої системи Java.

API – прикладний програмний інтерфейс, який допомагає сервісам взаємодіяти між собою та обмінюватися даними

Фреймворк – це певний написаний програмний код, який полегшує створення, підтримку та зміну продукту

JSON - формат обміну даними між серверами, програмами, сервісами.

HTTP – протокол передачі гіпер-текстових документів, для передачі та отримання запитів у мережі

HTML – стандартна мова розмітки для розробки сторінок та сайтів у мережі інтернет.

ACID – атомарність, узгодженість, ізольованість, довговічність – це принципи, які гарантують коректну та правильну роботу транзакцій

MVC – шаблон проектування, який забезпечує розподіл програми на 3 частини: контролер, вид, модель

					ІАЛЦ.467100.003 ПЗ	Арк
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

Spring Data – модуль фреймворку Spring, який відповідає за збереження, обробку та видачу даних

Spring Security – модуль фреймворку Spring, який відповідає за авторизацію та аутентифікацію

Spring Boot – модуль фреймворку Spring, який відповідає за взаємодію компонентів та надання додаткових можливостей для управління та запуску програми, зокрема надання власного серверу.

Spring Transaction – модуль фреймворку Spring, який відповідає за коректну роботу транзакцій

					ІАЛЦ.467100.003 ПЗ	Арк
						4
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## ВСТУП

На теперішній час у сучасному світі, завдяки технологіям людина полегшує своє життя, як під час роботи, так і в повсякденній діяльності. Кожен з нас має можливість онлайн заплатити за комунальні послуги, купити авіа-білети, навіть можливо разом переглядати фільм з близькою людиною, знаходячись за тисячі кілометрів один від одного.

Але сумно визнавати те, що в таку «епоху технологій» студент, який мешкає в гуртожитку немає можливості забронювати онлайн кімнату для відпочинку, щоб активно провести свій вільний від навчання час.

Зрозуміло, що активний відпочинок необхідний в житті кожного студента, щоб тримати своє тіло у формі, берегти своє здоров'я та проводити час з друзями, адже від цього також залежить, як людина буде якісно працювати та навчатися. Окрім цього, постійне навантаження на мозок є навіть небезпечним, адже це призводить до нервозності, стресам та поганому самопочуттю. А якщо до цього додати, ще високий ритм життя, погіршену екологію, то психологічне розвантаження є обов'язковим для кожного з нас. І найкращий спосіб для цього є активний відпочинок, який може надати нам і, в свою чергу, гуртожиток, надаючи хоч і невеликий, але все ж різноманітний спектр діяльності у вільний час – це зайняття танцями, музикою, йогою, гра у пінг-понг та зайняття у спортзалі

На разі, щоб зарезервувати дану подію, потрібно пройти довгі та багатоетапні кроки: знаходження людини, яка відповідає за кімнату, очікування, коли дана людина з'явиться в мережі, контактування з нею, розписування вільних годин, домовленість з нею, отримання ключів і так далі... А ще можна згадати про неможливість зміни часу, людський фактор при виникненні помилок та можливі недружні відносини, з так званим «адміністратором». Проблеми можна і далі перераховувати, адже їх вистачає, саме тому бронювання кімнати, сьогодні, є процесом, який хочеться змінити у кращу сторону, і саме з цієї причини виникла ідея написання даного дипломного проекту.

					ІАЛЦ.467100.003 ПЗ	Арк
						5
Змн.	Арк.	№ докум.	Підпис	Дата		

Дану проблему необхідно розглянути з трьох різноманітних аспектів і в кожному з можна зробити висновок про актуальність роботи.

Перша цільова аудиторія - це студенти, які бажають легко зарезервувати час, для їхнього активного відпочинку в гуртожитку. Вони після навчального процесу, мають бажання, якнайменше свого вільного часу витратити на такі дрібниці, як саме бронювання, адже цей час можна витратити на щось більш корисне, наприклад, для саморозвитку або вивчення нового матеріалу, який допоможе їм стати кращим. Також, є студенти, які поєднують навчання з роботою, для яких важлива кожна хвилина відпочинку.

Необхідно не забувати про студентів, які відслідковують процес резервації в гуртожитках в режимі офлайн. На їхні плечі впала додаткова робота та навантаження, які вони мають сумлінно та вчасно виконувати. Але, вони особистості з своїми цілями та завданнями – зокрема вони повинні виконувати свій навчальний план, вивчати свою майбутню спеціальність, а також проводити свій час з близькими та друзями. І зрозуміло, що очікувати повідомлення для бронювання, кожную хвилину, кожную секунду – це неприйнятно та абсурдно.

З останньої сторони – це адміністрація гуртожитку. Їм необхідно мати контроль над системою резервацій для своїх потреб, зокрема мати повну інформацію, хто, коли і з якою метою бронював певну кімнату відпочинку. Це може стати в нагоді при пошкодженні майна, щоб винуватець, який заподіяв шкоду зміг виправити свою помилку та відшкодувати незручності, який він завдав.

Це не всі проблеми наявної системи, але ті, на яких однозначно необхідно звернути увагу, щоб впевнитися в тому, що дана робота є актуальною для всіх гуртожитків.

Отже, проблема була найдена, тепер була необхідність її вирішити. В даній роботі будуть представлені дослідження та висновки, які я робив, щоб створити веб-сайт, щоб полегшити життя всім, хто приймає участь в

					ІАЛЦ.467100.003 ПЗ	Арк
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

резервуванні кімнати для відпочинку, таких як пінг-понг, танцювальна кімната, кімната, де можна займатися на музичних інструментів та спортзал.

Даний веб-сайт повинен бути інтуїтивно зрозумілим для користувача та мати привабливий графічний інтерфейс, щоб ним хотіли користуватися. Веб-додаток повинен працювати стабільно і надійно, забезпечувати всім вимогам, як і програмним так і користувацьким вимогам.

Першочерговим завданням є створення веб-сайту системи регулювання студентського дозвілля в гуртожитку, який буде написаний за допомогою сучасних технологій, які широко використовуються у світі. Дана робота буде включати в себе розробку, як клієнтської, так і серверної частини програми, що дозволяє отримати продукт, який можна буде використовувати у подальшому житті студентства.

Популярні аналоги, які використовуються сьогодні та дослідження стеку технологій будуть освітлені в наступних розділах.

					ІАЛЦ.467100.003 ПЗ	Арк
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 1

### АНАЛІЗ СУЧАСНИХ ІСНУЮЧИХ РІШЕНЬ

#### 1.1. Опис завдання веб-системи

Веб-додаток системи регулювання студентського дозвілля в гуртожитку був створений з метою спростити бронювання кімнат активного відпочинку мешканцями гуртожитку. Веб-сайт надає можливість авторизованому користувачеві зарезервувати кімнату для свого відпочинку, перед цим переглянувши розклад та наявні бронювання.

За необхідності студент може змінити час, перенісши свою резервацію на більш зручний для себе час. Також додаток надає можливість видалити своє бронювання, при необхідності. Авторизований користувач, може переглянути свої бронювання, а також переглянути інформацію про свій аккаунт, наявні кімнати та детальніше дізнатися про можливості та характеристики кожної з кімнат відпочинку.

Також для даної програми необхідний адміністратор, який буде забезпечувати контроль та доповнювати контент для веб-сайту. Окрім тих дій, який має і звичайний користувач, адміністратор має певні переваги та додаткові можливості для управління додатком, а саме. Він може редагувати інформацію про студентів, кімнат, гуртожитків. Може додавати нові види відпочинку. А також особливістю програми є те, що добавляти нових учасників має право тільки адміністратор, щоб забезпечити безпеку та конфіденційність університетських даних. Детальніші можливості кожної ролі будуть описані в інших розділах.

Програмна частина розділена на дві частини: на клієнтську та на серверну, що має багато переваг, з якими детальніше можна ознайомитися в наступних розділах та в мережі інтернет. Клієнтська частина надає користувачеві зручний графічний інтерфейс, а серверна частина відповідає за коректну роботу бізнес-логіки.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		8

## 1.2. Аналіз існуючих рішень

На сьогодні, є багато додатків, веб-сайтів та навіть мобільних додатків, які спрощують наше життя. Тому, не дивно, що існує достатньо аналогів мого додатку в різних вищеперерахованих типах. Також всі додатки мають свій унікальний дизайн, який приваблює клієнтів та мають багато різного функціоналу, який хоч і відрізняється і є унікальним, але має одну ідею та мету.

Дані програми були розроблені командами професіоналів, які задіяли найкращі свої практики, щоб створити надійний та актуальний продукт, який сподобається користувачам і буде надовго вжитку. Кожна команда проводила дослідження ринку, актуальності та інші розрахунки, щоб отримати найбільшу вигоду для себе та своєї аудиторії користувачів чи замовника.

Багато з даних програм, які будуть переліковані, хоч і мають необхідний функціонал, але вони не призначені саме для вирішення даної проблеми. Окрім того, мають кілька обмежень, які докорінно будуть впливати на вибір додатку та його впровадження.

І хоч, величезний спектр існуючих рішень присутні, але важко ментально пригадати, хоч один веб-сайт, який вирішуватиме дану проблему, тому буде висвітлено лише десктопні програми, тобто ті, які встановлюватимуться на комп'ютер, а не додатки в мережі інтернет. Провівши дослідження, я можу сказати, що веб-сайти стають більш популярними при створення продукту, чим десктопні програми, адже вони забезпечують легкодоступність, швидкість та надійність.

Попри все, необхідно дослідити найпопулярніші аналоги, щоб зрозуміти правильність свого рішення та, можливо, взяти до уваги найкращі особливості проектів аби втілити їх і у своєму продукті.

					ІАЛЦ.467100.003 ПЗ	Арк
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

### 1.2.1. Менеджер Microsoft Outlook

Microsoft Outlook - це програма створена компанією Microsoft[1], і є наявною в пакеті Microsoft Office. В першу чергу дана програма є засобом електронної пошти, але також містить додатково контакти, нотатки, можливість перегляду веб-сторінок, журнал, планувальник завдань та найголовніше для даної роботи – календар. Для того, щоб об’єктивно оцінити даний аналог необхідно не лише провести дослідження над той частиною, яка призначена для вирішення проблеми бакалаврської роботи, а й програму в цілому. Головна привітальна сторінка Microsoft Outlook зображена на рис. 1.1

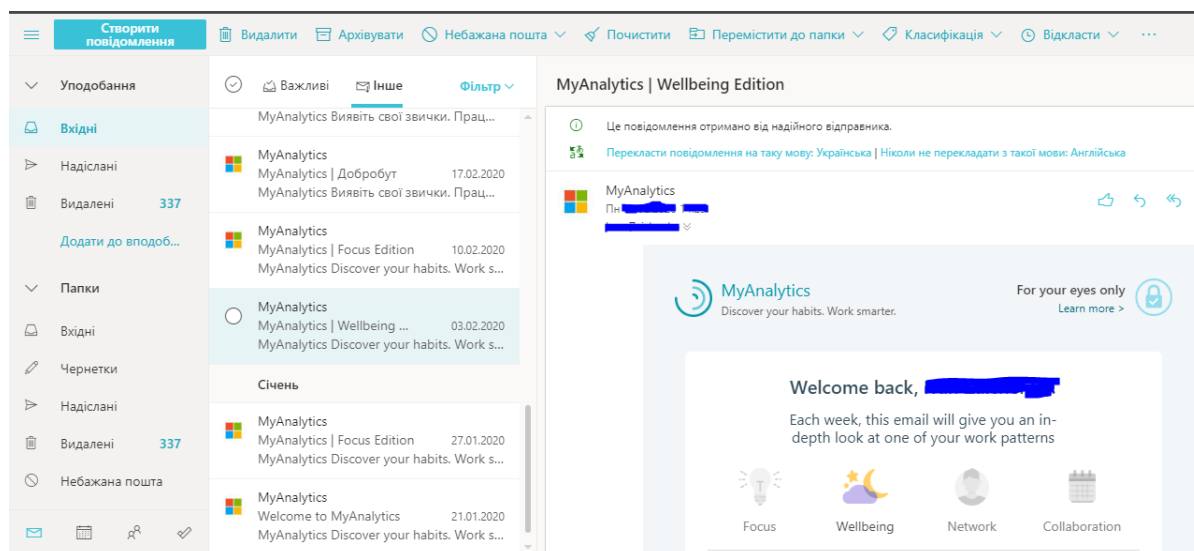


Рис. 1.1. Головна сторінка Microsoft Outlook

Даний продукт використовується зазвичай як окрема цілісна програма, але в деяких випадках є можливість працювати з Microsoft Exchange Server та Microsoft SharePoint Server. Це необхідно для корпорацій, які у складі містять величезну кількість працівників, щоб створити спільні календарі електронної пошти, доступні для всіх папки Exchange, SharePoint-списки та мітінги зустрічей для всіх команд. Були розроблені iOS та Android додатки для Microsoft Outlook, щоб поширити свою програму і на більшість мобільних платформ. Microsoft Visual Studio допомагає програмістам кодити власні

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		10

продукти, які тісно пов'язані з продуктами Office, і й зокрема Outlook. Також власникам Windows Phone присутня функція синхронізації даних Outlook з Outlook Mobile.

В третьому місяці теперішнього року Microsoft оновила ряд програмних особливостей, які повинні сподобатися користувачам додатку Teams. Найголовніша функція в оновленні - це взаємодія Outlook та Teams, який дозволяє користувачам копіювати ділову переписку електронною поштою Outlook в чат з даним користувачем в Teams. Це реалізувати не є проблемою, натиснувши відповідний значок. Очевидно, що дана функція є симетричною, що дозволяє скопіювати розмову в чаті Teams до електронної пошти в Outlook. Для того щоб люди могли без проблем коригувати свої повідомлення в компанії було добавлено спеціальні теги членам організації.

Даний додаток змінювався в кращу сторону з новішою версією. Він стає більш легшим у використанні звичайним користувачам, покращується графічний інтерфейс та легко інтегрується з все новими і новими іншими програмним забезпеченням. Провівши численні дослідження, я наведу ряд обґрунтованих причин чому Outlook є вибір мільйонів[2]:

- Чудова інтеграція з Microsoft Exchange

Outlook відмінно працює при присутності на вашому сервері Microsoft Exchange.

Користувачі нічого зайвого не повинні знати для свого підключення. Також це дозволяє легко перемикаати аккаунт між комп'ютерами.

- Outlook добре інтегрований з Active Directory.

Аутентифікація Active Directory необхідна користувачеві аби легко увійти на персональний комп'ютер, включити Outlook, а всі дані з Active Directory надаються на сервер Exchange і тому немає необхідності вводити свої пошту та пароль знову.

Microsoft Outlook сам має здатність записувати коректну електронну пошту, оскільки її адреса наявна в Active Directory, тому людині немає потреби вводити свої дані при автентифікації.

					ІАЛЦ.467100.003 ПЗ	Арк
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

- Outlook інтегрується з багатьма програмами

Аккаунт Outlook легко синхронізується між різними смартфонами та персональними комп'ютерами.

Окрім звичайних Android і iOS, такі мобільні платформи, як BlackBerry та Windows Mobile мають змогу інтегруватися з Outlook. Надзвичайно добре з цим справився додаток Skype, який є одним з першим, хто це зробив. На разі, важко знайти хоч якийсь додаток розташований на робочому столі, який не має змоги співпрацювати з Outlook. Також є багато модифікацій, які розширюють даний додаток, один з яскравих прикладів є Office Addins.com.

- Outlook дає змогу організувати свої активи

Деякі електронної пошти надають свою правила управління електронною поштою, але Outlook дійсно змінює правила гри. Звісно, ви можна сортувати свої повідомлення за допомогою фільтрів або надсилати, редагувати, пересилати повідомлення по певних критеріях. Проте у вас є додатково, можливість надсилати свої повідомлення поза електронні адреси своєї компанії.

Для того, щоб відслідковувати певну групу цікавих для вас повідомлень вам необхідно натиснути кнопку «Слідкувати», і також вам одразу ж прив'язуються нагадування під даним тегом. Також можна налаштувати «прапорець» повідомлень. Наприклад, для важливих повідомлень можна використовувати червоний прапорець, а для мінорних – зелений, що надає змогу найбільш ефективно та якісно відслідковувати свої повідомлення.

- Outlook використовує SharePoint для відтворення

Microsoft SharePoint - це платформа спільна для всіх користувачів, яка надає певні інструменти для керуванням веб-контентом та мережами. Але заздалегідь Microsoft розуміє, що деякі з людей не мають багато вільного часу та навіть навичок для переходу на даний сервер, для того аби подискутувати на різні теми або переглянути спільні файли.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		12

Для вирішення даної проблеми було створено новий функціонал, який успішно був впроваджений у систему. Необхідно натиснути на «SharePoint» при отриманні сповіщення або якщо змінився текст електронною поштою. Отже, Outlook та Exchange допомагає дискутувати на різних форумах та додавати матеріал у спільний для всіх документ.

- Інтеграція з Outlook Expedites Workflow

Повідомлення в Outlook не є єдиною функцією в електронній пошті. Організації також мають можливість створити онлайн-голосування для вирішення своїх конкретних задач. Отримувачі легко надсилають свої відповіді повідомленням, що значно заощаджує час всім, хто бере участь в обговоренні питань.

Варто відмітити, що є додаткова функція, яка регулює час відключення, тобто можна налаштувати точний час під час якого буде автоматичне виходження з серверу.

- Зручний інтерфейс користувача Outlook

Так як, багато користувачів використовує продукти Microsoft Office, то для них є знайомий графічний інтерфейс, що заощаджує час ознайомлення з програмним забезпеченням. Звісно, є багато нового функціоналу, яких варто дослідити, але база наявна для тих, хто працював з Microsoft Office. Також це є корисно для розробників, які мають бажання інтегрувати у своє програмне забезпечення Outlook для своїх цілей, адже це дозволяє зберегти час для програмування інших частин коду.

- Безпека

Попри багато скандальних новин щодо безпеки Outlook, програма має доволі захищену систем з своїми особливостями. Є багато стандартних функцій, які захищають користувача, деякі з них це – блокування зовнішніх веб-сайтів та непідтверджених зображень, виключає вкладені файли та не допускається виконуючі файли. Так, при необхідності, користувач може з легкістю обійти цю заборону, але за такі дії Outlook не бере на себе відповідальність.

					ІАЛЦ.467100.003 ПЗ	Арк
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

- Outlook пропонує пошту з єдиним доступом

Outlook надає змогу створювати не єдиний аккаунт, а декілька, що допомагає користувачам мати різну пошту для різних потреб та запитів. І тому продукт додає універсальний функціонал, який забезпечує юзерам симетричне управління повідомленнями для всіх підв'язаних аккаунтів до пристрою.

- Outlook надає інтегрований календар

І наостанок, найбільш важливий пункт, який і спричинив дослідження даного продукту – це можливість створення інтегрованого календаря, який містить в собі також контакти, списки завдань, події та інші корисні особливості. Користувач може назначити зустріч своїм друзям, колегам. Також є можливість забронювати певну подію на зручний для себе час, яка одразу буде висвітлена на календарі. Також при необхідності, є можливість редагування, видалення та створення нових подій. Приклад календарю Microsoft Outlook зображений на рис. 1.2

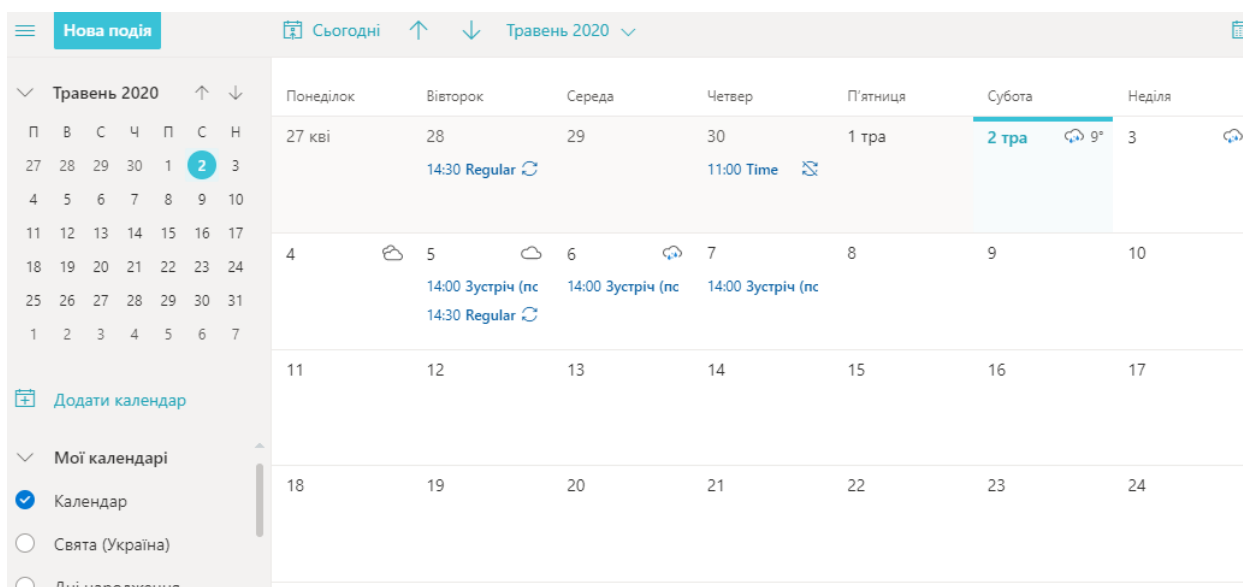


Рис. 1.2. Приклад календарю Microsoft Outlook

Microsoft Outlook надає можливість легко створювати подію, приклад цього зображений на рис. 1.3

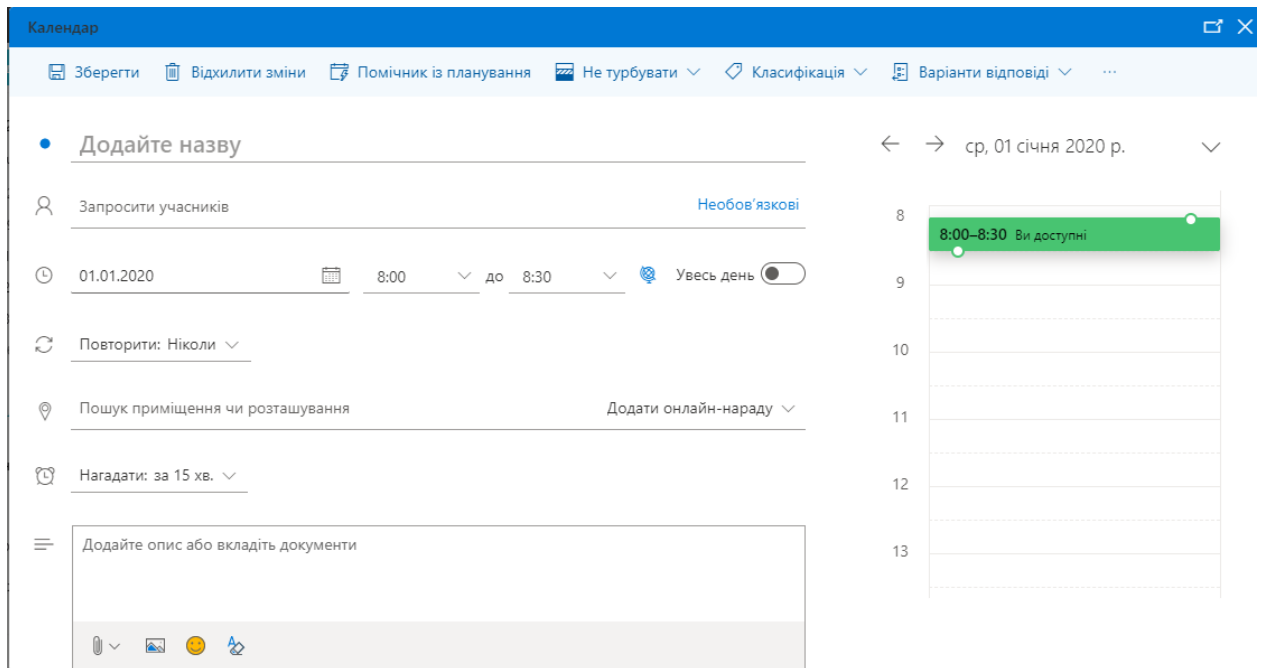


Рис. 1.3. Створення нової події в Microsoft Outlook

Microsoft Outlook є надійним та універсальним продуктом з великою кількістю функціоналу, здатний вирішити проблематику роботи, хоч і містить свої недоліки.

Роблячи висновки дослідження Microsoft Outlook, хочеться підкреслити наступні особливості продукту:

- Наявність календаря
- Можливість створення, зміну та видалення подій
- Наявність авторизації та аутентифікації
- Налаштування сповіщення для кожної події
- Налаштування фільтрів та папок
- Інтеграція з Microsoft продуктами

### 1.2.2. Сервіс Google Calendar

Google Календар - це система, яка допомагає керувати своїм часом створювати свій календар для власних потреб. Дане програмне забезпечення було розроблено ІТ-гігантом Google[3]. В середині 2006 року було випущено

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		15

десктопну версію продукту, а в Інтернеті дана програма з'явилася лише у 2009 році. Також в цьому році було розроблено та впроваджено додатки для платформ Android та iOS.

Даний продукт надає змогу юзерам створювати, змінювати, видаляти та проводити інші операції з своїми подіями. Налаштування сповіщень можна підв'язати до їх типу та відповідно часу. Цікавою особливістю є те, що окрім того, що можна до подій додавати контакти, також присутня вказання місцезнаходження майбутньої зустрічі. Для своїх потреб користувачі можуть налаштувати додаткові особливості Google календаря, такі як: підключення спеціальних календарів, додання дні народження контактів та автоматичне відсилка вітань та листівок з цього приводу, календарі для конкретних країн, для яких уже підв'язані усі святкові та вихідні дні. Було впроваджено машинне навчання, які надав Google календарю нові можливості. Одна з них – це інтеграція події з Gmail, де вся головна інформація про подію, така як назва, час, місце та інші з повідомлень юзера Gmail автоматично з'являється у календарі. Інша особливість – це нагадування, користувачам присилається аби вони пам'ятали про свою подію. Наступний пункт це - Розумні пропозиції, де веб-сайт надає варіанти при заповненні даних, таких як назва, контакти та місця при створенні подій. І, найбільш вагома - це цілі, де юзери вказують свою мету додавання події, а штучний інтелект автоматично надає користувачеві найбільш оптимальні часи для активності.

Було отримано різні відгуки на рахунок мобільних додатків Календаря Google. Попри те, що деякі критики висвітлювали, що дизайн є недоречним та займає дуже багато простору, але інші справедливо оцінили графіку за чистий та сміливий графічний інтерфейс. Те ж саме було з функцією «розумного підбору» деяким сподобалась дана ідея іншим ні. Проте, інтеграція між Календарем Google та Gmail було надано найвищі оцінки критиків та звичайних користувачів. Головний екран додатку зображений на рис 1.4 .

					ІАЛЦ.467100.003 ПЗ	Арк
						16
Змн.	Арк.	№ докум.	Підпис	Дата		

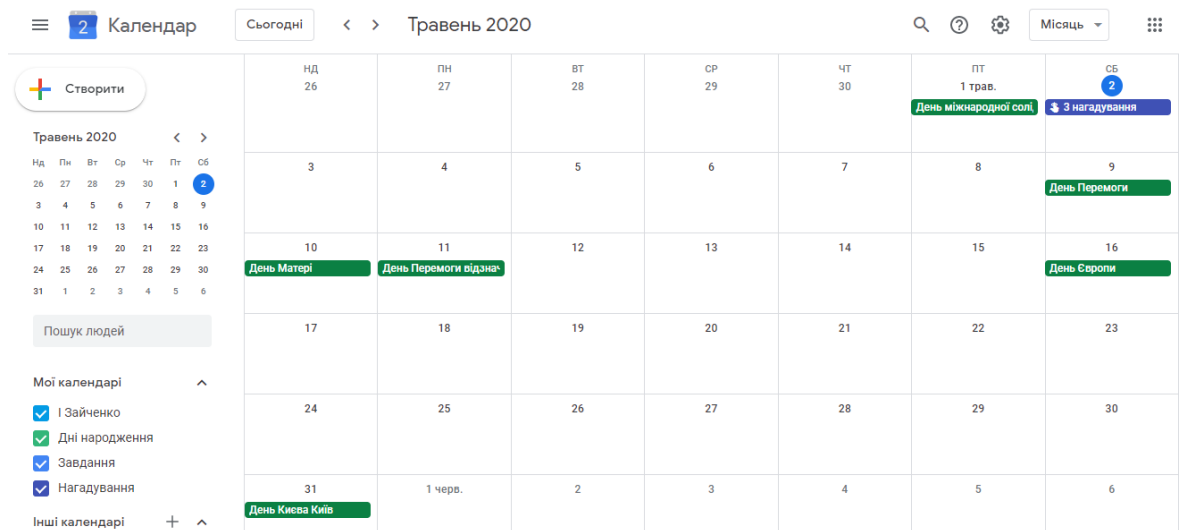


Рис 1.4. Головний екран Google Календаря

### Особливості Google Calendar:

- Найголовнішою особливістю Календаря Google є те, що користувачі мають право створювати, змінювати та видаляти події. Кожна з подій має час старту та час завершення події, також є подія, яка триває цілий день.
- Наявна особливість повторення події, користувачі можуть її запустити, щоб не заповнювати дані ще раз.
- Висвітлення подій в різному кольоровому фоні для пріоритизації завдань.
- Є різні форми перегляду календаря, такі як день, тиждень, місяць та рік.
- Наявність додання місцезнаходження для кожної з майбутньої події.
- Користувачі можуть налаштувати сповіщення на власну електронну пошту або на свій мобільний пристрій, щоб забезпечити дотримання завдань.
- Можливість запрошення до подій інших користувачів за посиланням. В свою чергу запрошені мають вибір чи погодитися

на дану пропозицію чи ні. У разі підтвердження, зустріч з'явиться у їхньому календарі відповідно.

- Наявна конфіденційна безпека, що надає користувачеві вибір видимості власних подій та календаря – приватна або публічна видимість.
- В календарі, хоч і підтримується місцевий час, але при необхідності користувач з легкістю може змінити часову зону для подій.
- Підключення особливих календарів, таких як календарі «дні народження» та «країни».
- Google Календар надає змогу юзеру синхронізувати події з різного програмного забезпечення, наприклад з Microsoft Outlook.

Приклад дня заповненого подіями зображено на рис. 1.5.

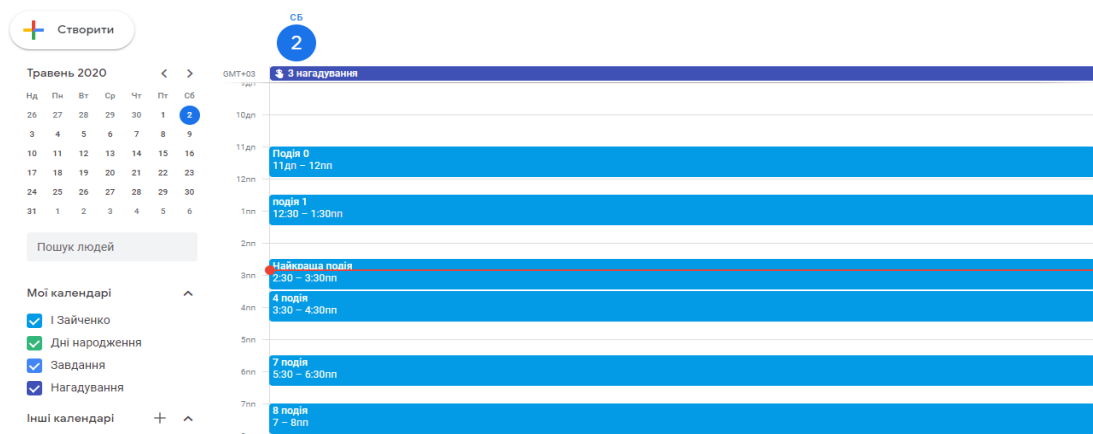


Рис. 1.5. Приклад роботи Google Календаря

Всі події зберігають в онлайн сховищі даних, тому додаток є наявний га будь-якому девайсі, якщо є підключення до мережі, а дані будуть синхронізованими та в цілісності, навіть якщо буде пошкоджено жорсткий диск. Продукт при необхідності має можливість синхронізації з Microsoft Outlook. Також є функція створення одночасно декілька календарів і надати контактам можливість перегляду до них, наприклад для спільної роботи в

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		18

організації. При необхідності додавання нових календарів, наприклад свят в країні, то це можна з легкістю інтегрувати за допомогою URL.

Ще десятиріччя тому було добавлено пошук заходів, яка надавала можливість перегляду інших календарів, але оновлення було відмінено через проблеми з інтерфейсом та підтримкою. Наразі, наявна аналогічна функція, переглянути календарі або додавання календарів за допомогою URL. Також можна це зробити надіславши відповідний запит іншому користувачеві з цікавим календарем. Для безпеки особистих даних, завдання неможливо передати іншим користувачам.

Розглянемо на прикладі створення нової події в Google календарі. Перш за все необхідно дати назву подій, окрім цього при необхідності можна вказати такі пункти, як контакти, місцезнаходження, конференція та опис. Окрім цього, є можливість налаштувати завдання та нагадування для події. Більш детально це зображено на рис. 1.6.

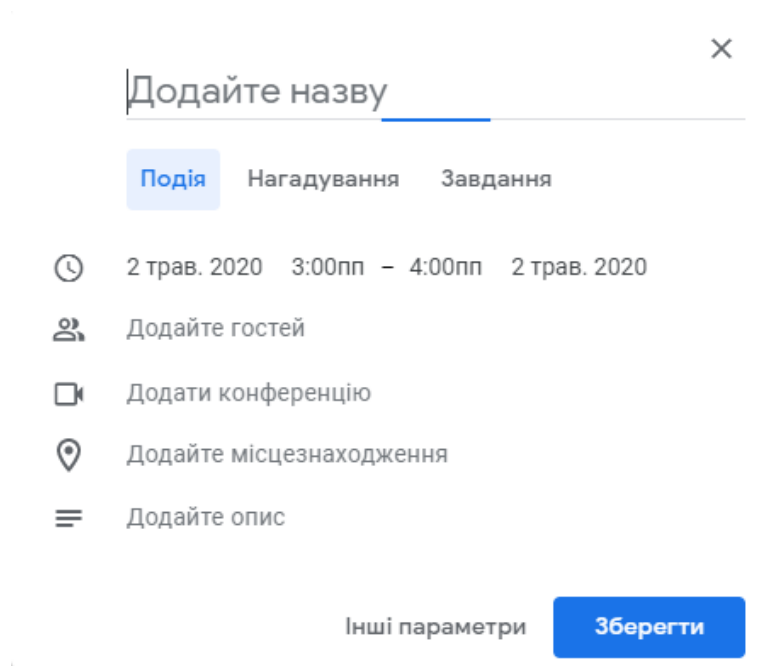


Рис. 1.6. Створення події в Google Календаря

Наявна синхронізація додатку з мобільними девайсами різних операційних систем, використовуючи гугл синхронізацію. Є можливість синхронізувати даний продукт з іншим програмним забезпеченням, що є

					ІАЛЦ.467100.003 ПЗ	Арк
						19
Змн.	Арк.	№ докум.	Підпис	Дата		

дуже привабливою особливістю для розробників[4]. Доступна функція нагадування за допомогою електронної пошти або навіть SMS-повідомленнями.

Google Календар є багатофункціональним та безпечним програмним забезпеченням. Даний додаток має здатність вирішити проблематику роботи, але він містить свої недоліки, які не дозволяють це зробити у повній мірі.

Дослідивши даний продукт, необхідно підкреслити наступні особливості продукту:

- Наявність календаря
- Додавання, редагування та видалення подій
- Авторизації та аутентифікації
- Сповіщення подій різними способами
- Інтеграція з Microsoft продуктами
- Повторення події
- Надання пріоритету подіям
- Різні форми перегляду
- Місцезнаходження події
- Додання інших користувачів
- Підключення спеціалізованих календарів

### 1.2.3. Jorte Calendar

Jorte – це програмне забезпечення, яке являє собою зручний та потужний інструмент для роботи з календарем та організацією подій[5]. Є можливість синхронізації, як з вбудованим календарем, так і з акаунтом Google Calendar. Головна сторінка Jorte зображено на рис. 1.7.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		20

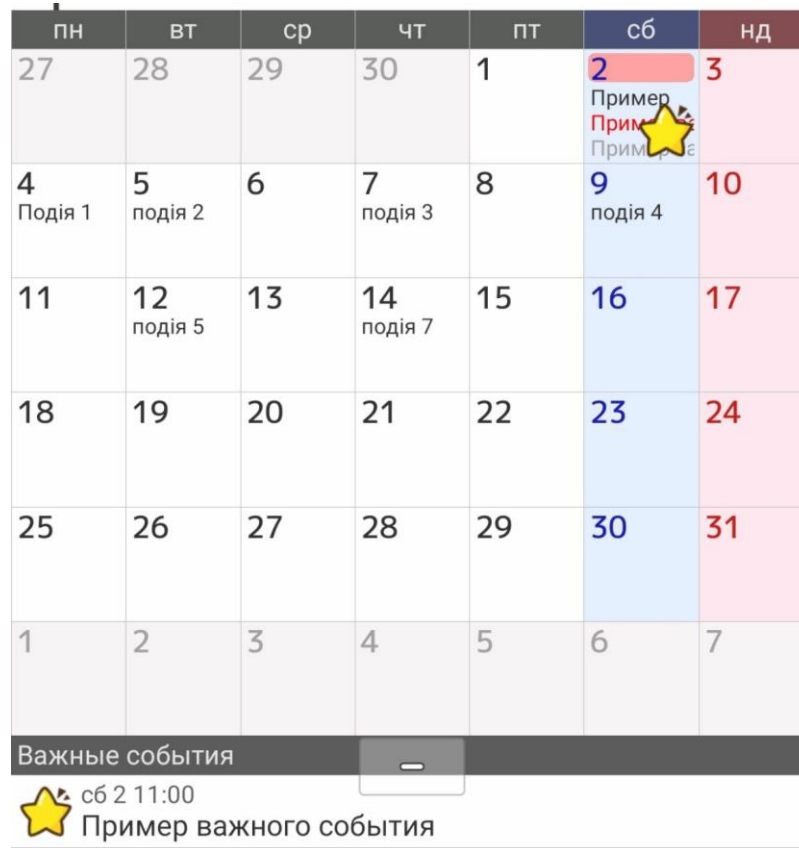


Рис. 1.7. Головний екран Jorte

Основний функціонал та особливості Jorte[6]:

- Календар на тиждень. Додаток забезпечує створенню тижневого плану і надасть завдання для досягнення цілей, що покращує керування розкладом.
- Можливість перегляду з колірним типом. Надає можливість пріоритизувати події та завдання. Для прикладу, найбільш важливі та значимі позначати червоним кольором.
- Синхронізація з існуючими Google Calendar
- Календар на місяць. Всі події будуть показані на в зручному для користувача вигляді, навіть при плануванні завдань на місяць вперед.
- Повторювальні події. Можливість при створенні події зробити її, як повторювальну в певний обраний проміжок часу. Для прикладу, кожен день, кожний тиждень, кожен місяць такожен

рік. Найпопулярніше використання даної функції – встановлення дня народження.

- Календар на день. Продукт надає змогу створити план на день, щоб чітко організувати день.
- Нагадування. Одна з найважливіших його особливостей – це чітке нагадування про заплановану подію.
- Сумісність з Android пристроями.
- Це безкоштовний продукт. Додаток знаходиться у вільному доступі в Google Play, будь-хто має право отримати продукт без плати.
- Статус завдань. Додаток має змогу управляти завданнями, надаючи їм певний стан виконання, для прикладу поточні, виконані та інші.
- Організовує робочі матеріали. Додаток керує документами систематизуючи їх по потрібним папкам і проектам
- Додаткові віджети. Можливість вибору віджету, для покращення ефективності використання додатку.
- Підтримка офіційних вихідних та святкових днів в залежності від обраної країни.
- Підтримка місцезнаходження. Дозволяє назначити місцезнаходження для кожної з подій при необхідності користувача.
- Зручний користувацький графічний інтерфейс
- Оптимізація для планшетів, яка відрізняється від мобільного додатку
- Підтримка великої кількості мов, що надає змогу використовувати програму більшій частині аудиторії
- В разі потреби графічний дизайн можливо змінити, покращуючи зовнішній вигляд, тему і кольори

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		22

- Програма має функцію яка перетворює голос в текст, що має багато своїх переваг
- Можливість прикріпити великий об'єм інформації до події
- Немає потреби для обов'язкової прив'язки аккаунта
- Підтримує імпорт та експорт csv файлів, які можна використати для синхронізації аккаунтів
- Синхронізація в хмарне сховище даних для синхронізації між різними пристроями користувача
- Створює віджет на головний екран
- Програма має невеликий обсяг пам'яті, що дозволяє зберігати на пристрої інші дані та програми.
- Синхронізація з Microsoft office та другими програмами, що дозволяє програмісту використовувати дану програму у своїх некомерційних проектах

Естетично та зручно для користувача було розроблено додавання нової події, вказавши всі необхідні поля та функції наглядно. Більш детально зображено на рис. 1.8.

The screenshot shows the 'Add Task' screen in the Jorte app. At the top, it displays the date 'сб, 2 трав. 2020' and the title 'задача' (task) under the 'Дневник' (calendar) tab. The form includes the following elements:

- Значки** (Tags): A search bar with the text 'Наимен.' and a menu icon.
- Когда** (When): Two date pickers separated by a hyphen, with a 'Дата' (Date) button.
- Время** (Time): Two checkboxes labeled 'Весь день' (All day) and 'Дн. отдыха' (Day off), both of which are checked.
- Цвет** (Color): A color selection button.
- Важно** (Important): A checked checkbox.
- Завершено** (Completed): A checked checkbox.
- Где** (Where): An empty text input field.
- Примеч.** (Remarks): An empty text input field.
- Отсчет до** (Countdown): A checked checkbox.
- Повторить** (Repeat): A button labeled 'Единичное событие' (One-time event).
- Напомнит** (Remind): A dropdown menu set to 'За 10 мин до' (10 minutes before), with a minus sign button to the right.
- Добавить напоминание** (Add reminder): A plus sign button.

Рис. 1.8. Додавання події в Jorte

Jorte, окрім того, що може додати подію на цей час, має можливість додати «щоденник» до нього – це функція, яка дозволяє прикріпити до існуючої події текст, фото, тег, шаблон та інші корисні особливості.

Jorte – це продукт, який містить функціональність для вирішення проблематики роботи, але наявні недоліки, що не дозволяють це зробити.

Оцінюючи даний продукт, можна виділити наступні особливості:

- Присутність календаря
- Додавання, редагування та видалення подій
- Присутність сповіщення
- Інтеграція з різними програмними забезпеченнями
- Повторизація подій
- Перегляд у формі дня, тижня, місяця, року
- Присутність місцезнаходження
- Додавання контактів
- Додаткові календарі

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		24

## ВИСНОВКИ ДО РОЗДІЛУ 1

У першому розділі були розглянуті та досліджені аналогічні веб-додатки та мобільні програми, які пов'язані з можливістю регулювання подій, оскільки ще не представлена система, яка в повному обсязі вирішує проблематику регулювання студентського дозвілля в гуртожитку. Проведені дослідження були систематизовані з ціллю визначення вимог до розроблювального додатку та використання їх в майбутні розробці. Під час оцінки аналогів були зроблені висновки про їх особливості, недоліки та переваги, які будуть наведені в таблиці 1.1.

Таблиця 1.1 Аналіз аналогів додатку

Наявність функціоналу	Microsoft Outlook	Google Calendar	Jorte
Календар	+	+	+
Додавання, редагування, видалення подій	+	+	+
Надійність та безпека	+	+	+
Сповіщення	+	+	+
Зручний графічний інтерфейс	-	+	+
Веб-додаток	+	+	-
Імпорт та експорт подій	-	+	+
Спільний доступ до подій	+	+	+
Створення додаткового календаря при необхідності	-	+	-
Інтеграція з студентським життям та дозвіллям	-	-	-
Контроль для адміністрації	-	-	-

Аналізуючи таблицю вище, можна зробити висновок, попри різноманіття функціоналу програм-аналогів, вони не полишені своїх недоліків. Незважаючи на те, що найкращим варіантом виявився сервіс Google Calendar, додатку не вистачає функціональності аби коректно та ефективно регулювати систему студентського дозвілля в гуртожитку.

Отже, при створенні веб-додатку необхідно дотримуватися найкращих практик досліджених аналогів, при тому не забуваючи про розробку ряду особливостей, які присутні лише в даній бакалаврській роботі.

					ІАЛЦ.467100.003 ПЗ	Арк
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		26

## РОЗДІЛ 2

# АНАЛІЗ СУЧАСНИХ ТЕХНОЛОГІЙ РОЗРОБКИ ДЛЯ РІШЕННЯ ПРОБЛЕМИ

### 2.1.Завдання та технології

Наразі існує великий спектр вибору мов програмування та технологій для розроблення свого програмного забезпечення. У сучасному ІТ для веб-додатків домінує клієнт-серверна архітектура, яка зображена на рис. 2.1.

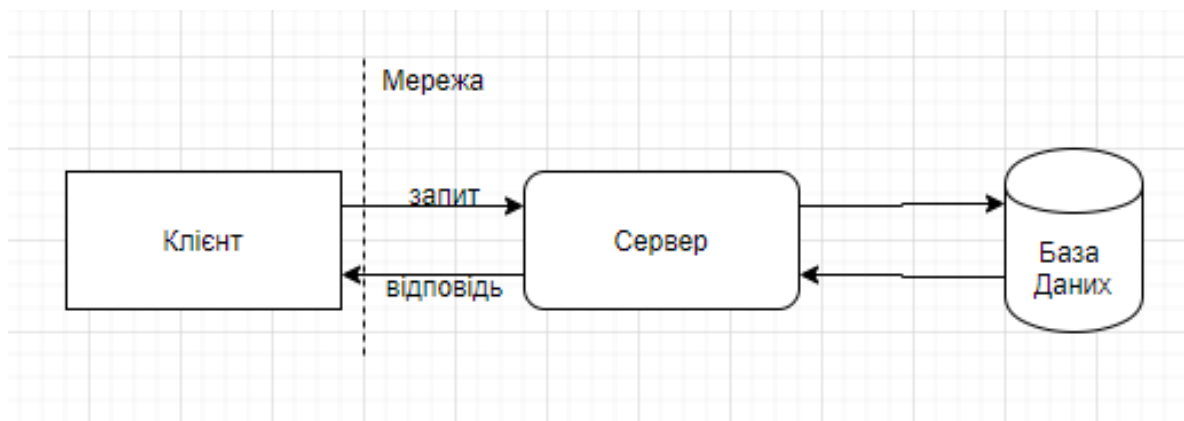


Рис. 2.1. Клієнт-серверна архітектура

Роблячи висновки з зображення вище, можна розділити веб-додаток на три основні частини:

- Клієнт – це частина архітектури, яка за допомогою протоколу і мережі надсилає запит серверу, очікуючи від нього відповідь.
- Сервер - це частина архітектури, яка приймає запит від клієнтської частини і за певною бізнес-логікою оброблює його та надає клієнту повноцінну відповідь за допомогою мережі.
- База даних – це сукупність елементів, які згруповані логічним чином у певну структуру, найчастіше схему, таблицю.

Буде досліджено сучасні та актуальні технології, щоб забезпечити веб-додаток якісною та гнучкою клієнт-сервісною архітектурою. Отже, в наступному розділі будуть висвітлені та оцінені дані технології, які будуть обрані та використані у розробці програмного забезпечення.

- Серверна частина : Java, C#, Python

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		27

- Клієнтська частина: JS, AngularJS, Vue.js
- База даних: PostgreSQL, MySQL, SQLite

За результатами досліджень буде обраний набір технологій для подальшого проектування та реалізації планувальника задач.

## 2.2. Вибір мови програмування

### 2.2.1. Мова програмування Java

Мова програмування Java – це об’єктно-орієнтована мова для розроблення програмного забезпечення загального призначення, яка реалізована за допомогою класів та об’єктів[7]. Дана мова програмування була створена, щоб позбутися глибокої залежності коду. Для того, щоб код Java виконувався на всіх машинах, необов’язково його перекомпілювати. Для того, щоб не залежати від особливості архітектури персонального комп’ютера, програмне забезпечення написане на мові Java, компілюється до байт коду і має можливість виконуватися на будь-якій Java Virtual Machine. Беручи до уваги дослідження системи контролю версії Github, то можна стверджувати, що мова програмування Java, є однією з найпопулярніших, які використовуються, наразі, а й в особливості для створення, розробки та підтримці веб-додатків. Підтвердженням цього є більше 10 мільйонів людей, які використовували Java для написання свого програмного забезпечення кожного року, за даними вже згаданого Github.

Розглянемо основні переваги мови програмування Java:

- Об’єктно-орієнтованість. Головною частиною в Java є об’єкт, майже все в даній технології є ним. Це дозволяє легко розширювати моделі та додатки в цілому.
- Платформонезалежність. Задля того, щоб Java не залежала від обраної платформи, вона була розроблена за допомогою байт-коду. Компіляція програм відбувається JVM та має можливість поширюватися мережею Інтернет.

					ІАЛЦ.467100.003 ПЗ	Арк
						28
Змн.	Арк.	№ докум.	Підпис	Дата		

- Легко розуміти розробникам, які мали досвід з об'єктно-орієнтованими мовами програмуваннями.
- Безпечність. Перевірка автентифікації відбувається за допомогою на спеціальними кодом шифрування з відкритим ключом.
- Є незалежним від архітектури комп'ютера на якому відбувається компіляція програми.
- Перевірка помилок на етапі компіляції, що заощаджує час розробки.
- Паралельність. Налаштування програмного коду задля забезпечення виконання програми одночасно за допомогою кількох ядер персонального комп'ютера, що надає змогу створювати швидковиконувальні, конкурентоспроможні додатки.
- Інтерпретованість. Оскільки байт код робить перетворення одразу в JVM, то процес виконання програми стає швидшим та ніде не зберігає непотрібні локальні змінні.
- Наявність великого спектру додаткових фреймворків, які полегшують життя розробникам. Приклад одного з найпопулярніших фреймворків Spring зображено на рис 2.2.

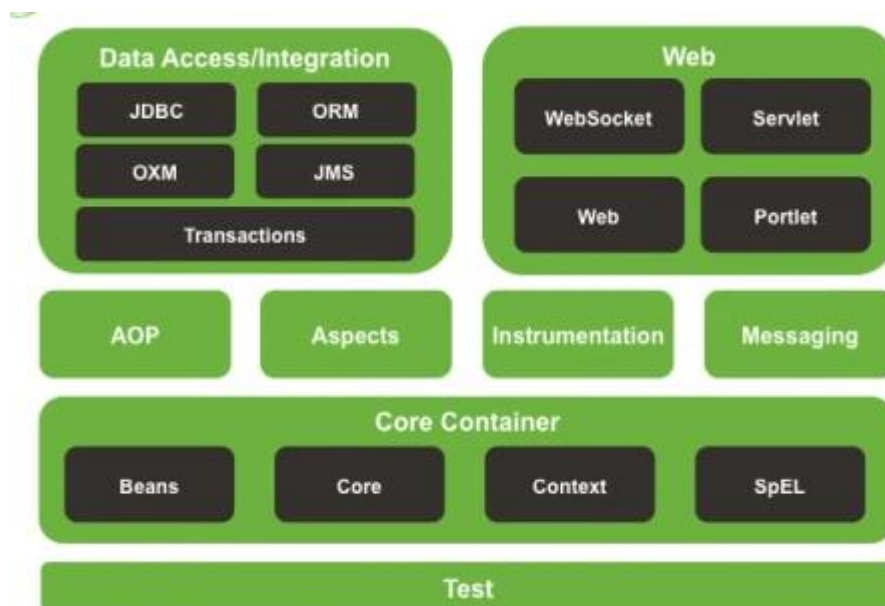


Рис. 2.2. Модульна структура Spring[8]

– Високопродуктивність. Високу продуктивність забезпечує компілятор Just In Time, який широко використовується у програмному забезпеченні.

– Динамічність. Мова програмування Java є легкою до змін та адаптацій, що надає гнучкість у розробці додатків.

Розглянемо основні недоліки мови програмування Java:

– Доступ до полів і методів відбувається повільніше чим в інших мовах програмування

– Відчувається брак швидкодії порівняно з іншими мовами програмування, таких як C, C++.

– У разі масштабного продукту відчувається заплутана та важка у розумінні ієрархія.

### 2.2.2. Мова програмування Python

Python (пайтон) – ОО мова програмування, в якій наявна строга динамічна типізація. Окрім того, важливою особливістю є те, що наявна підтримка кількох парадигм програмування, таких як об'єктно-орієнтованість, процедурність, функціональність та аспектно-орієнтованість.

Явне використання у таких сферах, як[9]:

– реалізація десктопних програм з графічним інтерфейсом

– реалізація веб-додатків;

– реалізація програмного забезпечення задач бізнесу;

– розробка операційних систем;

– прототипування.

У Python вбудована стандартна бібліотека, яка має величезний вміст корисних функцій, які реалізовані раніше. Однією з найпопулярніших у розробників особливістю мови є те, що Python легкий для розуміння новачкам, а в цілому програмний код є чистий та читабельний.

					ІАЛЦ.467100.003 ПЗ	Арк
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

Python доволі активно розвивається і наразі, що доводиться постійними та актуальними оновленнями збоку розробників. Додаються та редагуються вбудовані функції, але сутність мови не змінюється. Основними особливостями, які широко використовуються користувачами даної мови є:

- динамічна типізація
- динамічна перевірка типів
- забезпечення паралельності виконання
- повна інтроспекція
- автоматичне управління пам'ятю
- обробка виключень

Не тільки класи, а також методи, модулі та функції в Python є об'єктами. Окрім примітивних типів, звичайно наявні і вбудовані колекції, такі як словники, множини, списки, кортежі та інші.

Є підтримка, як одиночного, так і множинного наслідування. Присутня можливість наслідуватися від типів, як вбудованих, так і розширення. Одним з методів, щоб добитися елегантної реалізації складних шаблонів проектування є метапрограмування. Найпоширеніші фреймворки Python, мають вбудоване метапрограмування гнучкості системи та забезпечення неповторюваності програмного коду. Найпопулярніші фреймворки зображені на рис 2.3.



Рис. 2.3. Найпопулярніші фреймворки Python

Python має багато сторонніх модулів:

- для розробки веб-додатків
- для роботи з БД (зокрема з PostgreSQL, MySQL та інші);
- при використанні роботи з математичними та нуковими формулами та розрахунками (машинне навчання, аналіз)
- при розробці кросплатформиних десктопних програм, ігор з графічним інтерфейсом

Слід і звернути увагу на недоліки Python:

- повільна швидкодія з іншими мовами програмування
- недосконала реалізація паралельності виконання програм
- несумісність роботи строк з Юнікодом
- ненайкраще оновлення в третій версії Python

### 2.2.3. Мова програмування C#

C# – це об'єктно-орієнтована мова програмування, яка має подібний синтаксис до мов C ++ та Java. Андерс Хейлсберг створив дану мову програмування у 2000 році.

C# включає в себе події, типізацію статичну, поліморфізм делегати, переваження операторів, атрибути, тощо. Окрім об'єктно-орієнтованої парадигми C# також використовує процедурне, функціональне, рефлексивне, узагальнене та інші.

Найбільш переконливі переваги C#:

- присутній величезний список бібліотек та шаблонів;
- легкозрозумілий та читабельний код
- присутність фреймворку ASP.NET, який надає розробникам ефективну веб-розробку. ASP.NET технологією для створення веб-додатків, який випущений компанією Microsoft. ASP.NET - це найбільший фреймворк для створення веб-додатків порівняно з іншими веб-фреймворки. Присутня підтримка MVC, яка надає ASP.NET ще більше популярності.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		32

Додаткові модулі:

- Windows Presentation Foundation -це розширений API-інтерфейс для розробки десктопних додатків з графічним інтерфейсом, має привабливий дизайн і інтерактивність. Якщо порівнювати з технологією Windows Forms, то WPF забезпечує сучасну та оновлену модель для розробки веб-додатків.
- Silverlight – це технологія, яка надає засоби для розробки масштабного програмного забезпечення, для виконання в браузерах під різними операційними системами. Дані додатки Silverlight оброблюється браузерами різних типів. При запиті користувача на веб-сторінці, який має елементи Silverlight, браузер компілює і виконує програмний код Silverlight і надає елементи, які висвітлюються в задану місце на HTML-сторінці.
- Очікувано що, мова програмування C# має можливість, які надають змогу працювати, як і з локальними, так і з віддаленими БД використовуючи ADO.NET та Entity Framework.

Це не всі основні фреймворки C#, оскільки це одна з найпоширеніших та найсучасніших мов програмування, тому суттєві та актуальні оновлення випускаються розробниками часто та якісно. І оскільки, велика частина програмістів використовує C# для розробки, то існує і велика кількість фреймворків, які зображені на рис 2.4.



Рис. 2.4. Основні фреймворки C#[10]

					ІАЛЦ.467100.003 ПЗ	Арк
						33
Змн.	Арк.	№ докум.	Підпис	Дата		

Також попри величезний функціонал та низку переваг, дана мова програмування має і свої недоліки, які є дуже вагомими особливо для тих, хто тільки почав займатися програмуванням і хоче розвиватися в даному напрямку:

- Основним недоліком C# та ASP.NET полягає в тому, що дані технології використовувалися лише пристроями з операційною системою Windows. На разі, IT-організація Microsoft надала доступ до свого програмного коду .NET, а також реалізовано кросплатформеність, що забезпечує величезну гнучкість розробникам та замовникам у виборі технології для свого продукту.
- Проте програмісти повинні отримувати ліцензію для свого програмного забезпечення з метою створення додатків. Зрозуміло те, що дана ліцензія є коштовна для всіх операційних систем, на відмінну від Windows, яку все ж необхідно придбати.

## 2.3. Вибір системи управління бази даних

### 2.3.1. СУБД PostgreSQL

PostgreSQL є вільною і відкритою системою, мета якого полягає в управлінні, в основному, реляційних БД, що надає розширюваності та технічних стандартів для відповідних систем. Дана система була створена, щоб працювати з навантаженнями різної складності, до прикладу, від невеликих серверів аж до сховищ даних та веб-додатків з багатопотоковим напливом даних від користувачів. Зрозуміло, що дана система використовує транзакції, які підлягають принципам ACID: атомарність, довговічності, консистенції та ізоляції. Забезпечується підтримка тригерів, збережених процедур, автооновлюваних переглядів, матеріалізованих переглядів та зовнішніх ключів. На даний момент, PostgreSQL – це заслуга праці не лише компанії-розробниці, а також користувачів і організацій, які працюють з даною СУБД та вчасно реагують на актуальні оновлення Мова С – це

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		34

головна мова програмування для серверу PostgreSQL. Найпоширеніша функція – це розповсюдження набору текстових документів із головним програмним кодом.

Postgres був створений з метою забезпечення мінімального додавання кількості набору функцій, які необхідні за для того, щоб підтримувати усі наявні типи даних[10]. Бета-версія, створена та висвітлена на огляд критикам і користувачам була аж у 1988 році. Через шість років після цього було замінено на SQL інтерпретатор мови запитів. Маючи в наявності ліцензію PostgreSQL, можна отримувати всі актуальні оновлення, які були випущені, та випускаються і по цей день. Програмний код змінюється не тільки від оригінальних розробників, а від усіх програмістів світу, адже даний продукт знаходиться у відкритому для всіх репозиторії системи контролю версій. Головну сторінку PostgreSQL зображено на рис. 2.5.

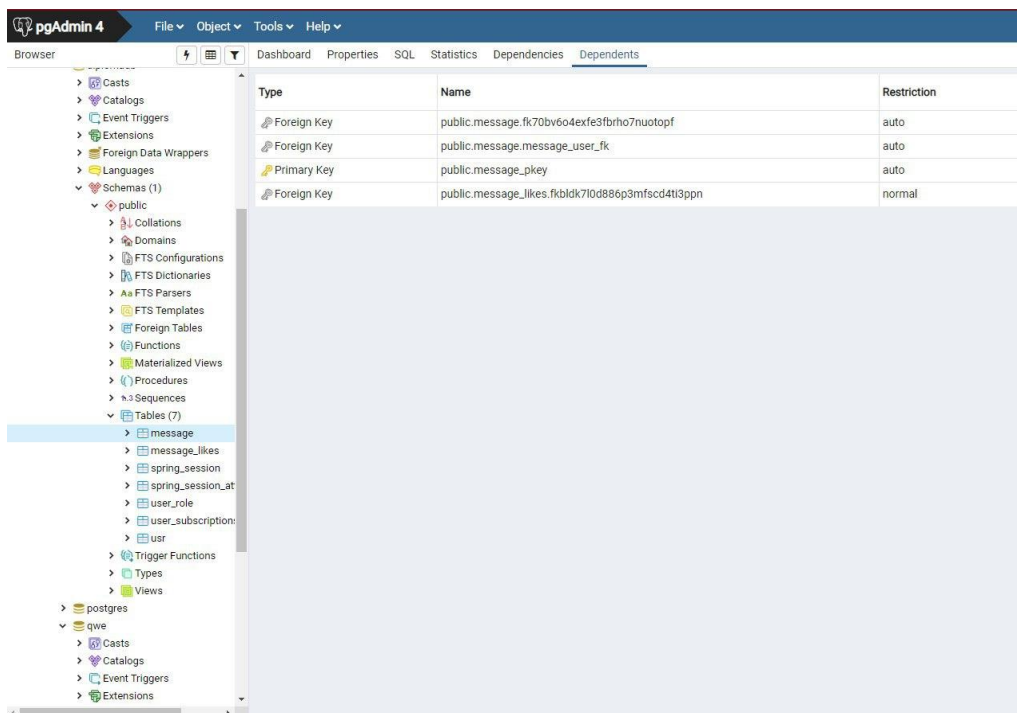


Рис. 2.5. Головна сторінка PostgreSQL

Основними переваги PostgreSQL, на яких варто звернути увагу це:

- Надійність. Забезпечення стабільної роботи не протягом кількох років, а навіть десятиліттями, попри високу активність та навантаження збоку користувачів.

- Часте оновлення та змога принести свій вклад у розробку даної СУБД. Головний програмний код є доступним абсолютно для всіх безкоштовно. Навіть при необхідності зміни заводських налаштувань, це можна зробити у легкий та ефективний спосіб. Розробка даної СУБД також присутня за допомогою організацій і юзерів, які використовують PostgreSQL у своїх цілях .
- Гнучкість. Оскільки, PostgreSQL – це об’єктно-реляційна структури даних СУБД.
- Цілісність транзакцій, адже, як було вказано вище дана система відповідає всім принципам ACID.
- Використання для керування та адміністрування графічного користувацького інтерфейсу. На даний момент, є можливість вибору GUI для цього, адже на світовому ринку є наявність багатьох зручних графічних інструментів, які доступні для користувачів PostgreSQL, як вбудовані програми, так і зовнішнє програмне забезпечення.
- Підтримка величезної кількості типів даних. Дана система має можливість працювати з такими даними, як: MAC адреси, XML дані, адреси вузлів, мережеві адреси, знімок ідентифікатора транзакцій, числові типи, серед яких типи з плаваючою точкою, бінарні, геометричні, грошові, а також за необхідності є можливість додати свій новий тип даних, для цього потрібно використати команду CREATE TYPE.

Попри величезну кількість переваг, необхідно пам’ятати і про недоліки PostgreSQL:

- Повільніший чим деякі СУБД, наприклад MySQL при складних обчислювальних операціях.
- Не забезпечує роботу з стандартом ANSI SQL 92.
- Виникають труднощі з вибором хостингу.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		36

### 2.3.2. СУБД MySQL

MySQL - це реляційна СУБД з відкритим головним програмним кодом кодом. Є дві версії ліцензії даної системи: перша – це безкоштовне і відкрите програмне забезпечення від компанії General Public License GNU[11]; друга – це система, яка в кошовній версії, яка надається для власних ліцензій.

Створення даної СУБД розвивалося ще у далекому 1994 році. Спершу система розроблювалася для особистого користування. Була створена на основі повільної та негнучкої системи mSQL використовуючи низькорівневу мову ISAM. Творці розробили новий покращений інтерфейс мови SQL, при цьому вони зберегли API, який використовувався у вищезгаданій mSQL. Оскільки, зберігалася API mSQL, то деякі з розробників втілили у життя змогу отримання ліцензії mSQL antecedent при цьому використовуючи MySQL, яка була у відкритому доступі. Компанія Sun Microsystems володіла даною СУБД, це ж та сама організація, яка мала у власності мову програмування Java.

Дана система є частиною технології LAMP, яка складається з таких компонентів, як:

- Операційна система Linux.
- Сервер Apache.
- База даних MySQL.
- Мова програмування одна з Perl, PHP, Python.

Drupal, Joomla, phpBB і WordPress – це маленька низка тих веб-додатків, які використовують MySQL. Дана система присутня у розробці таких відомих веб-сайтів як:

- YouTube
- Facebook
- Flickr
- Amazon
- Twitter

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		37

При виборі даної системи слід знати про такі переваги:

- **Безпечність.** MySQL відома тим, що це найбільш безпечна СУБД у світі. Доказом те, що її використовують відомі веб-додатки : WordPress, Twitter, Joomla, Facebook та інші. Коректна підтримка транзакцій, які присутні у останній версії MySQL, є цікавою для бізнесу, в першу чергу тих, хто займається електронною комерцією і мають доволі часті грошові перекази.
- **Масштабованість.** MySQL надає широку масштабованість задля того, щоб полегшити керування власними додатками, які мають невеликий розмір, при цьому знаходячись у великих сховищах даних, які займають терабайти.
- **Використання зручного користувачького інтерфейсу MySQL WorkBench,** який зображено на рис 2.6.

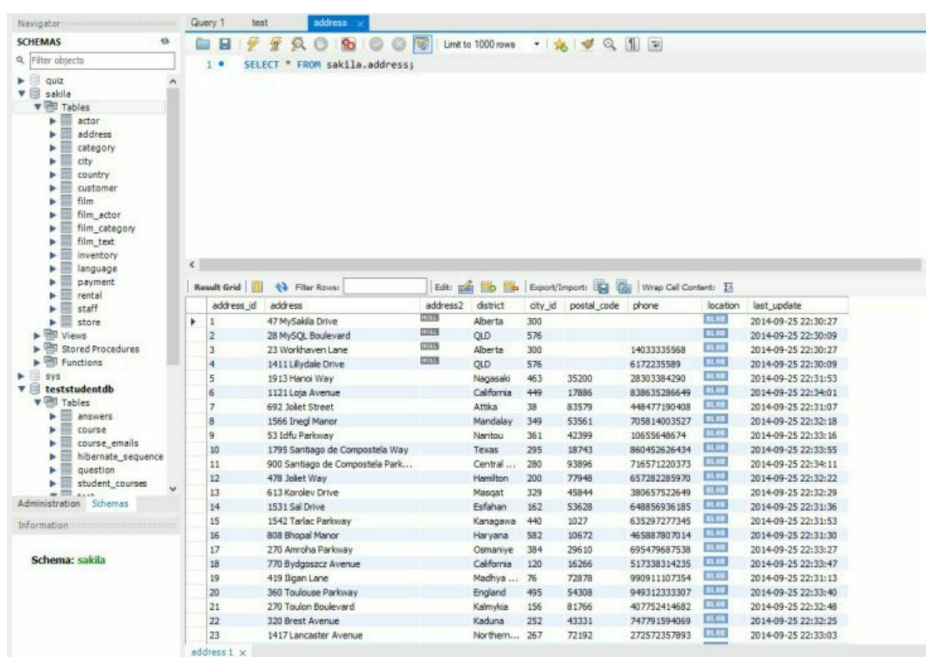


Рис. 2.6. Приклад роботи з MySQL WorkBench

- **Гнучкість,** що надає перевагу користувачам даної СУБД.
- **Ефективність.** Дана система містить виняткову систему зберігання даних, що надає серверам MySQL якісною роботи. СУБД створена для вирішення проблем для найбільш вимогливих додатків також, надаючи

гарну швидкість, унікальні кеші пам'яті та повнотекстові індекси, щоб підвищити продуктивність.

- MySQL безперебійна цілодобова робота системи надаючи спектр рішень для вирішення різноманітних проблем, серед яких кластерні сервери та спеціальні конфігурації.
- Коректна підтримка транзакцій. MySQL є найнадійнішим механізмом для транзакцій у БД, які є наявні на світовому ринку.
- Ефективне та легке управління процесом. При встановленні технології протягом 30 хвилин, MySQL надає зручність користування майже одразу. Незалежно від вашої платформи, MySQL – це одна з найкращих на ринку рішень з спектром функцій, які автоматизовані, від конфігурації, до керування баз даних.
- Зменшена ціна вартості ліцензій.
- Підтримка розробниками-професіоналами у разі відмов та неочікуваних помилок, якщо присутня ліцензія

Також наявний ряд недоліків MySQL:

- Нестабільність. Порівняно з іншими базами даних MySQL менш надійна. Проблеми стабільності відслідковуються з виконанням певних функцій, таких як посилення, транзакції та аудит.
- Деколи відмова з роботою при обчисленні великої кількості операцій в один момент часу.
- Oracle є власником MySQL, який не займається оновленням СУБД. Компанія немає бажання приймати нові версії продукту від користувачів. Немає взаємодії між розробниками та Oracle.
- Менша функціональність даної СУБД.
- Дана БД не є цілком сумісною з мовою SQL та обмежена в областях, які пов'язані з відмовостійкістю, зберіганням даних і продуктивну діагностику.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		39

### 2.3.3. СУБД SQLite

SQLite – це реляційна СУБД, яка наявна в бібліотеці програмування C. Характеризується відкритістю, наявністю сервера та багатофункціональністю, також вільно працює з веб-програмами. Створена та випущена у 2000 році. Ідея реалізації SQLite в тому, що надати дозвіл програмі працювати не маючи встановленого СУБД.

Відмінність від інших СУБД полягає в тому, що SQLite – це не механізм БД клієнт-сервер[12]. Це схоже на прикріплення системи в кінцеву програму. Однозначно, SQLite підтримує принципи ACID і реалізує більшість SQL. Однак, БД має синтаксис SQL, який слабо і динамічно типізований, а отже з цього висновок, що немає ніякої гарантії забезпечення цілісності домену. SQLite в даний момент є одним з популярних виборів програмного забезпечення вбудованої бази даних і для локального і для клієнтського використання в програмах, для прикладу, веб-браузери. Можливо, це найбільший розгорнутий двигун бази даних, на даний момент, він міститься у браузерах, у багатьох операційних системах, та наприклад, мобільних телефонах. SQLite прив'язаний до декількох мов програмування.

Розглянемо основні переваги SQLite:

- Відсутня конфігурація. SQLite немає необхідності встановлювати перед використанням БД, тому що даної процедури взагалі не існує. Відсутній серверний процес, який необхідно включати, зупиняти та конфігурувати.
- Адміну немає потреби створювати нову базу даних або надавати користувачам права. SQLite не потребує документів конфігурації. Немає необхідності, казати системі, що СУБД працює.
- Відсутність сервера, тому що двигун БД був створений як окремий сервер. Немає в необхідності проміжного процесу сервера. Всі наявні програми, маючи доступ до жорсткого диска, має право використовувати дану СУБД.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		40

- Зручний та інтуїтивно-зрозумілий користувацький інтерфейс, який зображено на рис 2.7.

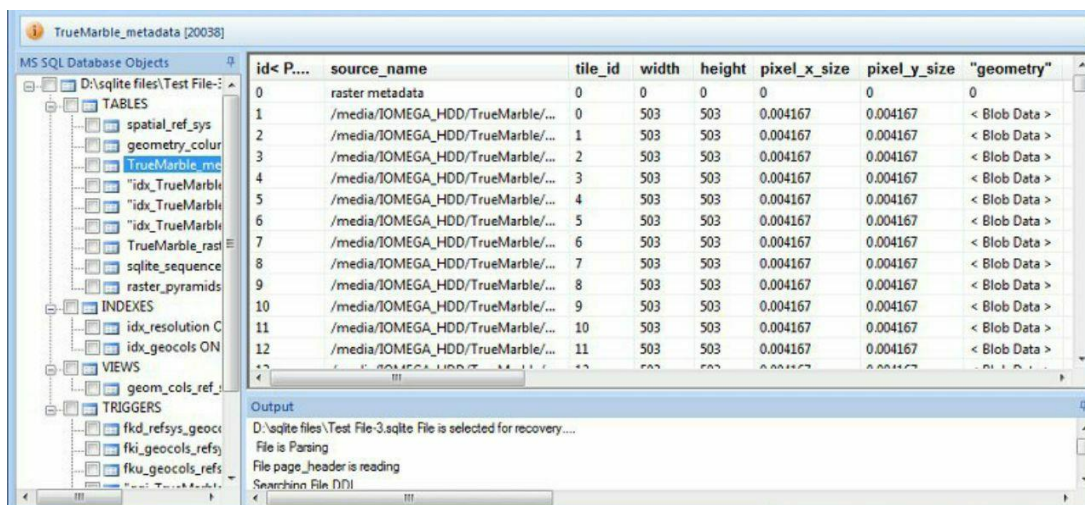


Рис. 2.7. Приклад роботи з SQLite

- Однофайлова БД. SQLite є звичайним файлом жорсткого диска, що може розташуватися в будь-якій частині вільного простору персонального комп'ютера. SQLite може також читати дані з файлу на жорсткому диску, адже він має здатність читати дані з бази даних, яка є файлом. Файли БД з легкістю копіюються на флешку USB та надсилаються поштою для спільного користування користувачами. Двигуни БД зберігають дані як колекцію кількох файлів. Файли стандартно містяться в розташуванні, доступ до якого має лише движок БД. Дані стають безпечніші, проте ускладнюється доступ для користувачів.
- Непотрібно відновлювати систему після аварії або збою.
- Кросплатформний документ БД. Файл, отриманий на одному комп'ютері, з легкістю копіюється та використовується на іншому комп'ютері і навіть відмінною архітектурою.
- Під час апгрейду до новішої версії додатку, SQLite не вимагає відновлення БД
- Відкритий доступ до програмного коду. За необхідності можна відшукати безкоштовно програмний код.

- SQLite одна найвикористовуваніших систем управління баз даних у сучасному світі. Вона використовується з всіма операційними системами і деякі використовують його, як стандарт.

Слід розглянути і недоліки SQLite:

- SQLite на даний момент не використовується для обробки звернень HTTP-протоколу з високим трафіком мережі.
- Обмеження розміру бази даних до двох гігабайт.

## 2.4. Вибір технології для клієнтської частини

### 2.4.1. Мова програмування JavaScript

JavaScript – це прототипно-орієнтована мова програмування, яка створена за принципом мови ECMAScript. Була розроблена в 1995 році і з того часу оновлювалася, і отримала теперішній вигляд. Широко JavaScript використовується у розробці веб-додатків та веб-браузерах для надання їм естетичного вигляду та привабливого дизайну.

Характерною рисою даної мови є вплив інших мов програмування, таких як Java, Python для надання комфортної роботи з JavaScript та легкого освоєння початківцями-програмістами та навіть вивчення його тими юзерами, які не кваліфіковані та не мають базових знань у програмування. Власником JavaScript офіційно є компанія Oracle[13].

JavaScript надає змогу виконувати функції, як: дії з cookie, зміна сторінок веб-браузерів, робота з тегами, редагування стилів, можливість доступу до всіх частин програмного коду сторінки та можливість внесення динамічних змін.

Використання JavaScript дуже широке і ніяк не обмежується: лише веб-додатками. Є можливість для створення тестового редактора і десктопних, мобільних та серверних додатків.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		42

JavaScript має величезну кількість фреймворків для різних потреб, тому для розробника – це дуже популярна та широко необхідна мова у багатьох ІТ-сферах. Приклади найпопулярніших фреймворків зображено на рис. 2.8.



Рис. 2.8. Популярні фреймворки JavaScript

Було досліджено такі переваги JavaScript:

- Всі сучасні веб-браузери використовують JavaScript у своїх цілях.
- Рекомендований для початківців, які не мають досвіду у розробки клієнтської частини.
- Легкозрозумілі та багатофункціональні налаштування.
- Широка взаємодія з технологіями мови розмітки HTML та формальною мовою опису документів CSS. Тому, зрозуміло, що JavaScript є один з найважливіших компонентів у веб-розробці.
- Розроблювати програмний код можна і в звичайних редакторах, хоч вони і не мають достатньо функціональності, але все ж таки мають можливість компіляції.
- Стресостійкість. Навіть при критичних помилках JavaScript буде продовжувати виконання своєї роботи.
- Garbage Collector за необхідності звільнює пам'ять програми.

										ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата							43

Попри велику кількість переваг JavaScript має і недоліки:

- Небезпечне використання у своїх програмах без надійного захисту, оскільки програмний код JavaScript знаходиться у відкритому доступі, що полегшує завдання зловмисників.
- Велика кількість помилок майже на всіх етапах розробки. Навіть попри те, що багато даних помилок були досліджені і вирішені на форумах, але все одно на це необхідно витратити свій час при розробці програмного коду.
- Повсюдне поширення. Специфічним недоліком можна вважати те, що частина активно використовуваних ПЗ (особливо додатків) перестануть існувати при відсутності цієї мови, оскільки цілком базуються на ньому.
- Повна залежність додатків від даної мови програмування.

## 2.4.2. Веб-фреймворк AngularJS

AngularJS – відкритий фреймворк JavaScript для клієнтської програмної частини програмного веб-додатку, архітектура та особливості даної технології зображено на рис 2.9.

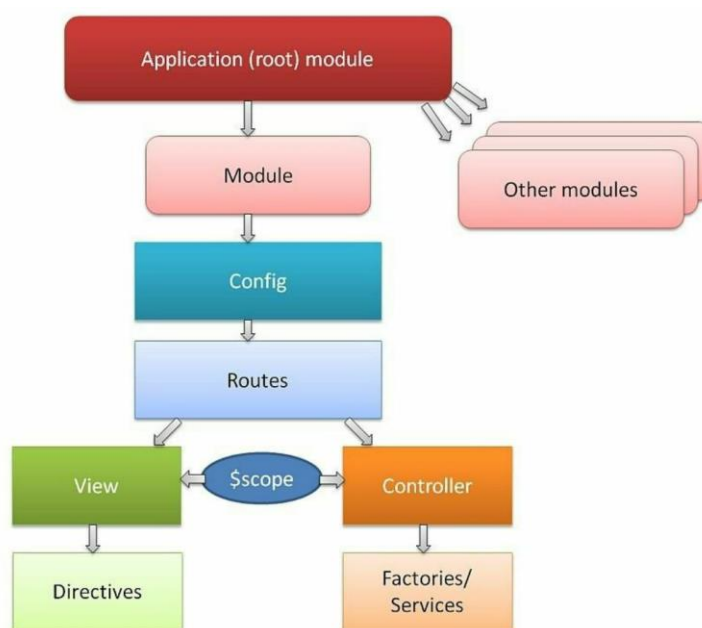


Рис. 2.9. Архітектура та особливості AngularJS[14]

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		44

Фреймворк підтримується відомою ІТ-компанією Google та спільнотою користувачів, які надають оновлення продукту. Дане програмне забезпечення спрощує створення клієнтської частини додатків. AngularJS написаний на мові програмування JavaScript, отже система працює на всіх програмних компонентах, які використовують дану мову програмування .

Перш за все, фреймворк розробляє SPA та використовує архітектуру MVC. AngularJS використовується для розробки багатьох сайтів, найвідоміший серед яких Google.

Досліджено дані особливості AngularJS:

- Зв'язування даних між собою, що зменшує цикли оновлення та зменшує програмний код на 20%.
- Директиви – це елементи AngularJS, що використовуються за необхідності повторно і маніпулюють DOM.
- Немає необхідності рендерити сторінки.
- Забезпечення вбудованими сервісами.
- Використання MVW.

Досліджено дані переваги AngularJS:

- Можливість створювати односторінкові додатки, при цьому надаючи чистий та підтримувальний програмний код.
- Прив'язка до HTML.
- Використання Unit-тестування задля забезпечення якості продукту.
- Повторне використання елемента, що забезпечує чистоті програмного коду.
- Представлення в AngularJS - це HTML сторінками. Бізнес логіка – JavaScript контролери.

Досліджено дані недоліки AngularJS:

- Складність. Доволі важкозрозумілий, особливо для тих, хто немає достатньої кваліфікації та знань.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		45

- Незрозумілість при рішення проблем. Попри те, що можна вирішити певну проблему багатьма способами, це є і недоліком, адже не вказано правильність на сторінці розробників.
- Повільна робота додатку при великій кількості спостерігачів.
- Негнучкий, оскільки при росту проекту, важко розширювати додаток з клієнтської частини.

### 2.4.3. Веб-фреймворк Vue.js

Vue - це фреймворком JS для розробки інтерфейса користувача. Vue розроблявся з самого початку для адаптивності. Орієнтованість на клієнтське представлення, легка інтеграція з програмними бібліотеками та проектами. Vue також використовується для SPA[15], поєднуючись з новітніми технологіями розробки. Обробка логіки на клієнтській частині Vue.js зображена на рис 2.10.

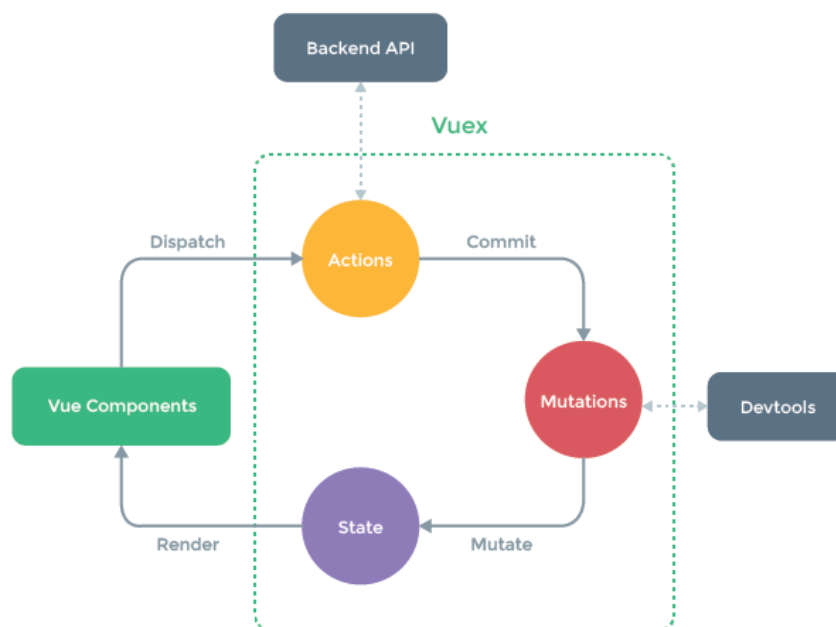


Рис.2.10. Обробка логіки на клієнтській частині Vue.js[15]

Vue.js розробив Evan You, в минулому співробітник компанії Google. Фреймворк наявний у програмному кодї таких організацій як Xiaomi, Baidu,

Alibaba та інші. Vue.js міститься в Laravel і PageKit, а система управління версіями GitLab нещодавно перейшла на даний фреймворк. Підтримується тільки компанією-розробницею. Попри невеликий розмір, не більше 30 кБ, і Vue.js має гарну продуктивність рівняючи з фреймворками AngularJS та ReactJS.

Були виявлені дані переваги Vue.js при дослідженні:

- Легка інтеграція з існуючими веб-додатками.
- Невеликий розмір фреймворку, не більше 30 кБ.
- Детальна документація на головному сайті Vue.js.
- Гнучкість. Компоненти є можливість писати у файлах HTML та JavaScript.
- Архітектура MVVM забезпечує двосторонній зв'язок.

Були виявлені дані недоліки Vue.js при дослідженні:

- Недостатньо документації, оскільки Vue.js молодий фреймворк, що розвивається, тому інформації про особливості фреймворку набагато менше чим у ReactJS та AngularJS.
- Створення нових сервісів Vue.js неангломовними розробниками, що знижує використання та розуміння даних технологій розробниками.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		47

## ВИСНОВКИ ДО РОЗДІЛУ 2

У даному розділі було досліджено вимоги до технологій розробки програмного забезпечення, а саме мови програмування для серверної частини, системи управління базами даних та технологій для представлення клієнтського інтерфейсу, з огляду на розроблюваний веб-додаток. Проведено дослідження найпопулярніших сучасних технологій розробки – мов програмування: Java, Python, C#, JavaScript; серед СУБД: PostgreSQL, MySQL, SQLite; серед веб-фреймворків для клієнтської частини програми: AngularJS, Vue.js. Детально оцінено їх переваги та недоліки.

Для серверної частини було обрано мову програмування Java у зв'язку з зручністю розробки, простого та зрозумілого синтаксису, підтримку ООП, наявність розвинутого фреймворку Spring для ефективного створення веб-додатків, кросплатформеність, регулярну появу актуальних оновлень та відкритих для використання великої кількості бібліотек для розробки програмного забезпечення.

Для збереження даних з досліджених СУБД було обрано PostgreSQL, тому що вона є масштабованою, високопродуктивною, та надає високий рівень безпеки користувачеві. Є можливість керування структурованими так і неструктурованими даними, що забезпечує стабільність управління даними для розроблюваного додатку. Дана СУБД є добре сумісною з фреймворком Spring та офіційно підтримується, тому забезпечується ефективність та зручність роботи із PostgreSQL.

Мову програмування для реалізації клієнтської частини було вирішено обрати JavaScript, оскільки: дана мова займає лідерські місця при виборі для реалізації веб-додатків тому JavaScript один з найкраще пристосованих для веб-розробки; велика кількість різноманітних фреймворків у відкритому доступі; легко інтегрується з серверною частиною, яка буде реалізована на Spring.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		48

## РОЗДІЛ 3

### РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ

#### 3.1. Функціональні вимоги до додатку

Необхідно пройти аутентифікацію та авторизацію за для того, щоб мати змогу використовувати систему у своїх цілях. Це надає програмному забезпеченню вищий рівень безпеки та надійності.

Авторизовані користувачі мають дану основну функціональність та доступ до таких веб-сторінок:

- головна сторінка;
- перегляд та редагування особистих даних;
- перегляд можливих кімнати для дозвілля;
- перегляд календаря запису бронювання усіх подій до кожної з кімнат відпочинку;
- можливість бронювання події на зручний йому час
- редагування та видалення при потребі бронювання;
- отримання листа на особисту електронну пошту при створенні, редагуванні чи видаленні події;
- перегляд історії свого бронювання.

У системі наявна також роль адміністратора, який має окрім функціональності користувача також дещо більшу функціональність задля того, щоб забезпечити якісне та ефективне управління веб-додатком:

- додання нових користувачів
- зміна та видалення подій інших користувачів
- перегляд усієї інформації про користувачів, університетів, гуртожитків та наявних кімнат
- додання нових університетів, гуртожитків та наявних кімнат
- редагування університетів, гуртожитків та наявних кімнат
- перегляд історії резервувань усіх юзерів

					ІАЛЦ.467100.003 ПЗ	Арк
						49
<i>Змн.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

### 3.2.Опис архітектури веб-додатку

Клієнт-серверна архітектура була обрана для розробки та створення даного веб-сервісу. Дана архітектура – це архітектура яка умовно поділяється на дві частини: клієнт та сервер.

Клієнти звертаються до сервера, який зберігає, оброблює та повертає дані, які необхідні користувачеві, тому він є центральною та головною частиною даної архітектури. Можна зробити припущення, яке буде вірним, що сервер за необхідності може існувати без юзерів, але користувачі без сервера уже ніяк не здатні працювати. Клієнтів зазвичай набагато більше ніж серверів. До прикладу, до єдиного сервера, який працює з сайтом підключається величезна кількість браузерів. Користувачі використовуючи програму-клієнт спілкуються з сервером, також клієнтом може виступати робот, штучний інтелект.

У навантажених веб-додатках серверів також є велика кількість, але це не означає, що клієнтів менше, а навпаки на такій системі найбільша аудиторія.

Схема клієнт-серверної архітектури зображено на рис 3.1.

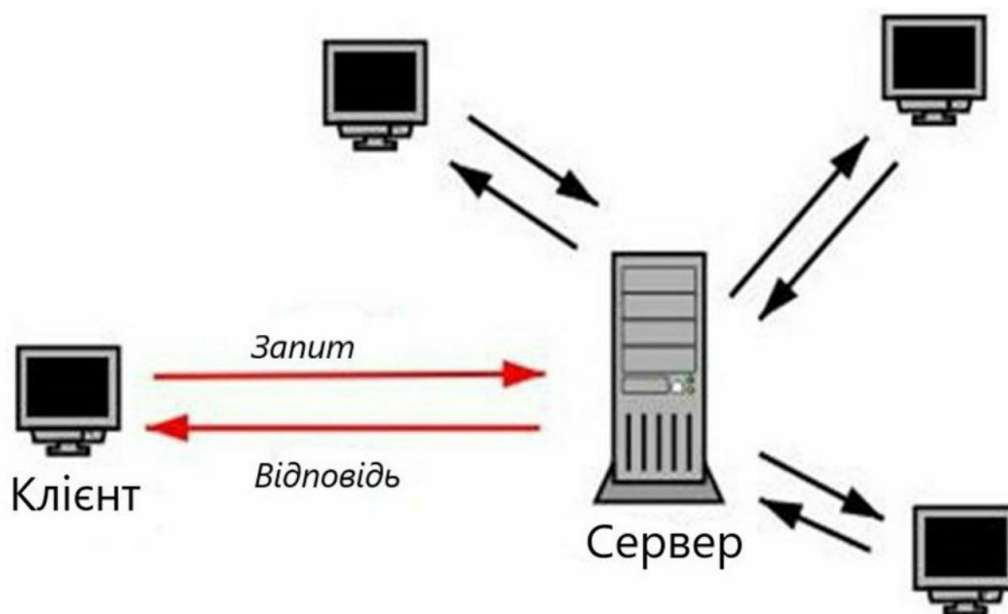


Рис. 3.1. Схема клієнт-серверної архітектури

Переваги клієнт-серверної архітектури[17]:

					ІАЛЦ.467100.003 ПЗ	Арк
						50
Змн.	Арк.	№ докум.	Підпис	Дата		

- Розділений програмний код клієнтської та серверної частини, що надає нам гнучкості у розробці, а саме команда бекендщиків та фронтендщиків має можливість паралельно виконувати свою роботу, тому розробка стає швидшою та ефективною.
- Знижені вимоги до клієнтських девайсів для локальних комп'ютерних мереж, оскільки обчислювальні операції будуть працювати на сервері
- Робота сервера та клієнтської частини доволі окремо, що надає паралельності та швидкості обробки запитів користувачів
- Можливість налаштування свого рівня безпеки

#### Недоліки клієнт-серверної архітектури:

- Дорога вартість серверного обладнання
- Необхідний навчений і підготовлений спеціаліст з великим багажем знань та відповідною кваліфікацією для обслуговування за сервером
- Жорстка залежність клієнтської частини від роботи серверної частини. У разі неполадок у серверній частині, клієнтська не буде працювати. Також при невеликій потужності сервера, запити клієнтів будуть повільно оброблюватися.

Як висновок, варто звернути увагу на те, що клієнт-серверна архітектура не поділяє машини на лише клієнтів або лише сервер, а має більш глибоку мету. Дана архітектура розподіляє навантаження і розділяє роботу між клієнтською та серверною частиною, що надає ефективності та швидкості роботи клієнтів.

Для створення веб-додатку брався архітектурний шаблон проектування MVC. Дане архітектурне рішення розбиває всю систему на три основні компоненти[18]:

- Model – це основний елемент шаблону, який динамічно структурує дані програмного забезпечення, який працює

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		51

паралельно від інтерфейсу користувача, даний шаблон проектування управляє даними додатку, бізнес-логікою та правилами їх використання та обробки.

- View – це частина архітектури, яка забезпечує представлення інформації. Наприклад, графіки, таблиці, події та інші дані.
- Controller – це частина архітектури, яка приймає дані надіслані користувачем у View, передає їх для обробки Model, очікує відповідь, яку повертає View для перегляду.

Схему та взаємодію між компонентами Model, View, Controller зображені на рис 3.2.

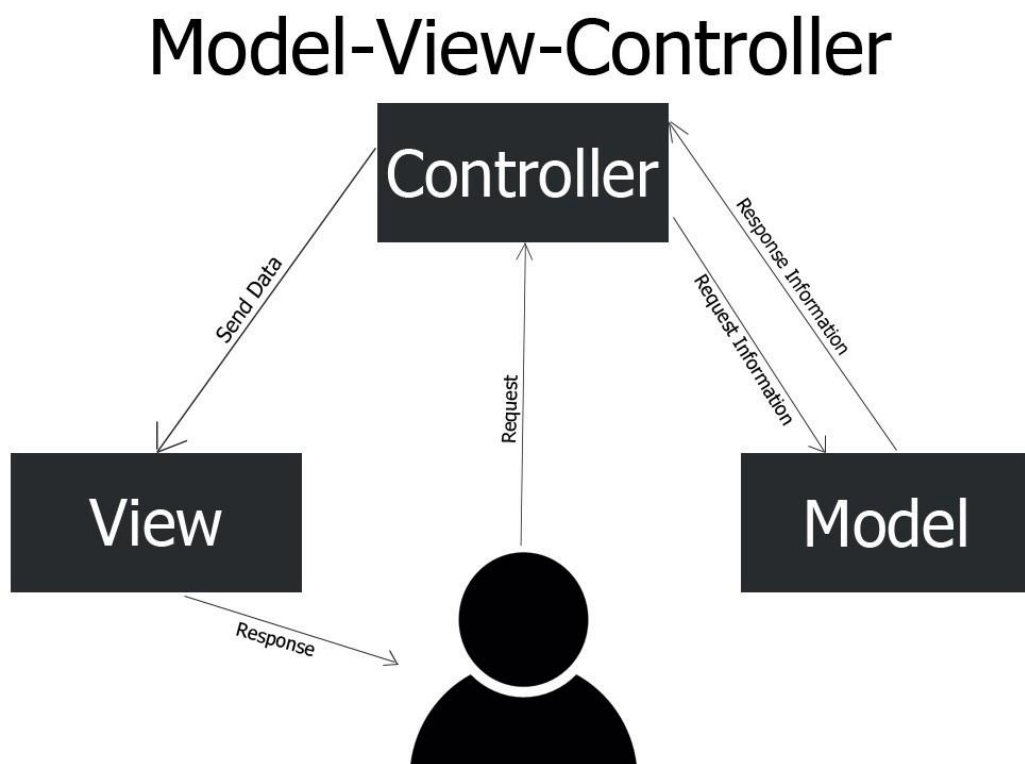


Рис. 3.2. Структура шаблону проектування MVC

Також, крім розподілу додатку на дані елементи, архітектура MVC вказує взаємодію між даними елементами.

Розглянемо основні переваги шаблону проектування MVC[19]:

- Згуртованість. Шаблон проектування MVC надає логічне групування операцій разом на певному контролері. Представлення для кожної Моделі теж згруповані.
- Якісна та зрозуміла архітектура всієї програми.

- Слабий зв'язок між компонентами MVC.
- Легка для модифікацій та змін програма через поділ обов'язків.
- Спрощується підтримка додатку.
- При необхідності може існувати кілька Представлень для однієї Моделі.

Недоліки шаблону проектування MVC:

- Велика кількість ресурсів для створення і підтримки даної архітектури.
- Кожен модуль повинен ділитися на три компоненти Model View Controller, що ускладнює архітектуру, оскільки це потрібно робити навіть і для маленьких модулів системи.

Отже, дослідивши архітектуру додатку переходимо на структурну схему нашої програми, яка зображена на рис 3.3.

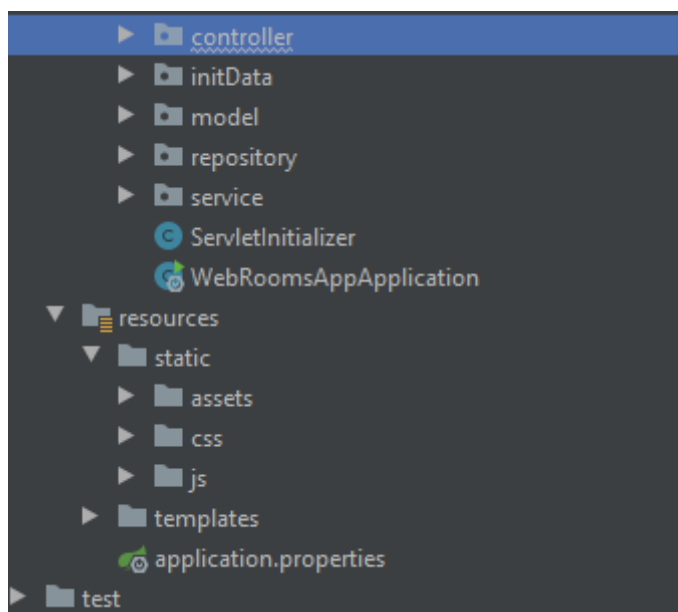


Рис. 3.3. Структурна схема веб-додатку

Основні компоненти нашого програмного забезпечення згруповані по package (програмні папки в Java) :

- Controller – містить всі контролери нашої програми, елементи MVC, які відповідають за взаємодію між View і Model

- `InitData` – містить значення контенту, які необхідно встановити за стандартом, наприклад університети, кімнати, гуртожиток, акаунти адміністраторів
- `Model` – містить всі моделі, дані нашої програми, а також відповідає за їх зберігання
- `Repository` – містить репозиторії, які необхідні для спілкування програми з базою даних
- `Service` – містить сервіси для обробки та роботи бізнес-логіки
- `WebRoomsAppApplication` – за допомогою `Spring Boot` запускає, контролює наше програмне забезпечення
- `Resource` – в даній папці містяться файли для клієнтської частини та файли конфігурації
- `Assets` – містить веб-контент, який необхідний для представлення, наприклад зображення фону, навігаційного меню, іконок і так далі
- `Css` – документи для оформлення стилів елементів представлення
- `Js` – файли JavaScript для конкретної роботи з клієнтською частиною, а також саме тут знаходиться один з найголовніших частин даного програмного забезпечення, а саме створення та підтримка календаря подій
- `Templates` – містить всі веб-сторінки, які реалізовані за допомогою мови розмітки HTML
- `Application.properties` – файл для роботи з конфігурацією програми. Саме тут знаходиться підключення до бази даних PostgreSQL, а також підключення MailSendera, задля забезпечення відправки повідомлень на електронну пошту при створенні, зміні та видаленні якоїсь події.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		54

### 3.3.Серверна частина додатку

Серверна частина даного веб-додатку складається з Controller, Service, Repository, Model та бази даних, як це зображено на рис 3.4.

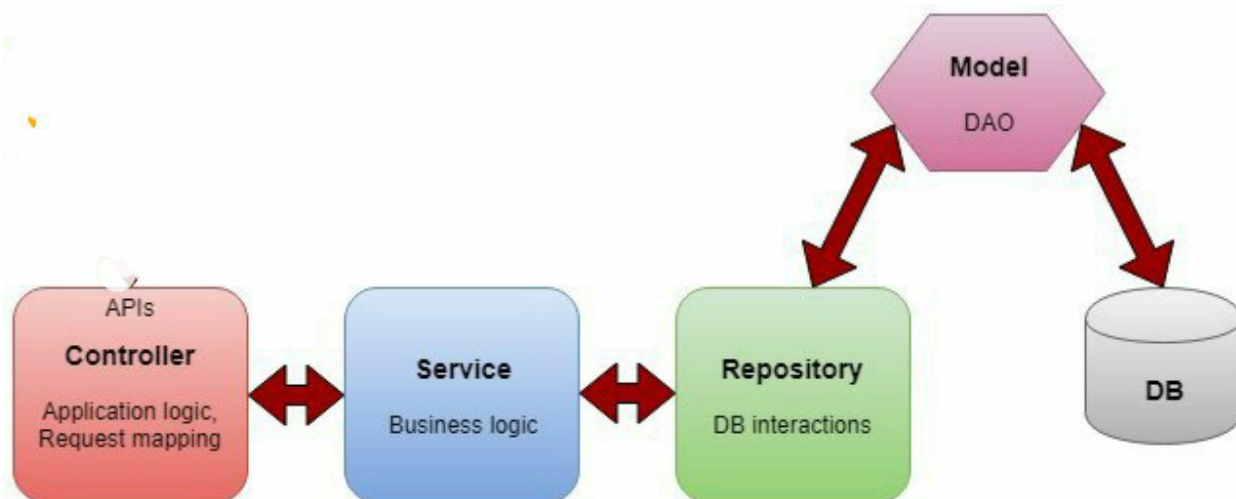


Рис. 3.3. Структурна схема веб-додатку

Controller отримує запит від користувача, оброблює дані та передає Service. Service виконує бізнес-логіку програми, якщо йому необхідні дані Model він звертається до Repository, який за допомогою вбудованого JDBC-драйвера звертається до бази даних PostgreSQL, який повертає Entity репозиторію, репозиторій у свою чергу повертає дані сервісу, який виконує певну бізнес-логіку і результат надає контролеру, який нарешті повертає відповідь представленню, щоб зобразити користувачеві.

Більш детально про кожного з них, і як він використовується у даній бакалаврській роботі буде описано в наступних частинах розділу.

#### 3.3.1. Model архітектури MVC веб-застосунку

Model має в собі всю бізнес-логіку програми. У ній міститься код та об'єкти, що створюють сутності веб-додатку, які можуть відповідати реальним об'єктам в реальному світі. Дана частина найбільш незалежна від інших частин архітектури.[20]

В даному веб-додатку існує всього 5 моделей:

- Account – це авторизований користувач даної веб-системи, студент, адміністратор та інші ролі
- Event – це подія бронювання нашого дозвілля
- Hostel – це гуртожиток в якому відбуватиметься бронювання
- Room – це кімната, в якій відбуватиметься бронювання дозвілля
- University – це університет, в якому відбуватиметься бронювання даної системи

Всі моделі зберігаються в базі даних, щоб мати більшу захищеність та легкий доступ до них. Отже, розглянемо дані сутності, як вони зберігатимуться в базі даних, їх поля та властивості. Варто зауважити, що поле id - унікальний ідентифікатор користувача, міститься у всіх сутностей.

Перша модель – це Account користувача:

- Username – логін користувача
- Password – пароль користувача
- Email – електронна пошта користувача
- Token – токен, який забезпечує високий рівень безпеки, створюється за допомогою Spring Security
- Role – роль користувача. Можливі ролі : неавторизований користувач (анонім), авторизований користувач, який пройшов процес авторизації та аутентифікації та адміністратор
- Firstname – ім'я користувача
- Lastname – фамілія користувача
- Address – адреса проживання користувача, його гуртожиток
- Group – група користувача в університеті
- Lastlogin – останнє вдале входження користувача у систему
- Event – кількість заброньованого дозвілля в даний момент
- University – університет в якому навчається студент
- isAdmin – наявність у користувача адміністраторських прав

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		56

Розглянемо сутність Event:

- title – назва, з якою метою відбувається бронювання події
- start – початок бронювання події користувачем
- end – кінець бронювання події користувачем
- room – кімнаті в якій відбувається бронювання
- Account – користувач, який забронював дану подію

Розглянемо сутність Hostel:

- name – назва, номер гуртожитку
- address – адреса гуртожитку
- university – університет, до якого відноситься даний гуртожиток

Розглянемо сутність University:

- name – повна назва університету

Розглянемо сутність Room:

- name – назва кімнати дозвілля. Поки наявні такі кімнати, як пінг-понг, спортзал, танцювальна та музична кімната
- floor – поверх гуртожитку, на якому знаходиться кімната
- capacity – кількість студентів, яка вміщує дана програма
- sportsEquipment – наявність спортивних інструментів для проведення активного дозвілля
- lunchZone – наявність зони для перекусу
- lighting – наявність додаткового освітлення
- description – додаткова інформація, яку необхідно вказати
- hostel – гуртожиток, до якої відноситься дана кімната

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		57

### 3.3.2. Service, Repository, Controller веб-додатку

Попри те, що вони виконують різні функції, їх вирішено додати в один пункт, оскільки вони тісно пов'язані між собою. Service за необхідності звертається до Repository, щоб отримати дані для виконання певної бізнес-логіки.

Репозиторій - це шаблон доступу до даних, моделі, в якому об'єкти даних за допомогою драйвера передаються в сховище даних, найчастіше це база даних. Має обов'язково реалізовувати операції CRUD, та інші додаткові операції за необхідності. Основна перевага даного шару є в тому, що забезпечує механізм доступу до ваших для будь-якого сховища динамічно. Наприклад, в одному веб-додатку є різні сховища даних, такі як PostgreSQL та MySQL і в цьому випадку немає необхідності переписувати програмний код чи його розширювати, лише змінити базу даних у конфігурації[21].

Сервісний шар - це єдина схема, яка дозволяє обробляти, керувати бізнес-логікою, часто використовується в архітектурах SaaS. Використовуючи даний рівень різні реалізації презентацій отримують доступ до вашої бізнес-логіки через загальний інтерфейс[22].

Контролер – це компонент архітектури MVC, який отримує дані від Model або View, перетворює їх на команди та передає далі по логічному ланцюжку[23].

Основні сервіси та репозиторії програмного забезпечення:

- AccountService – сервіс для обробки бізнес-логіки користувача, додання нового юзера, логіну, створення токена Spring Security, редагування, видалення аккаунта, експорт даних та перегляд всіх наявних користувачів. Використання UserDetailsService для більш безпечного рівня продукту.
- EventService – один з найголовніших сервісів веб-додатку, він необхідний для управління подіями, зокрема створення, редагування, видалення, валідація, обчислювальні операції з подіями, експорт, архівування подій

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		58

- HostelService – сервіс для управління гуртожитками, забезпечуючи всі необхідні операції для ефективного управління.
- MailService – сервіс для надсилання повідомлень електронною поштою користувачам при якійсь дії з подією, наприклад додавання, редагування, видалення, як збоку користувача, так збоку і адміністрації
- RoomService – сервіс для керування кімнатами дозвілля в гуртожитку, який забезпечує коректне роботу з даними об’єктами
- UniversiteService – сервіс який забезпечує коректну роботу бізнес-логіки з університетом
- AccountRepository – репозиторій, який забезпечує доступ до даних, які пов’язані з користувачем, виконує стандартні CRUD-операції, а також виконує пошуки за окремими полями за необхідності
- EventRepository – репозиторій, забезпечує роботу з даними, які на пряму пов’язані з подіями, забезпечує виконання CRUD-операцій та виконує пошук подій за певними критеріями
- HostelRepository – репозиторій, забезпечує роботу з моделями гуртожитків, виконання CRUD-операцій, пошук подій за певними полями.
- RoomRepository – репозиторій для забезпечення роботи з кімнатами дозвілля, виконує CRUD-операції, а також введе пошук кімнат за вказаними критеріями
- UniversiteRepository – репозиторій задля забезпечення коректної роботи з університетами, виконує CRUD-операції.
- Account, Event, Hostel, Room, University Controllers – це контролери, які забезпечують взаємодію між відповідними їх Model та View, приймаючи дані та перетворюючи їх у команди, передаючи їх далі по архітектурі продукту.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		59

### ВИСНОВКИ ДО РОЗДІЛУ 3

В розділі 3 розглянуто загальний опис розробленої клієнт-серверного веб-додатку, визначено вимоги, які необхідно задовольняти реалізованому продукту, як для неавторизованих, так і для авторизованих користувачів двох типів: звичайного користувача та адміністратора. Всі вимоги повністю досліджено та реалізовано у веб-додатку.

Описано та реалізовано у веб-застосунку клієнт-серверну архітектуру, зокрема MVC (Model, View, Controller), яка наразі є популярною для створення веб-додатків, тому що надає потрібний рівень інкапсуляції, простоту читабельності програмного коду для підтримки продукту та мінімізацію виникнення помилок.

Описано загальні риси використаної в програмному забезпеченні клієнт-серверної архітектури, зокрема серверної частини. Детально розібрані дані компоненти: моделі, сервіси, контролери та репозиторії, які використані у веб-застосунку.

Для створення даного веб-застосунку були використані: мова програмування Java (фреймворк Spring) – для серверної частини; JavaScript – для розробки клієнтської частини (разом з HTML та CSS) та модулю, який відповідає за бронювання дозвілля в гуртожитку; за підключення до баз даних відповідала СУБД PostgreSQL.

					ІАЛЦ.467100.003 ПЗ	Арк
						60
Змн.	Арк.	№ докум.	Підпис	Дата		

## РОЗДІЛ 4

### АНАЛІЗ РЕЗУЛЬТАТІВ СТВОРЕНОГО ВЕБ-ЗАСТОСУНКУ

#### 4.1. Ілюстрація роботи програми (інструкція з користування)

Оскільки результатом даної бакалаврської роботи є веб-застосунок з зручним та інтуїтивно-зрозумілим графічним користувацьким інтерфейсом, тому ілюстрація роботи програми та інструкція з користування будуть майже ідентичні, тому немає необхідності розділяти їх окремими пунктами, щоб не нагромаджувати роботу.

Для того, щоб запустити адміністратору дане програмне забезпечення необхідно обрати будь-яку операційну систему (з встановленими раніше середовищем розробки IntelliJ Idea, JDK, СУБД PostgreSQL), відкрити IntelliJ з програмним кодом, встановити з'єднання з СУБД, скомпілювати та запустити програмне забезпечення. Запустивши, ми переходимо на сторінку авторизації та аутентифікації, та нажимаємо кнопку «Login», для того щоб ввести свої облікові дані, а саме логін(або електронна пошта) та пароль. Наявна функція «Remember me», яка дозволяє системі зберігати сесію користувача, навіть, коли він вийде з веб-сайту. Веб-сайт відповідає нормам безпеки, тому пароль скритий і злоумисник не зможе побачити пароль користувача при вводі. Дані веб-сторінки зображені на рис 4.1.

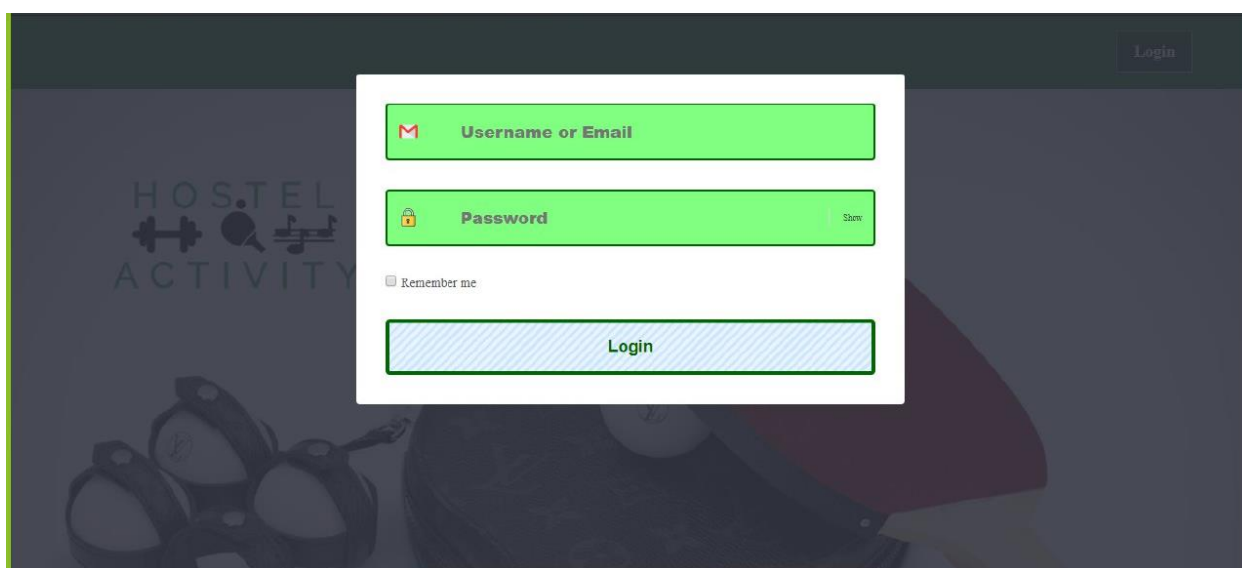


Рис. 4.1. Сторінка авторизація та аутентифікації веб-додатку

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		61

Саме тут і проходила авторизація та аутентифікація. Вдало пройшовши даний етап, користувач отримує свої права та привілеї, як «user» або «admin», відповідно, що і користувацький інтерфейс буде іншим, в адміністратора буде більше функціональності, щоб керувати процесом веб-додатку.

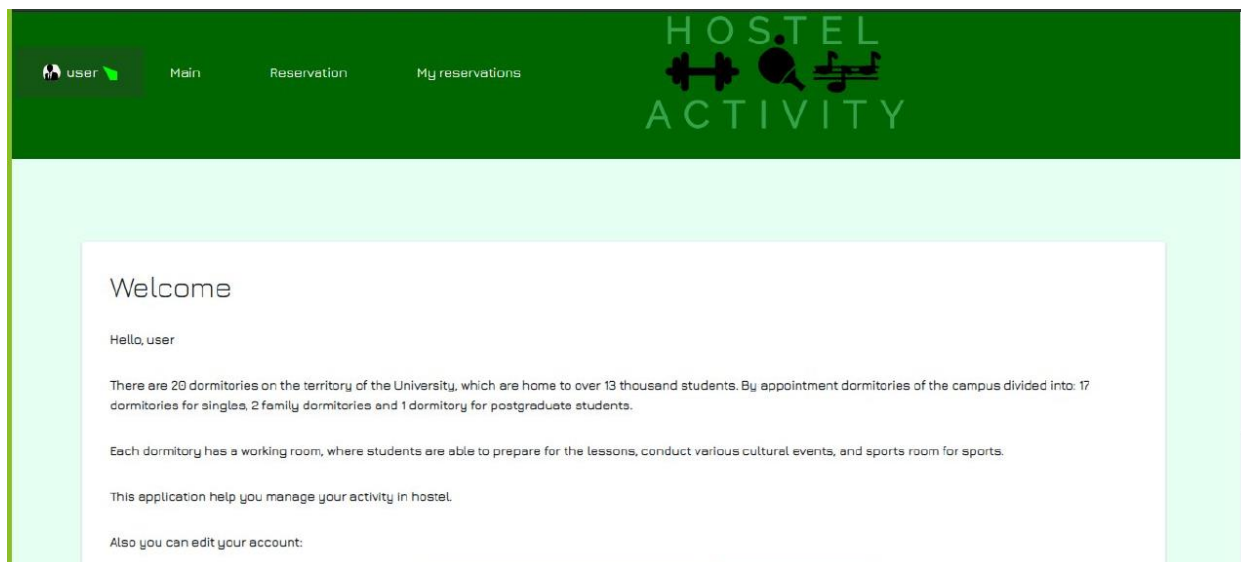


Рис. 4.2. Головна сторінка веб-додатку

Спершу розглянемо функціональність збоку користувача. Після процесу логування він переходить на головну сторінку(main), в якій описано проект та певна інформація про гуртожитки, яка зображена на рис. 4.2. Також на кожній сторінці зображено панель меню, в користувача – це кнопки «main», «reservation», «my reservation», а також управління акаунтом з кнопками «edit»(змінити дані акаунта), «logout»(вийти з облікового запису).

Для того, щоб створити замовлення необхідно перейти на «Reservation», там будуть вказані всі гуртожитки студмістечка(дана дія зображена на рис 4.3), необхідно обрати свій гуртожиток, після цього з'явиться перелік усіх можливих кімнат відпочинку, де можна забронювати певну діяльність (для прикладу це Ping pong, Music room, Gym, Dance room).

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		62



Рис. 4.3. Перелік доступних гуртожитків

Перейшовши по вибраній кімнаті, ми переходимо на систему бронювання кожної кімнати(приклад зображено на рис 4.4). Дана система містить усі власні та синхронізовані з іншими користувачами бронювання на з 10:00 до 22:00 (стандартна робота даних приміщень в гуртожитку). Для зручності використання наявні 2 версії календаря: на весь тиждень(7 days) та на день(1 day). Для маніпулюванням дат необхідно використовувати кнопки «prev»(минулий відрізок часу) «next»(наступний) та «now»(теперішній).

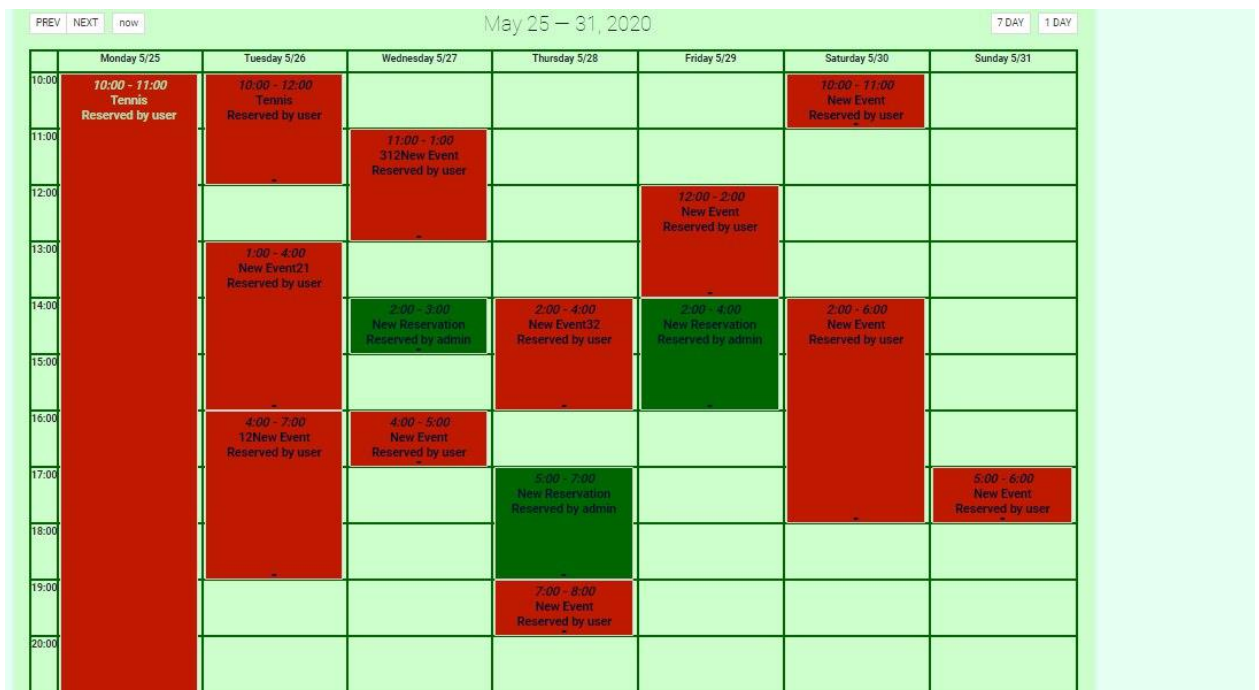


Рис. 4.4. Система бронювання кожної кімнати

Наявні два типи власності подій:

- Власні (відмічені зеленим кольором)
- Сторонні (відмічені червоним кольором)

Для того, щоб створити подію необхідно вибрати зручний час на шкалі, і якщо місце вільне, провести курсором та забронювати час пройшовши форму заповнення події, після цього утвориться дана подія для всіх користувачів. Для того, щоб створити подію(форма вказана на рис 4.5), потрібно вказати дані поля для збереження в базу даних:

- Event title (назва та ціль бронювання)
- Event starts (початок бронювання, вказується дата та час)
- Event ends (кінець бронювання, вказується дата та час)

### Create new Event

Event title	Tennis
Event starts:	2020-05-25T10:00:00
Event ends:	2020-05-26T12:00:00

Create Cancel

Рис. 4.5. Форма створення нового бронювання

Важливим є те, що інші користувачі(крім адміністраторів) не зможуть редагувати, видалити невласне бронювання, а також не мають змоги створити резервацію, якщо буде пересікатися час з існуючими бронюваннями.

При необхідності редагуванні/видаленні власного бронювання, необхідно на нього натиснути та перейти в інтуїтивно-зрозумілу форму заповнення, яка зображена на рис 4.6. Для зміни даних, редагуємо уже відомі

поля Event title, Event Starts, Event Ends на ті, які нам потрібні, але потрібно звернути увагу на те що якщо новий час накладатиметься на існуюче бронювання, зміни відміняться.

### Modify an Event

Event Title: New Event

Event starts: 2020-05-29 12:00:00

Event ends: 2020-05-29 14:00:00

Save
Delete
Cancel

Рис. 4.6. Форма редагування/видалення бронювання

Щоб зберегти зміни необхідно натиснути «Save», для видалення бронювання - «Delete», а щоб відмінити зміни - «Cancel». Після цього всі зміни будуть видні і іншим користувачам.

Наявна функціональність – управління подією в режимі онлайн, а саме зажавши курсором подію, її можна перетягнути і вказати новий час початку та кінця, а при необхідності можна збільшити час події, просто опустивши вниз іконку, звісно, якщо там є можливість.

Список усіх заброньованих подій можна переглянути перейшовши по панелі меню «My reservation», приклад зображено на рис 4.7.

My Reservations 🗑️ ⬆️ 🔍

	Title	Room	Hostel	Start	End	
✎	New Reservation	Ping-pong room	Hostel 1	2020-04-15 07:45:00.0	2020-04-15 09:45:00.0	🗑️
✎	New Reservation	Ping-pong room	Hostel 1	2020-04-17 08:30:00.0	2020-04-17 11:15:00.0	🗑️
✎	New Reservation	Ping-pong room	Hostel 1	2020-04-17 11:45:00.0	2020-04-17 15:00:00.0	🗑️
✎	New Reservation	Ping-pong room	Hostel 1	2020-04-18 17:00:00.0	2020-04-18 18:00:00.0	🗑️
✎	New Reservation	Ping-pong room	Hostel 1	2020-04-19 13:00:00.0	2020-04-19 14:00:00.0	🗑️

Рис. 4.7. Резервації даного користувача

Окрім того, за необхідності користувач може змінити свої дані, як це зображено на рис 4.8.

Edit admin account information

Email: fgzfgzfgz@gmail.com ✓

Address: Str. Borsch 20

Group: IP-04

Name: View

Surname: Zaichenko

Password: [toggle]

Role: ROLE\_ADMIN

University: KPI

Save

Рис. 4.8. Зміна даних користувача

Функціональність адміністратора доповнює користувацьку, це означає, що окрім даних функцій, які описані зверху наявна і додаткова функціональність, зокрема(надавати знімки не є актуальним оскільки сторінок буде дуже багато, а вони будуть ідентичні попереднім, також в разі необхідності, всі дані сторінки будуть показані в демонстрації веб-додатку):

- Додання нового користувача
- Перегляд всіх користувачів
- Створення нового університету
- Перегляд списку всіх наявних університетів
- Перегляд відповідності користувачів та університетів
- Перегляд відповідності університетів та гуртожитків
- Створення нового гуртожитку
- Створення нової кімнати для відпочинку
- Перегляд усіх кімнат та їхніх характеристик гуртожитку
- Редагування кімнат, гуртожитків, університетів
- Перегляд резервацій усіх користувачів з усіма даними (логін користувача, кімната бронювання та її гуртожиток, ціль бронювання, час початку та кінця), який зображено на рис. 4.9.

Edit admin account information

Email: fgzf.fgzf.1999@gmail.com ✓

Address: Str. Borsch 20

Group: IP-54

Name: Vesuj

Surname: Zaichenko

Password: [eye icon]

Role: ROLE\_ADMIN

University: KPI

Save

Рис. 4.9. Резервування усіх користувачів та усіх кімнат

## 4.2. Тестування програмного забезпечення

Контроль за якістю продукту є дуже важливою частиною кожного життєвого циклу розробки, а особливо підтримки веб-застосунку та є рекомендованою для створення та впровадження нової функціональності на систему контролю версій.

Було реалізовано дані типи тестів:

- Unit-тестування (програмні тест-кейси для кожного з модулів для забезпечення коректної їх роботи)
- Інтеграційні тести (програмні тести взаємодії між модулями)
- Системні тести (програмні тести на роботу всієї системи)
- Мануальне тестування (ручне тестування користувацького інтерфейсу).

За результатами усіх тестів були усунені дані типи помилок:

- Помилки інтерфейсу – некоректно підібрані стилі для даного програмного забезпечення, що створює умови для виникнення «недружелюбного» інтерфейсу, що приводить до зменшення кількості відвідувачів веб-застосунку.
- Функціональні. Перевірка коректності роботи веб-додатку та усіх його модулів та частин.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		67

- Авторизації та аутентифікації. Були перевірено всі права та доступи звичайних користувачів та адміністрації.

Отже, за результатами тестування було виправлені всі помилки, а також додані повідомлення клієнтам веб-додатку про помилки, що виникають, якщо користувачі введуть некоректні дані.

#### **4.3. Порівняння з існуючими аналогами**

Розроблений веб-застосунок забезпечує дотримання усіх визначених раніше вимог, зокрема:

- можливість створення, редагування, видалення резервацій активностей студентів в кімнатах гуртожитку;
- можливість управління адміністрацією процесом;
- розсилка всіх дій пов'язаних з бронюванням на електронну пошту;
- перегляд історії бронювань
- авторизація та аутентифікація
- надійність та безпека та інші вимоги описані в розділі 2.

Були досліджені переваги даного веб-застосунку перед сучасними аналогами системи:

- направленість на «студмістечко»
- відсутність реклами
- легкий та інтуїтивно-зрозумілий користувацький інтерфейс
- легке маніпулювання з подіями
- розсилка подій на електронну пошту
- підтримка процесу адміністрацією кожного з гуртожитків

Отже, виконавши аналіз розробленого веб-додатку з сучасними аналогами, можна сказати, що даний продукт є гарною альтернативою для кожного студмістечка, забезпечуючи їх повною функціональністю та надійністю, вирішуючи проблематику бакалаврської роботи.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		68

#### 4.4.Рекомендації для подальшого вдосконалення та підтримки

Створений веб-додаток можна використовувати для локальної системи для регулювання студентського життя кожного гуртожитку студмістечка, хоч і має і свої плани для покращення. Саме тому, створений продукт можна і необхідно вдосконалювати в кращу сторону, додаючи нову функціональність. Серед актуальних покращень для майбутніх підтримки та розширення:

- Додання адміністрації гуртожитків та активних представників студмістечка для керування процесом бронювання
- Створення «Бота» в соціальній мережі Телеграм для кращого відслідковування резервацій
- Розгорнути хостинг в мережі Інтернет
- Розробка мобільної версії продукту для різних платформ
- Реалізація інтернаціоналізації (додання нових мов)
- Розширення аудиторії, зокрема інші ВУЗи України
- Інтеграція з існуючим електронним кампусом

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		69

## ВИСНОВКИ ДО РОЗДІЛУ 4

У розділі 4 було розглянуто реалізований веб-додаток, його функціональність для усіх груп користувачів. Було наведено скріншоти працюючої програми, щоб довести роботоздатність та її ефективність. Проведено дослідження клієнтського графічного інтерфейсу та розписано інструкцію користувача. Були виконані всі вимоги, які спроектовані в технічному завданні.

Тестування продукту пройшло успішно, як програмно, так і мануально. Були впроваджені найнеобхідніші типи тестів: unit-тести, інтеграційні та системні тести. Результатом цього були усунені дані типи помилок: помилки інтерфейсу, функціональні помилки, помилки авторизації та аутентифікації.

Зроблено аналіз всіх переваг даного програмного забезпечення з існуючими сучасними аналогами, за яким можна зробити висновок, що розроблений веб-додаток є гарною альтернативою для гуртожитків, забезпечуючи їх достойною системою регулювання студентського життя.

Було наведено перелік функцій та розробок для покращення даного веб-застосунку, щоб зробити його ще кращим, ефективнішим та зручнішим у використанні.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		70

## ВИСНОВКИ

Результатом бакалаврської роботи є створення доступної системи, яка призначена для систематизації та зручного регулювання студентського дозвілля в гуртожитках.

В ході роботи даного дипломного проекту, зроблено наступні висновки:

1. Було проаналізовано сучасні існуючі аналоги та на основі порівняння їх переваг та недоліків сформовано вимоги до розроблюваного веб-додатку.
2. Порівнявши доступні технології реалізації веб-застосунку, було обрано створення клієнт-серверної архітектури, використовуючи мови програмування Java (зокрема фреймворк Spring) та Javascript, в ролі СУБД обрано PostgreSQL.
3. Створений веб-додаток задовольняє усім визначеним вимогам та містить необхідний мінімум функціоналу, зокрема: забезпечено повний функціонал для коректної та зручної резервації студентської діяльності в гуртожитку; управління процесом роботи веб-застосунку відбувається під наглядом адміністрації; підключено сповіщення по електронній пошті; збереження архіву резервацій користувачів; реалізована відповідна авторизація та аутентифікація; забезпечено надійну та безпечну роботу продукту.
4. Усі модульні, інтеграційні, системні та мануальні тести веб-додатку були успішно реалізовані та пройдені. Проаналізовано усі переваги даного програмного забезпечення перед існуючими аналогами. Оскільки це початкова версія продукту, система має ряд недоліків, серед яких - запуск на локальному сервері. Досліджено першочергові розширення в разі подальшого розвитку веб-застосунку.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		71

Дана бакалаврська робота надала можливість більш детально та поглиблено вивчити сучасні технології розробки веб-застосунків, а також розширити та вдосконалити практичні навички при проектуванні клієнт-серверної системи.

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		72

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Outlook для бізнесу – Microsoft [Електронний ресурс]. Режим доступу: <https://www.microsoft.com/uk-ua/microsoft-365/outlook/outlook-for-business/> (дата звернення 19.04.2020)
2. Outlook.com - безкоштовна особиста електронна пошта [Електронний ресурс]. Режим доступу: <https://office.live.com/start/> (дата звернення 19.04.2020)
3. Agenda - Google Calendar [Електронний ресурс]. Режим доступу: <https://calendar.google.com/calendar/r/> (дата звернення 19.04.2020)
4. Як ефективно використовувати Google Calendar? [Електронний ресурс]. Режим доступу: <https://roman.net.ua/howto/> (дата звернення 19.04.2020)
5. Jorte Calendar [Електронний ресурс]. Режим доступу: <https://www.jorte.com/> (дата звернення 19.04.2020)
6. Jorte - функціональний органайзер для android [Електронний ресурс]. Режим доступу: <https://android4all.ru/soft/instruments/199-jorte-funktsionalnyj-organajzer-dlja-android/> (дата звернення 19.04.2020)
7. Java | Oracle [Електронний ресурс]. Режим доступу: <https://www.java.com/> (дата звернення 24.04.2020)
8. The Spring Framework arhitecture [Електронний ресурс]. Режим доступу: [https://www.researchgate.net/figure/The-Spring-Framework-arhitecture-1-Some-of-the-most-important-modules-are-Core\\_fig1\\_321757987/](https://www.researchgate.net/figure/The-Spring-Framework-arhitecture-1-Some-of-the-most-important-modules-are-Core_fig1_321757987/) (дата звернення 24.04.2020)
9. Мова C# і .NET Framework [Електронний ресурс]. Режим доступу: [https://professorweb.ru/my/csharp/charp\\_theory/level1/infonet.php/](https://professorweb.ru/my/csharp/charp_theory/level1/infonet.php/) (дата звернення 24.04.2020)
10. PostgreSQL - об'єктно-реляційна система управління базами даних [Електронний ресурс]. Режим доступу: <https://web-creator.ru/articles/postgresql> (дата звернення 15.04.2019)

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		73

11. Проектування бази даних MySQL [Електронний ресурс]. Режим доступу <http://strong-planet.com.ua/database.html/> (дата звернення 24.04.2020)
12. SQLite — замечательная встраиваемая БД [Електронний ресурс]. Режим доступу <https://habr.com/ru/post/149356/> (дата звернення 24.04.2020)
13. Топ-5 JS-фреймворків для фронтенд-розробки [Електронний ресурс]. Режим доступу: <https://habr.com/ru/company/ruvds/blog/476286/> (дата звернення 24.04.2020)
14. AngularJS — Superheroic JavaScript MVW Framework [Електронний ресурс]. Режим доступу: <https://angularjs.org/> (дата звернення 15.04.2019)
15. Vue.JS: особливості, переваги та недоліки [Електронний ресурс]. Режим доступу: <https://jetruby.com/ru/blog/vue-js-preimuschestva-i-nedostatki/> (дата звернення 24.04.2020)
16. Клієнт-серверна архітектура [Електронний ресурс]. Режим доступу: [https://uk.wikipedia.org/wiki/Клієнт-серверна\\_архітектура/](https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура/) (дата звернення 12.05.2020)
17. MVC для веб: простіше нікуди [Електронний ресурс]. Режим доступу: <https://habr.com/ru/post/181772/> (дата звернення 12.05.2020)
18. Концепція MVC для чайників [Електронний ресурс]. Режим доступу: <https://ruseller.com/lessons.php?id=666> (дата звернення 12.05.2020)
19. CRUD using the Repository Pattern in MVC [Електронний ресурс]. Режим доступу: <https://www.c-sharpcorner.com/UploadFile/3d39b4/crud-using-the-repository-pattern-in-mvc/> (дата звернення 12.05.2020)
20. Best Practice for Services with MVC [Електронний ресурс]. Режим доступу: <https://stackoverflow.com/questions/22500280/best-practice-for-services-with-mvc/> (дата звернення 12.05.2020)
21. What is the role of Controller in MVC model? [Електронний ресурс]. Режим доступу: <https://stackoverflow.com/questions/47884943/what-is-the-role-of-controller-in-mvc-model> (дата звернення 12.05.2020)

					ІАЛЦ.467100.003 ПЗ	Арк
Змн.	Арк.	№ докум.	Підпис	Дата		74

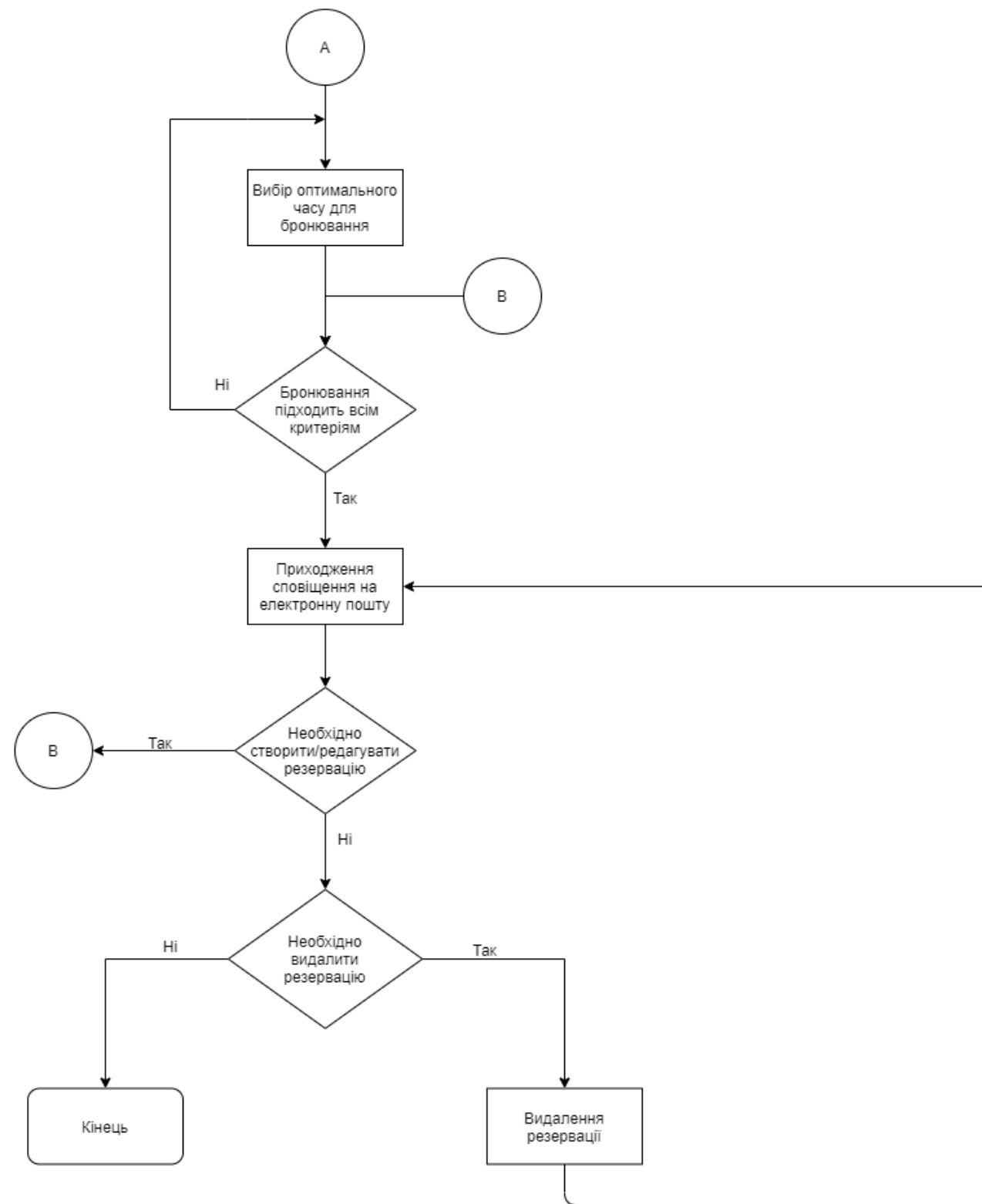
# **ДОДАТОК 1**

**ВЕБ-додаток системи регулювання студентського дозвілля в  
гуртожитку**

**Алгоритм веб-додатку  
ІАЛЦ.467100.004 Д1**

Аркушів 1

Київ 2020 р



					<b>ІАЛЦ.467100.004 Д1</b>			
Зм.	Арк.	№ докум.	Підпис	Дата	ВЕБ-додаток системи регулювання студентського дозвілля гуртожитку Алгоритм веб-додатку	Літ.	Маса	Масштаб
Розроб.	Зайченко І. В.							
Перевір.	Аленін О. І.					Арк.	Аркушів	
Н. контр.	Сімоненко В. П.				<b>Дипломна робота</b>			КПІ ФІОТ кафедра ОТ гр. ІП-64
Затверд.								

## **ДОДАТОК 2**

**ВЕБ-додаток системи регулювання студентського дозвілля в  
гуртожитку**

**Функціональна схема, діаграма класів  
ІАЛЦ.467100.005 Д2**

Аркушів 1

Київ 2020 р



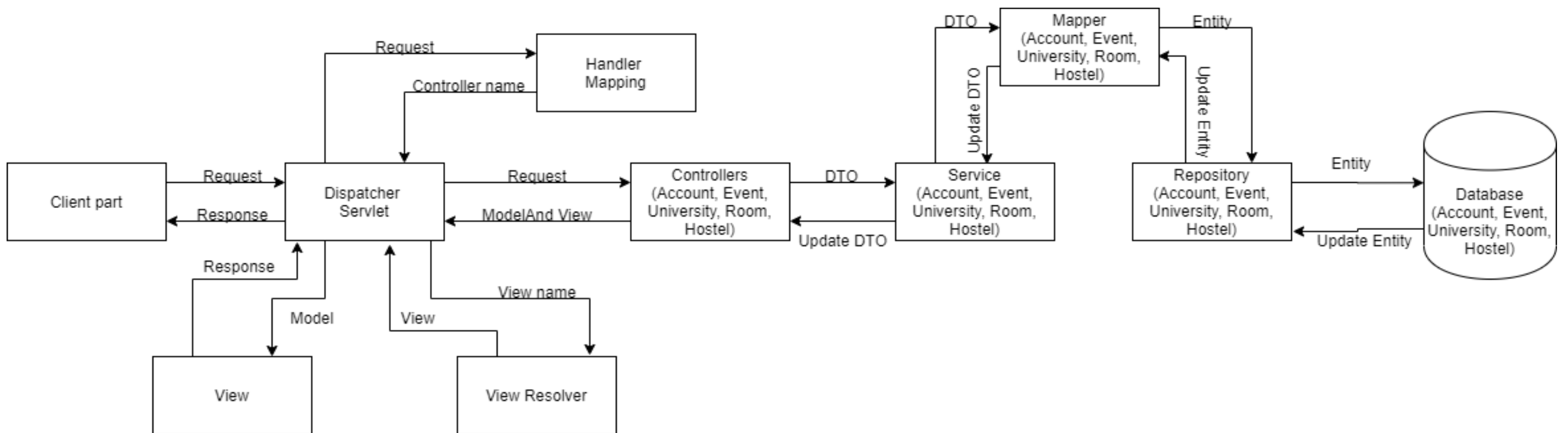
## **ДОДАТОК 3**

**ВЕБ-додаток системи регулювання студентського дозвілля в  
гуртожитку**

**Структурна схема клієнт-серверного веб-додатку  
ІАЛЦ.467100.006 ДЗ**

Аркушів 1

Київ 2020 р



					<b>ІАЛЦ.467100.006 ДЗ</b>					
					ВЕБ-додаток системи регулювання студентського дозвілля гуртожитку Структурна схема клієнт - серверного веб-додатку			Літ.	Маса	Масштаб
Зм.	Арк.	№ докум.	Підпис	Дата						
Розроб.	Зайченко І. В.				Дипломна робота			Арк.	Аркушів	
Перевір.	Аленін О. І.							КПІ ФІОТ кафедра ОТ гр. ІП-64		
Н. контр.	Сімоненко В. П.									
Затверд.										

## **ДОДАТОК 4**

**ВЕБ-додаток системи регулювання студентського дозвілля в  
гуртожитку**

**Текст програми  
ІАЛЦ.467100.007.Д4**

**Аркушів 11**

**Київ – 2020року**

```

@Entity
@NoArgsConstructor
@AllArgsConstructor
@Getter
@Setter
@ToString
public class Account {
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Id
    private Long id;

    @NotNull
    @Size(min = 3, max = 100, message = "Username must have at least 3 characters.")
    private String userName;

    @NotNull
    @Size(min = 3, max = 100, message = "Password must have at least 3 characters.")
    private String password;

    @Transient
    private String confirmPassword;

    @Email(message = "Email address is not valid.")
    @NotNull
    private String email;

    private String token;

    private String role = "ROLE_USER";

    private String firstName;

    private String lastName;

    private String address;

    private String groupName;

    private String lastLogin;

    private int countEvent = 0;

    @NotNull
    @ManyToOne(cascade = CascadeType.MERGE)
    private University university;

    public Boolean isAdmin() {
        return this.role.equals("ROLE_ADMIN");
    }

    public void addCountEvent() {
        countEvent++;
    }

    public Boolean isMatchingPasswords() {
        return this.password.equals(this.confirmPassword);
    }
}

```

```

@Entity
@NoArgsConstructor

```

					ІАЛЦ.467100.007 Д4	Акл.
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

@AllArgsConstructor
@Getter
@Setter
@ToString
public class Event {

    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;

    private String title;

    @Column(name = "start")
    private Date start;

    @Column(name = "stop")
    private Date end;

    @NotNull
    @ManyToOne
    private Room room;

    @NotNull
    @ManyToOne
    private Account account;

    public Event(String title, Date start, Date end, Room room, Account account) {
        super();
        this.title = title;
        this.start = start;
        this.end = end;
        this.room = room;
        this.account = account;
    }
}

@Entity
@NoArgsConstructor
@AllArgsConstructor
@Getter
@Setter
@ToString
public class Hostel {
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Id
    private Long id;

    @NotNull
    @Size(min = 3, max = 100, message = "Name must have at least 3 characters.")
    private String name;

    private String address;

    @NotNull
    @ManyToOne
    private University university;
}

@Entity

```

					ІАЛЦ.467100.007 Д4	Акл.
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

@AllArgsConstructor
@Getter
@Setter
@ToString
public class Room {
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Id
    private Long id;

    @NotNull
    @Size(min = 3, max = 100, message = "Name must have at least 3 characters.")
    private String name;

    private int floor = 0;

    private int capacity = 0;

    private String sportsEquipment = "NO";

    private String recreationArea = "NO";

    private String lunchZone = "NO";

    private String lighting = "NO";

    private String description = "NO";

    @ManyToOne
    @NotNull
    private Hostel hostel;
}

@Entity
@NoArgsConstructor
@AllArgsConstructor
@Getter
@Setter
@ToString
public class University {
    @GeneratedValue(strategy = GenerationType.AUTO)
    @Id
    private Long id;

    @NotNull
    @Size(min = 3, max = 100, message = "Name must have at least 3 characters.")
    private String name;
}

@Repository
public interface AccountRepository extends JpaRepository<Account, Serializable> {
    Account findOneByUsername(String name);

    Account findOneByEmail(String email);

    Account findOneByUsernameOrEmail(String username, String email);

    Account findOneByToken(String token);

    Account findOneByUniversity(University university);
}

```

					ІАЛЦ.467100.007 Д4	Акл.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

```

List<Account> findByUniversity(University university);

@Modifying
@Transactional
@Query("update Account a set a.email = :email, a.firstName = :firstName, "
+ "a.lastName = :lastName, a.address = :address, a.groupName =
:groupName, "
+ "a.role = :role, a.university = :university "
+ "where a.userName = :userName")
int updateUser(
    @Param("userName") String userName,
    @Param("email") String email,
    @Param("firstName") String firstName,
    @Param("lastName") String lastName,
    @Param("address") String address,
    @Param("groupName") String groupName,
    @Param("role") String role,
    @Param("university") University university);

@Modifying
@Transactional
@Query("update Account a set a.email = :email, a.firstName = :firstName, "
+ "a.lastName = :lastName, a.address = :address, a.groupName =
:groupName, "
+ "a.password = :password, a.role = :role, a.university = :university "
+ "where a.userName = :userName")
int updateUserWithPassword(
    @Param("userName") String userName,
    @Param("email") String email,
    @Param("firstName") String firstName,
    @Param("lastName") String lastName,
    @Param("address") String address,
    @Param("groupName") String groupName,
    @Param("password") String password,
    @Param("role") String role,
    @Param("university") University university);

@Modifying
@Transactional
@Query("update Account a set a.lastLogin = CURRENT_TIMESTAMP where a.userName =
?1")
int updateLastLogin(String userName);

@Modifying
@Transactional
@Query("update Account a set a.countEvent = :countEvent where a.userName =
:userName")
int updateCountEvent(
    @Param("userName") String userName,
    @Param("countEvent") int countEvent);
}

@Repository
public interface HostelRepository extends JpaRepository<Hostel, Serializable> {

    Hostel findById(Long id);

    Hostel findByName(String name);

    List<Hostel> findByUniversity(University university);

```

					ІАЛІЦ.467100.007 Д4	Акл.
Змн.	Анк.	№ докум.	Підпис	Дата		5

```

@Query("select un from Hostel un " +
      "where un.name = :name and un.university = :university")
Hostel findByNameAndUniversity(
    @Param("name") String name,
    @Param("university") University university);

@Modifying
@Transactional
@Query("update Hostel u set u.name = :name, u.address = :address, u.university =
:university "
      + "where u.id = :id")
int updateHostel(
    @Param("id") Long id,
    @Param("name") String name,
    @Param("address") String address,
    @Param("university") University university);
}

```

```

@Repository
public interface RoomRepository extends JpaRepository<Room, Serializable> {

```

```

    Room findById(Long id);

    Room findByName(String name);

    List<Room> findByHostel(Hostel hostel);

```

```

@Query("select ro from Room ro " +
      "where ro.name = :name and ro.hostel = :hostel")
Room findByNameAndHostel(
    @Param("name") String name,
    @Param("hostel") Hostel hostel);

```

```

@Modifying
@Transactional
@Query("update Room r set r.name = :name, r.floor = :floor, "
      + "r.capacity = :capacity, r.sportsEquipment = :sportsEquipment,
r.recreationArea = :recreationArea, r.lunchZone = :lunchZone, "
      + "r.lighting = :lighting, r.description = :description, r.hostel =
:hostel "
      + "where r.id = :id")
int updateRoom(
    @Param("id") Long id,
    @Param("name") String name,
    @Param("floor") int floor,
    @Param("capacity") int capacity,
    @Param("sportsEquipment") String sportsEquipment,
    @Param("recreationArea") String recreationArea,
    @Param("lunchZone") String lunchZone,
    @Param("lighting") String lighting,
    @Param("description") String description,
    @Param("hostel") Hostel hostel);
}

```

```

@Repository
public interface UniversityRepository extends JpaRepository<University, Serializable>
{

```

```

    University findById(Long id);

    University findByName(String name);

```

					ІАЛІЦ.467100.007 Д4	Акл.
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

@Modifying
@Transactional
@Query("update University o set o.name = :name "
      + "where o.id = :id")
int updateUniversity(
    @Param("id") Long id,
    @Param("name") String name);
}

jQuery(document).ready(function($){
    var formModal = $('#cd-user-modal'),
        formLogin = formModal.find('#cd-login'),
        formSignup = formModal.find('#cd-signup'),
        formForgotPassword = formModal.find('#cd-reset-password'),
        formModalTab = $('#cd-switcher'),
        tabLogin = formModalTab.children('li').eq(0).children('a'),
        tabSignup = formModalTab.children('li').eq(1).children('a'),
        forgotPasswordLink = formLogin.find('.cd-form-bottom-message a'),
        backToLoginLink = formForgotPassword.find('.cd-form-bottom-message a'),
        mainNav = $('#.main-nav');
    signInBtn = $('#.cd-btn.main-action');

    //open modal
    mainNav.on('click', function(event){
        $(event.target).is(mainNav) && mainNav.children('ul').toggleClass('is-
visible');
    });

    //open sign-up form
    mainNav.on('click', '.cd-signup', signup_selected);
    //open login-form form
    mainNav.on('click', '.cd-signin', login_selected);

    signInBtn.on('click', login_selected);

    //close modal
    formModal.on('click', function(event){
        if( $(event.target).is(formModal) || $(event.target).is('.cd-close-form') ) {
            formModal.removeClass('is-visible');
        }
    });
    //close modal when clicking the esc keyboard button
    $(document).keyup(function(event){
        if(event.which=='27'){
            formModal.removeClass('is-visible');
        }
    });

    //switch from a tab to another
    formModalTab.on('click', function(event) {
        event.preventDefault();
        ( $(event.target).is( tabLogin ) ) ? login_selected() : signup_selected();
    });

    //hide or show password
    $('#.hide-password').on('click', function(){
        var togglePass= $(this),
            passwordField = togglePass.prev('input');

```

					ІАЛЦ.467100.007 Д4	Акл.
Змн.	Адк.	№ докум.	Підпис	Дата		7

```

        ( 'password' == passwordField.attr('type') ) ? passwordField.attr('type',
'text') : passwordField.attr('type', 'password');
        ( 'Hide' == togglePass.text() ) ? togglePass.text('Show') :
togglePass.text('Hide');
        //focus and move cursor to the end of input field
        passwordField.setCursorAtEnd();
    });

    //show forgot-password form
    forgotPasswordLink.on('click', function(event){
        event.preventDefault();
        forgot_password_selected();
    });

    //back to login from the forgot-password form
    backToLoginLink.on('click', function(event){
        event.preventDefault();
        login_selected();
    });

    function login_selected(){
        mainNav.children('ul').removeClass('is-visible');
        formModal.addClass('is-visible');
        formLogin.addClass('is-selected');
        formSignup.removeClass('is-selected');
        formForgotPassword.removeClass('is-selected');
        tabLogin.addClass('selected');
        tabSignup.removeClass('selected');
    }

    function signup_selected(){
        mainNav.children('ul').removeClass('is-visible');
        formModal.addClass('is-visible');
        formLogin.removeClass('is-selected');
        formSignup.addClass('is-selected');
        formForgotPassword.removeClass('is-selected');
        tabLogin.removeClass('selected');
        tabSignup.addClass('selected');
    }

    function forgot_password_selected(){
        formLogin.removeClass('is-selected');
        formSignup.removeClass('is-selected');
        formForgotPassword.addClass('is-selected');
    }
    if(!Modernizr.input.placeholder){
        $('[placeholder]').focus(function() {
            var input = $(this);
            if (input.val() == input.attr('placeholder')) {
                input.val('');
            }
        }).blur(function() {
            var input = $(this);
            if (input.val() == '' || input.val() == input.attr('placeholder')) {
                input.val(input.attr('placeholder'));
            }
        }).blur();
        $('[placeholder]').parents('form').submit(function() {
            $(this).find('[placeholder]').each(function() {
                var input = $(this);
                if (input.val() == input.attr('placeholder')) {
                    input.val('');
                }
            });
        });
    }

```

						ІАЛПЦ.467100.007 Д4	Акт.
Змн.	Анк.	№ докум.	Підпис	Дата			8

```

    }
  })
});

jQuery.fn.putCursorAtEnd = function() {
  return this.each(function() {
    // If this function exists...
    if (this.setSelectionRange) {
      var len = $(this).val().length * 2;
      this.focus();
      this.setSelectionRange(len, len);
    } else {
      $(this).val($(this).val());
    }
  });
};
/* -----
videointro bg
----- */
jQuery(document).ready(function($) {
  //this is used for the video effect only
  if ( $('.cd-bg-video-wrapper').length > 0 ) {
    var videoWrapper = $('.cd-bg-video-wrapper'),
        mq = window.getComputedStyle(document.querySelector('.cd-bg-video-wrapper'),
        '::after').getPropertyValue('content').replace(/"/g, "").replace(/'/g, "");
    if ( mq == 'desktop' ) {
      // we are not on a mobile device
      var videoUrl = videoWrapper.data('video'),
          video = $('<video loop><source src="'+videoUrl+'.mp4" type="video/mp4"
/><source src="'+videoUrl+'.webm" type="video/webm" /></video>');
      video.appendTo(videoWrapper);
      video.get(0).play();
    }
  }
});
/* -----
Login error popup
----- */
jQuery(document).ready(function($) {
  //open popup
  if (location.search == "?error1"){
    $('.cd-popup').addClass('is-visible');
  };

  //close popup
  $('.cd-popup').on('click', function(event) {
    if ( $(event.target).is('.cd-popup-close') || $(event.target).is('.cd-popup') )
    {
      event.preventDefault();
      $(this).removeClass('is-visible');
    }
  });
});

```

					ІАЛЦ.467100.007 Д4	Акл.
Змн.	Адк.	№ докум.	Підпис	Дата		9

```

//close popup when clicking the esc keyboard button
$(document).keyup(function(event){
    if(event.which=='27'){
        $('.cd-popup').removeClass('is-visible');
    }
});
});

/* -----

navigation layout

----- */
jQuery(document).ready(function(){
    //cache DOM elements
    var mainContent = $('.cd-main-content'),
        header = $('.cd-main-header'),
        sidebar = $('.cd-side-nav'),
        sidebarTrigger = $('.cd-nav-trigger'),
        topNavigation = $('.cd-top-nav'),
        searchForm = $('.cd-search'),
        accountInfo = $('.account'),
        dropdownMenuElements = [$('.account'), $('.users'), $('.universities'),
$('.hostels'), $('.rooms')];

    //on resize, move search and top nav position according to window width
    var resizing = false;
    moveNavigation();
    addOpenDropdownMenuHandlers(dropdownMenuElements);

    $(window).on('resize', function(){
        if (!resizing) {
            (!window.requestAnimationFrame) ? setTimeout(moveNavigation, 300) :
window.requestAnimationFrame(moveNavigation);
            resizing = true;
        }
    });

    //on window scrolling - fix sidebar nav
    var scrolling = false;
    checkScrollbarPosition();
    $(window).on('scroll', function(){
        if (!scrolling) {
            (!window.requestAnimationFrame) ? setTimeout(checkScrollbarPosition, 300) :
window.requestAnimationFrame(checkScrollbarPosition);
            scrolling = true;
        }
    });

    //mobile only - open sidebar when user clicks the hamburger menu
    sidebarTrigger.on('click', function(event){
        event.preventDefault();
        $('[sidebar, sidebarTrigger]').toggleClass('nav-is-visible');
    });

    //click on item and show submenu
    $('.has-children > a').on('click', function(event){
        var mq = checkMQ(),
            selectedItem = $(this);
        if( mq == 'mobile' || mq == 'tablet' ) {
            event.preventDefault();
        }
    });

```

					ІАЛЦ.467100.007 Д4	Акл.
Змн.	Арк.	№ докум.	Підпис	Дата		10

```

        selectedItem.parent('li').removeClass('selected');
    } else {
        sidebar.find('.has-children.selected').removeClass('selected');
        accountInfo.removeClass('selected');
        selectedItem.parent('li').addClass('selected');
    }
}
});

function addOpenDropdownMenuHandler(item) {
    item.children('a').on('click', function(event) {
        var mq = checkMQ(),
            selectedItem = $(this);
        if( mq == 'desktop' ) {
            event.preventDefault();
            item.toggleClass('selected');
            sidebar.find('.has-children.selected').removeClass('selected');
        }
    });
}

function addOpenDropdownMenuHandlers(items) {
    items.forEach(item => {
        addOpenDropdownMenuHandler(item);
    })
}

$(document).on('click', function(event){
    if( !$(event.target).is('.has-children a') ) {
        sidebar.find('.has-children.selected').removeClass('selected');
        accountInfo.removeClass('selected');
    }
});

//on desktop - differentiate between a user trying to hover over a dropdown item
vs trying to navigate into a submenu's contents
sidebar.children('ul').menuAim({
    activate: function(row) {
        $(row).addClass('hover');
    },
    deactivate: function(row) {
        $(row).removeClass('hover');
    },
    exitMenu: function() {
        sidebar.find('.hover').removeClass('hover');
        return true;
    },
    submenuSelector: ".has-children",
});

function checkMQ() {
    //check if mobile or desktop device
    return window.getComputedStyle(document.querySelector('.cd-main-content'),
    '::before').getPropertyValue('content').replace(/'/g, "").replace(/"/g, "");
}

function moveNavigation(){
    var mq = checkMQ();

    if ( mq == 'mobile' && topNavigation.parents('.cd-side-nav').length == 0 ) {

```

					ІАЛЦ.467100.007 Д4	Акл.
						11
Змн.	Апк.	№ докум.	Підпис	Дата		

```

        detachElements();
        topNavigation.appendTo(sidebar);
        searchForm.removeClass('is-hidden').prependTo(sidebar);
    } else if ( ( mq == 'tablet' || mq == 'desktop' ) &&
topNavigation.parents('.cd-side-nav').length > 0 ) {
        detachElements();
        searchForm.insertAfter(header.find('.cd-logo'));
        topNavigation.appendTo(header.find('.cd-nav'));
    }
    checkSelected(mq);
    resizing = false;
}

function detachElements() {
    topNavigation.detach();
    searchForm.detach();
}

function checkSelected(mq) {
    //on desktop, remove selected class from items selected on mobile/tablet
    version
    if( mq == 'desktop' ) $('<span>.has-children.selected</span>').removeClass('selected');
}

function checkScrollbarPosition() {
    var mq = checkMQ();

    if( mq != 'mobile' ) {
        var sidebarHeight = sidebar.outerHeight(),
            windowHeight = $(window).height(),
            mainContentHeight = mainContent.outerHeight(),
            scrollTop = $(window).scrollTop();

        ( ( scrollTop + windowHeight > sidebarHeight ) && ( mainContentHeight -
sidebarHeight != 0 ) ) ? sidebar.addClass('is-fixed').css('bottom', 0) :
sidebar.removeClass('is-fixed').attr('style', '');
    }
    scrolling = false;
}
});

$content_wrapper.on('click', function(event){
    if( !$(event.target).is('#cd-menu-trigger, #cd-menu-trigger span') ) {
        $lateral_menu_trigger.removeClass('is-clicked');
        $navigation.removeClass('lateral-menu-is-open');
        $content_wrapper.removeClass('lateral-menu-is-open').one('webkitTransitionEnd
otransitionend oTransitionEnd msTransitionEnd transitionend', function(){
            $('body').removeClass('overflow-hidden');
        });
        $('#cd-lateral-nav').removeClass('lateral-menu-is-open');
        //check if transitions are not supported
        if($('html').hasClass('no-csstransitions')) {
            $('body').removeClass('overflow-hidden');
        }
    }
});
});

```

					ІАЛІЦ.467100.007 Д4	Акл.
Змн.	Анк.	№ докум.	Підпис	Дата		12