

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук (або центр післядипломної освіти, або навчально-науковий центр заочної форми навчання)
(повна назва)

Кафедра _____ програмної інженерії
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА Пояснювальна записка

рівень вищої освіти _____ перший (бакалаврський)

Програмна система для рекомендації та підбору актуальних культурних подій. Mobile Application.
(тема)

Виконав:

студент 4 курсу, групи ПЗП-20-2

Дегтяр В.Е.

(прізвище, ініціали)

Спеціальність 121 – Інженерія програмного забезпечення

(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Програмна інженерія

(повна назва освітньої програми)

Керівник доц. кафедри ПІ Побіженко І.О.

(посада, прізвище, ініціали)

Допускається до захисту
Зав. кафедри

(підпис)

З.В.Дудар

(прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет _____ комп'ютерних наук _____
 Кафедра _____ програмної інженерії _____
 Рівень вищої освіти _____ перший (бакалаврський) _____
 Спеціальність _____ 121 – Інженерія програмного забезпечення _____
 Тип програми _____ Освітньо-професійна _____
 Освітня програма _____ Програмна Інженерія _____
 (шифр і назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

« ____ » _____ 2024 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові _____ Дегтяр Владиславу Едуардовичу _____
 (прізвище, ім'я, по батькові)

1. Тема роботи _____ Програмна система для рекомендації та підбору
 актуальних культурних подій. Mobile Application. _____

Затверджена наказом по університету від _____ 20.05. 2024р. № 471 Ст _____

2. Термін подання студентом роботи до екзаменаційної комісії _____ 17.06.2024 _____

3. Вихідні дані до роботи Розробити клієнтську частину з використанням мови програмування Swift та UI фреймворку SwiftUI для програмної системи для рекомендацій та підбору актуальних культурних подій.

4. Перелік питань, що потрібно опрацювати в роботі

Вступ, аналіз предметної галузі, формування вимог до програмної системи, архітектура та проектування програмного забезпечення, опис прийнятих програмних рішень, тестування розробленого програмного забезпечення, висновки, додатки.

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Аналіз предметної галузі	08.04.2024	<i>виконано</i>
2	Створення специфікації ПЗ	12.04.2024	<i>виконано</i>
3	Проектування ПЗ	15.04.2024	<i>виконано</i>
4	Розробка ПЗ	10.05.2024	<i>виконано</i>
5	Тестування ПЗ	30.05.2024	<i>виконано</i>
6	Оформлення пояснювальної записки	01.06.2024	<i>виконано</i>
7	Підготовка презентації та доповіді	11.06.2024	<i>виконано</i>
8	Попередній захист	15.06.2024	<i>виконано</i>
9	Нормоконтроль, рецензування	16.06.2024	<i>виконано</i>
10	Здача роботи у електронний архів	17.06.2024	<i>виконано</i>
11	Допуск до захисту у зав. кафедри	20.06.2024	<i>виконано</i>

Дата видачі завдання 8 квітня 2024р.

Студент (ка) 
(підпис)

Дегтяр В.Е.

Керівник роботи _____
(підпис)

доц. кафедри ПІ Побіженко І.О.
(посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до кваліфікаційної роботи бакалавра: 77 с., 17 рис., 10 джерел.

КЛІЄНТСЬКА ЧАСТИНА, ПРОГРАМНА СИСТЕМА, СИСТЕМА ДЛЯ РЕКОМЕНДАЦІЇ КУЛЬТУРНИХ ПОДІЙ, SWIFT, SWIFTUI.

Об'єкт розробки – мобільний додаток клієнтської частини для програмної системи для рекомендації та підбору актуальних культурних подій.

Мета розробки – розробка програмної системи для рекомендації та підбору актуальних культурних подій.

Метод рішення – середовище розробки XCode, мова програмування Swift та фреймворк SwiftUI.

У результаті розробки створено клієнтську частину програмної системи для рекомендації та підбору актуальних культурних подій.

CULTURAL EVENTS RECOMMENDATION SYSTEM, SOFTWARE, FRONTEND, SWIFT, SWIFTUI.

The object of development is the client part mobile application for the software system for recommending and selecting relevant cultural events.

The purpose of the development is to develop a software system for recommending and selecting relevant cultural events.

The solution method is the XCode development environment, the Swift programming language, and the SwiftUI framework.

As a result of the development, the client part of the software system was created for recommending and selecting relevant cultural events.

Я, Дегтяр Владислав Едуардович, студент гр. ПЗПІ-20-2, здобувач вищої освіти на першому (бакалаврському) рівні кафедри «Програмна інженерія», заявляю: моя кваліфікаційна робота на тему «Програмна система для рекомендації та підбору актуальних культурних подій. Mobile Application.», що буде представлена в екзаменаційну комісію для публічного захисту, виконана самостійно, в ній не містяться елементи плагіату. Всі запозичення з друкованих та електронних джерел мають відповідні посилання.

Я ознайомлений з діючим положенням «Про протидію академічному плагіату в ХНУРЕ», згідно з яким виявлення плагіату є підставою для відмови в допуску кваліфікаційної роботи до захисту та застосування дисциплінарних заходів.

ЗМІСТ

ВСТУП	8
1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ	9
1.1 Аналіз предметної галузі.....	9
2.2 Виявлення проблем та актуалізація рішень.....	12
2.3 Постановка задачі.....	13
2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ	15
3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ	17
3.1 UML проєктування ПЗ.....	17
3.2 Проєктування архітектури ПЗ.....	19
3.3 Проєктування бази даних.....	23
3.4 Огляд алгоритмів та методів.....	25
3.5 Створення UI/UX.....	26
4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ	30
4.1 Інтеграція з Apple Maps для відображення подій.....	30
4.2 Інтеграція з Azure й OpenAI API для ШІ чат-бота.....	32
4.3 Управління станом клієнтської частини за допомогою @State і @ObservedObject.....	33
4.4 Вікно акаунту користувача для зміни аватару, редагування особистих даних та зміни локалізації.	34
5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	36
5.1 План тестування.....	36
5.2 Тест-кейси.....	36
ВИСНОВКИ	40
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	42
ДОДАТОК А Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ.....	43
ДОДАТОК Б Слайди презентації.....	45
ДОДАТОК В Специфікація вимог до програмного продукту.....	51
ВСТУП	53
1.1 Огляд продукту.....	53
1.2 Мета.....	54
1.3 Межі.....	54
1.4 Посилання.....	55
1.5 Означення та аббревіатури.....	56
2. ЗАГАЛЬНИЙ ОПИС	58

2.1 Перспективи продукту	58
2.2 Функції продукту	59
2.3 Характеристика користувачів.....	59
2.4 Загальні обмеження	59
2.5 Припущення й залежності.....	60
3 КОНКРЕТНІ ВИМОГИ.....	61
3.1 Вимоги до зовнішніх інтерфейсів.....	61
3.1.1 Інтерфейс користувача.....	61
3.1.2 Апаратний інтерфейс.....	61
3.1.3 Програмний інтерфейс	62
3.1.4 Комунікаційний протокол	62
3.1.5 Обмеження пам'яті	63
3.1.6 Операції	63
3.2 Атрибути програмного продукту	63
3.2.1 Надійність.....	63
3.2.2 Доступність	64
3.2.3 Безпека.....	64
3.2.4 Супроводжуваність.....	65
3.2.5 Переносимість.....	65
3.2.6 Продуктивність	65
3.2.7 Вимоги бази даних.....	65
ДОДАТОК Г Тези ХХVІІІ міжнародного молодіжного форуму «Радіоелектроніка та молодь у ХХІ столітті».....	67
ДОДАТОК Д Код для управління авторизацією користувача	73

ВСТУП

Розвиток цифрових технологій суттєво змінює способи взаємодії з культурою та дозвіллям. В умовах глобалізації та збільшення мобільності населення зростає потреба в персоналізованих і розумних системах рекомендацій культурних заходів. "Eventify" – це інноваційний програмний продукт, який розв'язує проблему вибору культурних подій, враховуючи індивідуальні вподобання та геолокацію користувачів.

Збільшення кількості культурних заходів, таких як концерти, виставки, ярмарки та фестивалі, викликає потребу в системі, що не лише інформує про події, а й допомагає користувачам знайти найбільш відповідні заходи, базуючись на їхніх минулих вподобаннях та поточному місцеперебуванні. "Eventify" використовує передові алгоритми машинного навчання для створення персоналізованих рекомендацій, що робить процес вибору подій зручним та інтуїтивно зрозумілим.

Система "Eventify" розроблена для задоволення потреб сучасної аудиторії, яка цінує якість та персоналізацію культурного досвіду. Використовуючи новітні технологічні рішення, сервіс забезпечує швидке інформування про заходи, їх фільтрацію та пошук, зручне бронювання квитків і взаємодію з ШІ-асистентом для вибору подій.

Основними завданнями "Eventify" є забезпечення високої точності та відповідності рекомендацій, ефективне управління даними користувачів, а також відповідність сучасним стандартам безпеки та конфіденційності. Розробка цієї системи має важливе значення для культурної інтеграції та розвитку, враховуючи індивідуальні потреби користувачів у динамічному світі.

Отже, "Eventify" сприяє підвищенню доступності та задоволення від культурних заходів, роблячи процес вибору подій максимально персоналізованим, зручним та швидким.

1 АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметної галузі

Сфера програмної системи "Eventify" охоплює індустрію розваг та культурних заходів, що включає організацію та пошук різноманітних подій: концерти, виставки, ярмарки та фестивалі. Культурні заходи відіграють важливу роль у соціальній взаємодії та особистісному розвитку, надаючи платформу для вираження культурної ідентичності та сприяючи міжкультурному обміну. Ця галузь характеризується широким розподілом подій за географічною локалізацією, великою кількістю учасників та різноманітним інтересом аудиторії.

Учасники культурних заходів належать до різних вікових та соціальних груп, що шукають розваги, які відповідають їхнім інтересам та географічному розташуванню. Однак, вони зіштовхуються з проблемою перевищення інформаційного потоку та вибору серед великої кількості доступних подій. Це викликає потребу в розробці ефективного інструменту, який допомагав би у фільтрації, рекомендації та персоналізації культурних заходів для користувачів.

Сфера культурних заходів включає різноманітні аспекти, від планування та організації до маркетингу та продажу квитків. Організатори подій постійно шукають способи залучення аудиторії, поліпшення взаємодії з відвідувачами та підвищення задоволення учасників.

В сучасному світі росте потреба в персоналізації досвіду, оскільки споживачі бажають відвідувати події, що відповідають їхнім інтересам та вподобанням. Використання технологій для аналізу даних про попередні бажання, вподобання та геолокацію користувачів дозволяє створити рекомендаційні системи, які надають високу релевантність та персоналізацію інформації про заходи.

Розвиток технологій, зокрема у сферах мобільних додатків, соціальних мереж та інтерактивних платформ, відкриває нові можливості для організації та просування культурних заходів. "Eventify" поєднає ці технології для створення зручної, інтуїтивно зрозумілої та доступної системи підбору подій.

На ринку існує багато сервісів для бронювання та придбання квитків, але лише незначна кількість з них забезпечує глибоку персоналізацію та інтеграцію з інтерактивними картами та штучним інтелектом для покращення користувацького досвіду.

Для аналізу було обрано афішу kontramarka.ua. На цьому сайті присутня можливість для перегляду подій, замовлення квитків для концертів, театру, фестивалів але немає можливості подивитись усі ці заходи на мапі (див. рис 1.1).

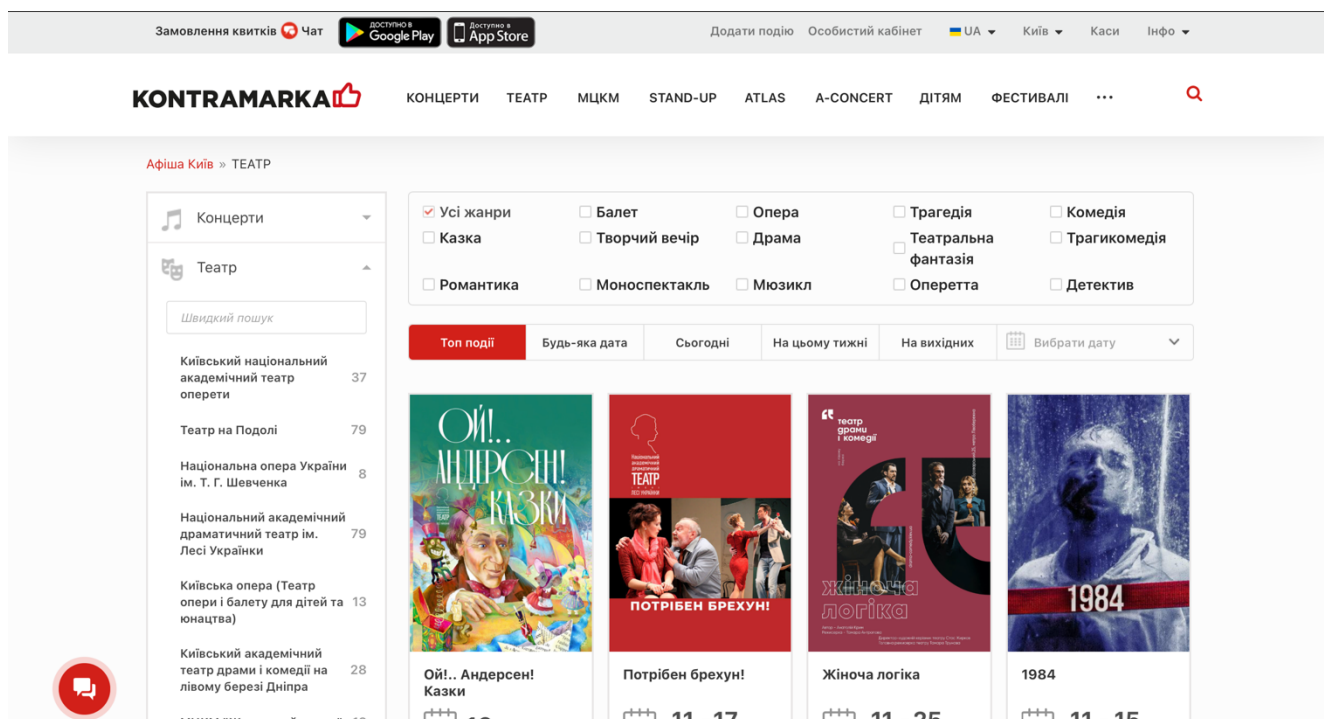


Рисунок 1.1 – Афіша з квитками на події Kontramarka.ua (за даними [1])

Іншою аналогічною системою є афіша подій KARABAS. На сайті Karabas.com користувачі можуть переглядати інформацію про майбутні заходи, такі як концерти, театральні вистави, клубні вечірки та фестивалі. Відвідувачі також мають можливість замовити квитки на ці події. Проте, як і в першому випадку, відсутня функція перегляду всіх заходів на карті або інтерактивного інструменту для рекомендацій (див. рис. 1.2)

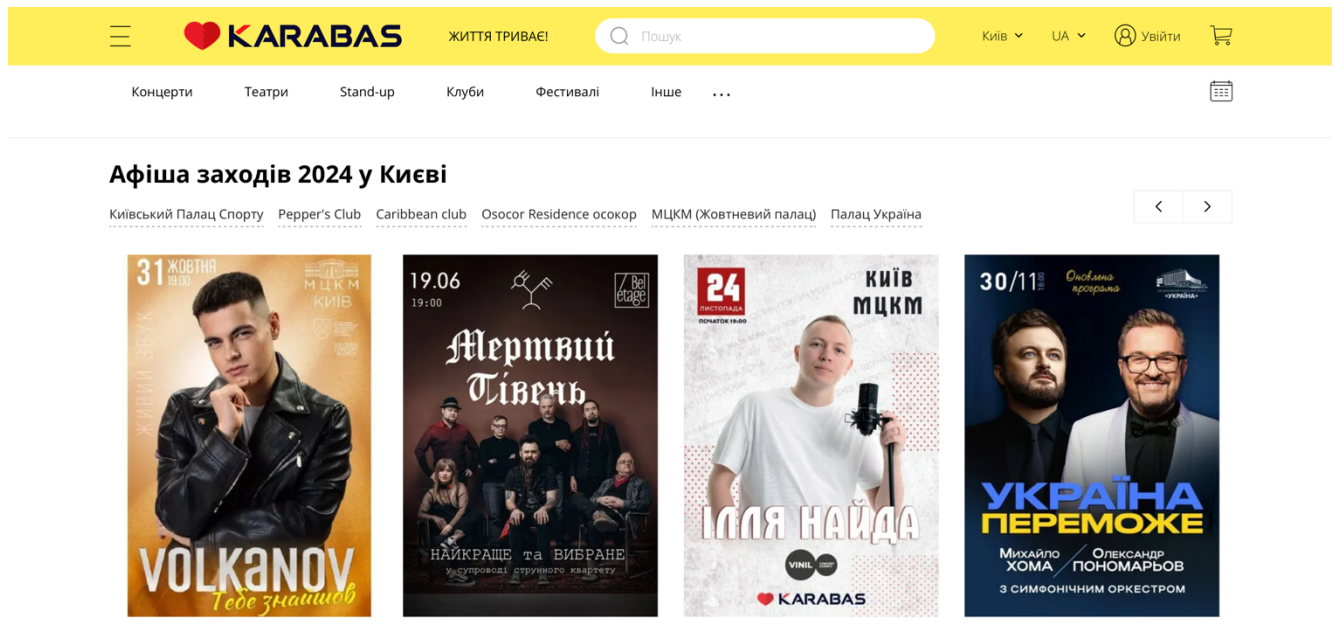


Рисунок 1.2 – Афіша подій karabas.com (за даними [2])

Сучасна індустрія культурних подій потребує цифрових рішень, які допомагають користувачам орієнтуватися у великій кількості заходів, підбираючи найбільш відповідні на основі їхніх уподобань та попереднього досвіду. "Eventify" відповідає на висхідний попит на індивідуалізований культурний досвід, допомагаючи користувачам знаходити події, які найбільше відповідають їхнім потребам, підвищуючи загальне задоволення та залученість аудиторії.

Основні функції "Eventify" включають:

- **Замовлення квитків:** Інтеграція з платформами організаторів заходів для зручного бронювання квитків та запрошень на різноманітні події безпосередньо через інтерфейс системи.
- **Перегляд подій на карті:** Можливість перегляду подій на інтерактивній карті, що полегшує вибір заходів залежно від їхнього місцезнаходження.
- **Спілкування з ШІ-асистентом:** Інтеграція штучного інтелекту для допомоги у виборі заходів, відповіді на запитання та надання додаткової інформації про події.

2.2 Виявлення проблем та актуалізація рішень

У системі "Eventify", яка призначена для рекомендації культурних подій, можна реалізувати ряд функцій, спрямованих на розв'язання проблем вибору культурних заходів та оптимізацію рішень. Ось декілька прикладів таких можливостей.

- Персоналізація досвіду користувача: Система може збирати дані про попередні відвідування та вподобання користувачів, щоб краще підбирати рекомендації та забезпечувати індивідуальний підхід.
- Аналіз та фільтрація інформаційного шуму: Розширені алгоритми фільтрації можуть допомогти обирати релевантні події серед великого обсягу інформації, що робить процес вибору зручнішим для користувачів.
- Інтеграція з інтерактивними картами для локалізації заходів: Це полегшує користувачам планування часу та маршрутів для відвідування подій.
- Виявлення та реагування на зміни уподобань користувачів: Система може адаптуватися до змін у поведінці користувачів та пропонувати нові види заходів відповідно до їх оновлених інтересів.
- Підтримка в реальному часі через ШІ-асистента: ШІ-асистент може надавати оперативну підтримку користувачам, відповідаючи на запитання та рекомендуючи заходи на основі актуальних запитів та географічного місця.

Ці функції спрямовані на розв'язання конкретних проблем, які виникають у користувачів при виборі культурних заходів, та допомагають покращити користувацький досвід.

Щоб покращити взаємодію користувачів із сервісом та забезпечити більш ефективний вибір культурних заходів, можна вжити такі кроки для актуалізації рішень у системі "Eventify":

- Збір та аналіз даних користувача: Розробка інструментів для збору інформації про попередні пошуки, відвідування та відгуки користувачів, що дозволить створювати детальні профілі уподобань.

- Розвиток персоналізованих рекомендацій: Використання машинного навчання для аналізу зібраних даних та автоматизації процесу рекомендацій, що базуються на індивідуальних інтересах і місцеперебуванні користувачів.

- Оптимізація інтерфейсу користувача: Розробка зручного інтерфейсу з легким доступом до функцій фільтрації, пошуку та перегляду заходів на карті.

- Інтеграція з інтерактивними картами: Впровадження візуалізації подій на мапі для полегшення планування відвідувань, включаючи маршрутизацію та інформацію про доступність.

- Розробка системи сповіщень: Введення сповіщень, які інформують користувачів про нові заходи або зміни в уже запланованих подіях, заснованих на їхніх вподобаннях.

- Застосування ШІ-асистента для підтримки у реальному часі: Інтеграція розмовного ШІ-асистента, який може миттєво відповідати на запитання користувачів, допомагати у виборі заходів та розв'язувати проблеми.

Ці можливості сприяють кращому розумінню уподобань користувачів, покращенню персоналізації рекомендацій та забезпеченню зручного та ефективного користувацького досвіду.

2.3 Постановка задачі

Завдання цієї роботи полягає у розробці програмної системи "Eventify", призначеної для рекомендації та персоналізації культурних заходів. Система має забезпечувати ефективне збирання, зберігання, обробку та аналіз даних про інтереси та поведінку користувачів у контексті відвідування подій.

Основні завдання роботи включають:

- Автоматичне збирання даних: Система повинна автоматично збирати дані про вподобання користувачів, їх геолокації та історію відвідувань заходів.

- Розробка надійної бази даних: Необхідно створити базу даних для ефективного збереження великого обсягу інформації про користувачів та заходи.

- Обробка зібраних даних: Система повинна аналізувати зібрані дані для визначення відповідності заходів уподобанням користувачів.

- Розробка алгоритмів персоналізації: Створення алгоритмів, які враховують історичні дані та поточні запити користувачів для персоналізації рекомендацій.

- Інтеграція з інтерактивними картами: Впровадження функціонала для візуалізації заходів на мапі, що полегшує планування відвідувань.

- Інтеграція ШІ-асистента: Включення ШІ-асистента, який у реальному часі консультує користувачів, відповідає на їхні запитання та допомагає у виборі заходів.

- Заходи захисту даних: Імплементация механізмів захисту даних для запобігання несанкціонованому доступу та забезпечення конфіденційності інформації.

Метою роботи є створення ефективного та надійного інструменту, який дозволить користувачам знаходити культурні заходи, що відповідають їхнім інтересам та потребам, підвищуючи загальне задоволення та взаємодію з культурним контентом.

2 ФОРМУВАННЯ ВИМОГ ДО ПРОГРАМНОЇ СИСТЕМИ

Мова програмування та технології:

- Мобільний додаток реалізовано за допомогою мови Swift[3];
- Використання SwiftUI[4] для розробки інтерфейсу користувача.

Взаємодія з серверною частиною:

- Реалізація клієнтського API[5] для забезпечення комунікації між мобільним додатком та сервером;
- Реалізація обробників запитів, які взаємодіють з бекендом через REST, а також обробка відповідей для оновлення інтерфейсу відповідно до отриманих даних.

Функціональність мобільного додатку:

- Створення можливості логіну та реєстрації з вказуванням особистих даних та уподобань для подальшого формування рекомендацій.
- Надання динамічно оновлюваної мапи поточних подій біля користувача додатка відносно його геолокації.
- Список подій та інформацію про кожне з подій: час початку, тривалість, вікові обмеження, вартість квитків тощо.
- Надання можливості спілкування з ШІ-асистентом в окремому вікні чату.
- Можливість налаштування профілю: аватар, вподобання, мова тощо.

Безпека та конфіденційність:

- Забезпечення безпеки даних користувачів та виконання регулятивних вимог щодо обробки особистої інформації.
- Імплементация безпечних методів авторизації для доступу до функціональності додатка, що дозволяє забезпечити конфіденційність даних користувачів.

Користувацький інтерфейс:

- Розробка інтуїтивно зрозумілого інтерфейсу, який адаптується під різні моделі пристроїв;

- Імплементация функціональності пошуку, фільтрації та сортування заходів за датою завдяки компонентам UI бібліотеки.

Обробка даних та інтерактивність:

- Реалізація мапи з інформацією та типом події для більш інтерактивного та зручного способу взаємодії з додатком.

- Реалізація сторінок з детальною інформацією: Створення сторінок з докладним описом культурних заходів, включаючи описи, фотографії та інтерактивні елементи для бронювання квитків і відзначення заходів як улюблених.

Продуктивність та масштабованість:

- Оптимізація завантаження ресурсів: Використання лінивого завантаження компонентів для покращення швидкодії та ефективності системи.

Документація та тестування:

- Розробка документації: Створення документації для опису компонентів додатка та їхньої взаємодії.

- Тестування: Написання юніт-тестів та тестів "від кінця до кінця" для перевірки функціональності інтерфейсу та взаємодії з бекендом.

3 АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ

3.1 UML проєктування ПЗ

Діаграма варіантів використання (Use Case Diagram) є ключовою UML діаграмою, яка моделює функціональні вимоги системи з огляду на зовнішніх акторів. Вона допомагає визначити основні взаємодії між акторами та системою, виокремлюючи можливості системи, які повинні бути втілені.

У контексті системи "Eventify" діаграми варіантів використання стають корисними для уточнення потреб користувачів та опису функціональності системи. Ось кілька прикладів таких діаграм, що демонструють взаємодію різних типів користувачів із системою (див. рис. 3.1, 3.2, 3.3).

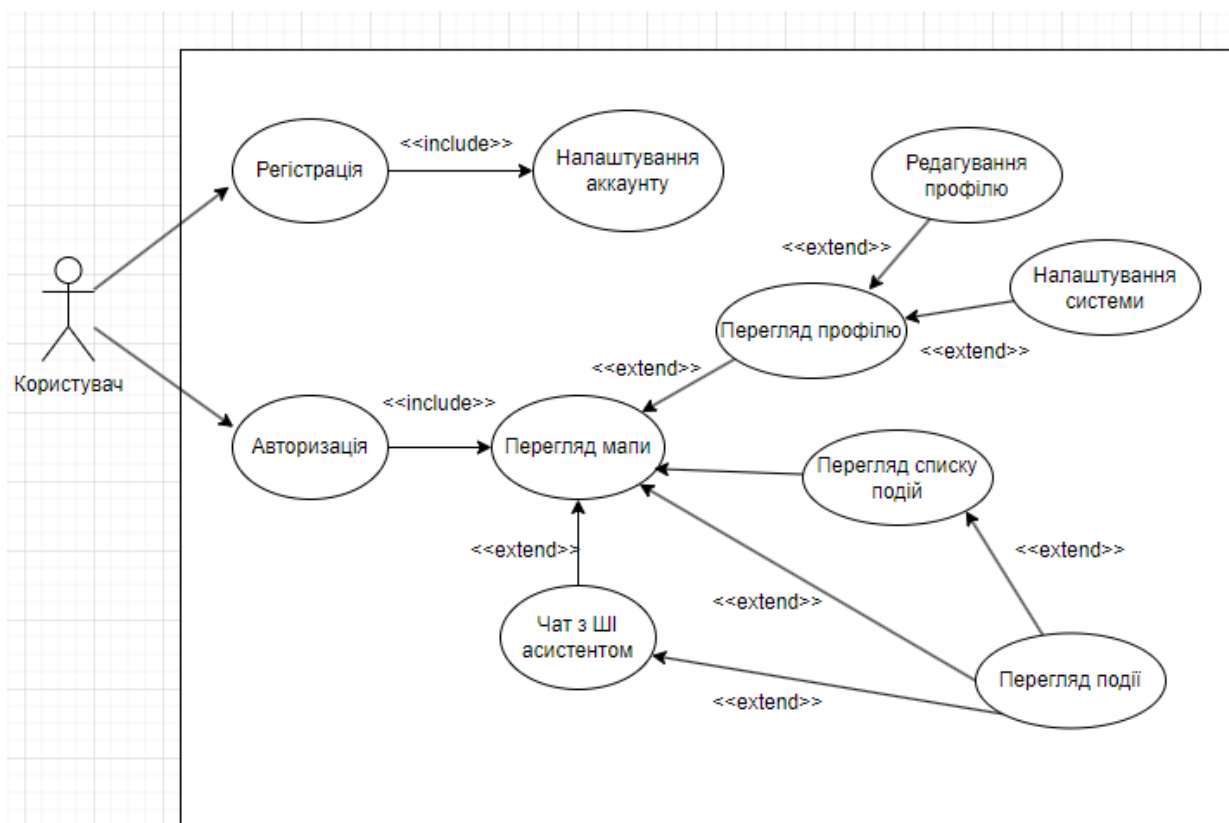


Рисунок 3.1 – Use-Case діаграма для Користувача (рисунок виконано самостійно)

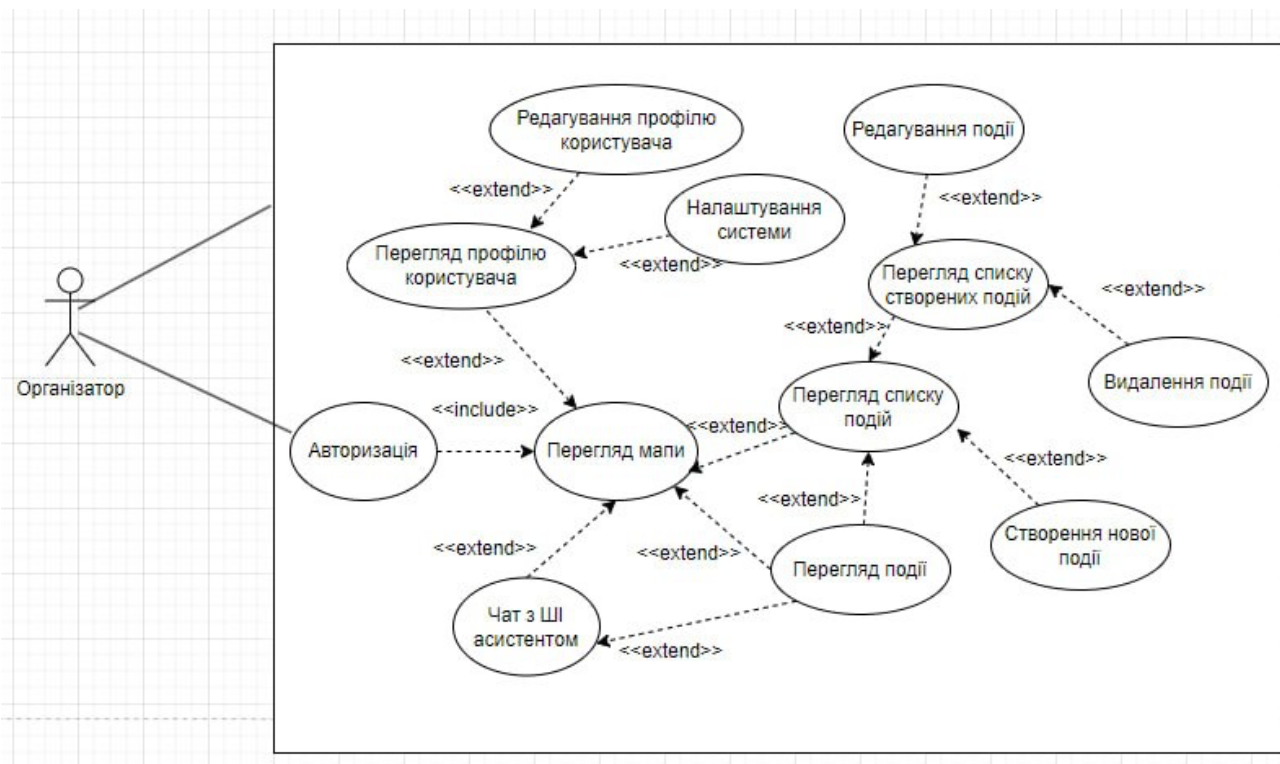


Рисунок 3.2 – Use-Case діаграма для Організатора (рисунок виконано самостійно)

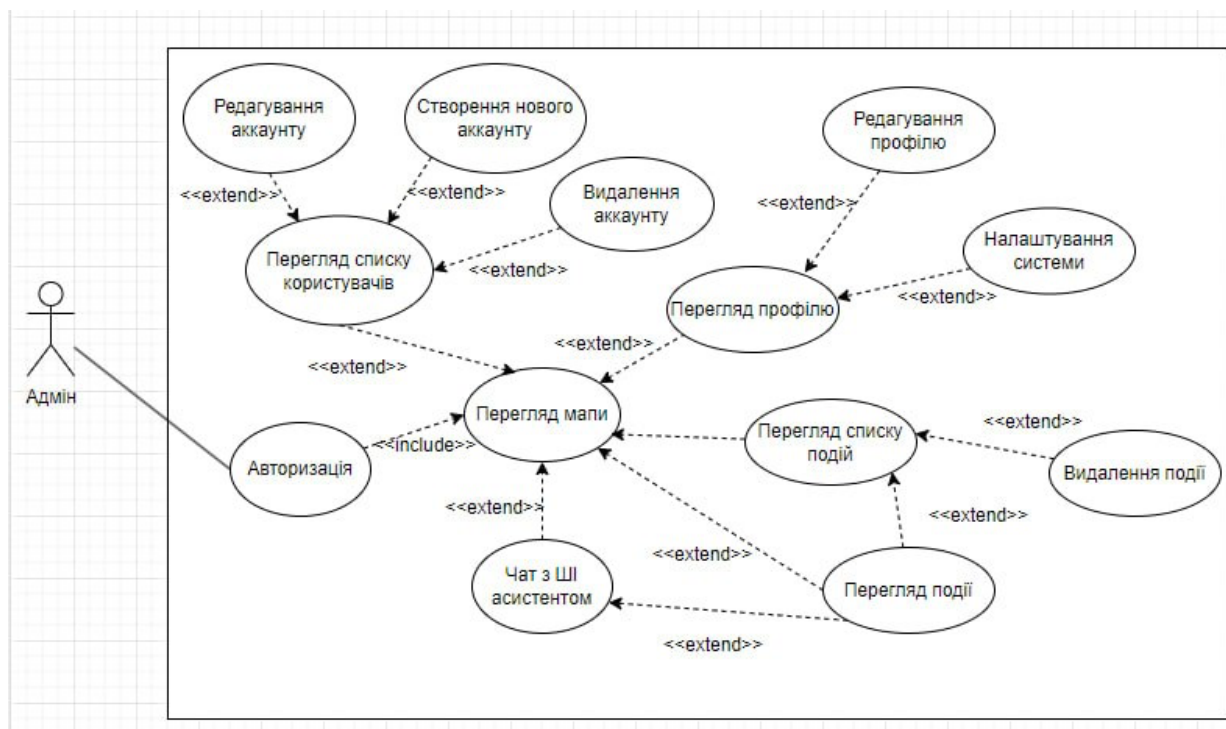


Рисунок 3.3 – Use-Case діаграма для Адміністратора (рисунок виконано самостійно)

Розглянемо акторів програмної системи “Eventify” та їхні варіанти використання.

– Користувачі є звичайними відвідувачами, які шукають інформацію про події та взаємодіють із системою для планування дозвілля. Вони можуть реєструватися та авторизуватися, переглядати мапу подій для зручного вибору та навігації, спілкуватися з ШІ-асистентом для отримання допомоги або порад, переглядати та управляти своїм профілем, а також переглядати список подій, фільтрованих за їхніми критеріями.

– Організатори є представниками, які організують заходи та використовують систему для управління подіями. Вони можуть створювати нові події та редагувати інформацію про вже виставлені заходи.

– Адміністратори керують системою, забезпечують її стабільність та управління контентом. Вони мають можливість управляти користувацькими обліковими записами, додавати, видаляти та модифікувати акаунти, а також розглядати та затверджувати події, створені організаторами.

3.2 Проєктування архітектури ПЗ

Клієнт-серверна архітектура є фундаментальною для розподілу функцій між клієнтською (frontend та mobile частинами) та серверною (backend) складовими системи "Eventify". Взаємодія між цими компонентами відбувається за допомогою мережевих протоколів, що надає користувачам можливість взаємодії з веб-інтерфейсом та мобільними додатками. У рамках архітектури також існує інтегрований ШІ компонент, який забезпечує розуміння природної мови та персоналізоване обслуговування користувачів, вносячи в систему функції розширеного аналізу та адаптивних рекомендацій.

Клієнтська сторона:

- Фронтенд: Розроблений на React.js, фронтенд забезпечує інтерактивний користувацький інтерфейс, приймає користувацькі введення та відображає дані, отримані через API.

- Мобільний додаток: Розроблений з використанням Swift для iOS, забезпечує адаптований інтерфейс для мобільних користувачів і також взаємодіє з сервером через API.

- Взаємодія з API: Клієнтська сторона отримує та відправляє дані до сервера через RESTful API, що забезпечує гнучкість у взаємодії з бекендом.

Серверна сторона (бекенд):

- Бекенд: Реалізований на ASP.NET з використанням C# як основної мови програмування, бекенд керує бізнес-логікою, обробкою даних, взаємодіє з базою даних та надає дані фронтенду через API.

- База даних: Використовується Microsoft SQL Server (MS SQL) як реляційна СУБД для зберігання та управління даними. Вона забезпечує швидкий доступ та надійне зберігання інформації про користувачів, заходи, відгуки та інші системні дані.

Мобільний додаток:

а) Технології:

1) SwiftUI: Для розробки користувацького інтерфейсу додатка використовується SwiftUI, що забезпечує швидке та просте створення інтерактивних та зручних інтерфейсів.

2) Swift: Мова програмування Swift використовується для розробки логіки додатка. Swift є потужною та ефективною мовою, яка дозволяє створювати надійний та швидкий код.

3) HTTP та RESTful API: Для взаємодії між фронтендом і бекендом використовуються HTTP-протоколи та RESTful API. Це забезпечує стандартний та ефективний спосіб обміну даними між клієнтом та сервером.

У своєму мобільному проєкті є використовую архітектурне рішення MVM (Model-View-ViewModel):

- Модель (Model): Відповідає за представлення даних та бізнес-логіку додатка. Це може бути клас або структура даних, яка зберігає інформацію та виконує операції з нею.

- Представлення (View): Відповідає за візуальне представлення даних та обробку подій від користувача. Це може бути SwiftUI-вид або інший компонент інтерфейсу, який відображає інформацію та сприймає взаємодію з користувачем.

- ViewModel: Посередник між моделлю та представленням. Він отримує дані від моделі, обробляє їх та підготовлює для відображення в представленні. ViewModel також може містити логіку для валідації даних, обробки подій користувача та інше.

Інкапсуляція даних:

Для забезпечення безпеки та конфіденційності даних, модель і ViewModel можуть використовувати інкапсуляцію, обмежуючи доступ до даних та оперуючи ними лише через публічні методи та властивості.

Оптимізація швидкодії:

Під час розробки додатка варто приділяти увагу оптимізації швидкодії, наприклад, за допомогою використання ефективних алгоритмів та структур даних у моделі та ViewModel.

Тестування:

Використання паттерну MVVM сприяє легкості тестування окремих компонентів програми. Моделі та ViewModel можуть бути протестовані незалежно від представлення, що дозволяє швидко та ефективно перевіряти їх поведінку та функціональність.

Штучний інтелект в системі "Eventify" втілюється за допомогою LangChain[6], розробленого на Python[7]. Ця технологія застосовується для реалізації ШІ-асистента, який здатен вести діалог з користувачами, надавати персоналізовані рекомендації щодо заходів та обробляти запитання природною мовою.

У системі "Eventify" використовується HTTP-протокол для стандартного засобу комунікації між клієнтською та серверною частинами через Інтернет. Також використовується RESTful API для обміну даними між фронтендом і бекендом, що забезпечує простий та зрозумілий спосіб організації мережевих запитів.

Для створення діаграми розгортання системи "Eventify" ми відобразимо основні компоненти системи та їх взаємозв'язки. Діаграма розгортання покаже фізичну архітектуру системи, включаючи клієнтські й серверні вузли, а також комунікаційні зв'язки між ними. (див. рис. 3.4).

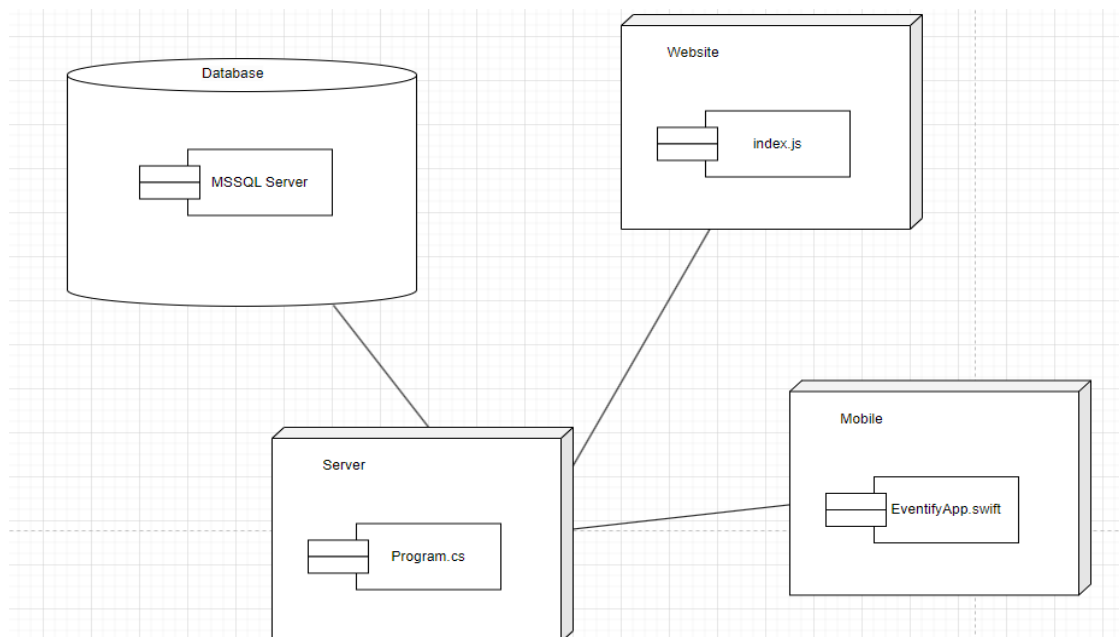


Рисунок 3.4 – Діаграма розгортання (рисунок виконано самостійно)

Основна мета діаграми розгортання полягає у відображенні фізичної структури системи, що включає апаратне та програмне забезпечення, мережеві з'єднання та інші технічні ресурси. Вона забезпечує зручний спосіб візуалізації розташування та взаємодії компонентів системи.

3.3 Проектування бази даних

Вибір Microsoft SQL Server для створення бази даних системи рекомендацій подій відповідає потребам у сучасному, надійному та адаптивному рішенні. SQL Server відомий своєю стабільністю, швидкістю обробки даних та здатністю інтегруватися з іншими продуктами Microsoft, що робить його ідеальним для систем на базі Windows. Його потужні інструменти для обробки даних, включаючи управління, аналіз та формування звітів, а також сильна підтримка та активна спільнота користувачів, роблять його прекрасним вибором для проєктів будь-якого розміру.

SQL Server Management Studio (SSMS) є інтегрованим середовищем для управління різними компонентами SQL Server, включаючи бази даних, звіти та аналіз. Це програмне забезпечення, розроблене Microsoft, забезпечує зручний інтерфейс для адміністрування та роботи з базами даних на платформі SQL Server. SSMS дозволяє створювати та редагувати бази даних, виконувати SQL-запити, розгортати бази даних, керувати безпекою, налаштовувати резервне копіювання та відновлення даних, а також проводити моніторинг та оптимізацію продуктивності. Це середовище має велику популярність серед адміністраторів баз даних і розробників, які працюють з продуктами SQL Server.

Таблиці бази даних, що використовуються сервером:

- Таблиця «User» використовується для зберігання основної інформації про користувача в системі. Вона містить його особисті та облікові дані.
- Таблиця «Event» зберігає основну інформацію про подію, таку як назва, час проведення та організатор.
- Таблиця «Message» містить інформацію про повідомлення, які користувач надсилає штучному інтелекту.

- Таблиця «Tag» зберігає інформацію про теги, які використовуються для відрізнєння подій, а також дає можливість користувачам підписуватися на нові події.
- Таблиця «Location» зберігає інформацію про місцєперебування подій на карті.
- Таблиця «ViewHistory» використовується для зберігання списку подій, що зацікавили користувача, для більш ефективного підбору рекомендацій у майбутньому.
- Таблиця «Settings» зберігає налаштування системи для кожного користувача окремо.
- Таблиця «Roles» містить дані про ролі в системі, допомагаючи визначити доступи відповідно до типу користувача.
- Таблиця «Tag-Event» є проміжною таблицею між «Tag» і «Event», оскільки одна подія може мати багато тєгів і один тєг може бути прив'язаний до багатьох подій.
- Таблиця «User-Tag» є проміжною таблицею між «User» і «Tag», оскільки один користувач може підписатися на кілька тєгів, і один тєг може мати багато підписників (див. рис. 3.5).

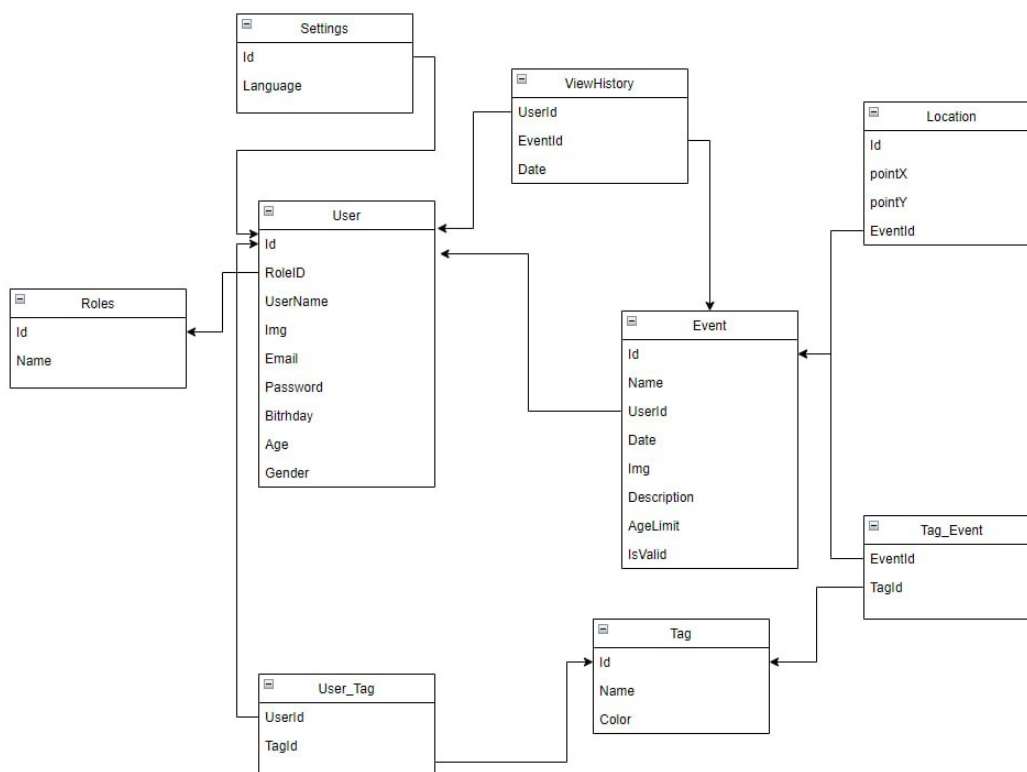


Рисунок 3.5 – ER діаграма (рисунок виконано самостійно)

База даних знаходиться в третій нормальній формі, що підтверджує її нормалізованість. Однією з переваг такої структури є відсутність повторюваної інформації, що забезпечує зручність у зберіганні та обробці даних. Це сприяє ефективному використанню пам'яті, особливо при роботі з великими обсягами даних.

3.4 Огляд алгоритмів та методів

В системі "Eventify" одним з ключових алгоритмів є процес реєстрації та авторизації користувачів, який безпосередньо пов'язаний з персоналізацією рекомендацій подій. При реєстрації користувач вводить свої персональні дані та проходить опитування, яке допомагає системі зрозуміти його інтереси. Цей процес починається з того, що користувач починає з введення своїх основних даних, таких як ім'я, електронна адреса та пароль, що є стандартною процедурою для створення

нового облікового запису. Після цього система запрошує користувача пройти коротке опитування, яке включає питання про типи подій. Ця інформація використовується для створення профілю користувача, який допомагає системі точніше підбирати рекомендації.

Процес авторизації здійснюється коли користувач входить у систему. Під час цього етапу введені користувачем дані перевіряються на відповідність збереженим у базі даних, що забезпечує доступ до персоналізованого контенту та рекомендацій. Якщо дані збігаються, користувач отримує доступ до свого профілю та персоналізованих подій.

Важливою частиною цього процесу є також забезпечення безпеки даних користувачів. Використання сучасних методів шифрування та безпечних протоколів зберігання паролів гарантує, що особиста інформація та переваги користувачів захищені від несанкціонованого доступу.

3.5 Створення UI/UX

В системі "Eventify" одним з ключових аспектів є UI/UX він відіграє важливу роль у забезпеченні задоволення користувачів та наданні їм зручного та приємного досвіду використання програми.

UI (інтерфейс користувача) описує елементи та взаємодію користувача з програмою, такі як кнопки, поля введення, меню тощо. UX (взаємодія користувача) визначає загальний досвід використання програми, включаючи його зручність, інтуїтивність та задоволення від використання.

Обравши кольори для програми Eventify, я врахував її цільову аудиторію та цілі проєкту. Наприклад, для культурних заходів можна використовувати теплі, живі кольори, які стимулюють енергію та творчий підхід. Для досягнення цих цілей було обрано трав'янисто-зелений колір, який є легким для зорового сприйняття.

Щодо шрифтів, важливо обрати читабельний та легко сприйманий шрифт, який буде зручним для читання на екрані мобільного пристрою. Краще використовувати шрифти без засічок, які добре читаються на екранах різного розміру. Саме тому я обрав шрифт San Francisco, оскільки він є системним для

пристроїв компанії Apple та найпоширенішим серед користувачів смартфонів на iOS.

При вході у додаток користувача зустрічає екран авторизації, він виконан у приємних зелених кольорах, має анімацію дихання та є зручним та інтуїтивним (див. рис. 3.5).

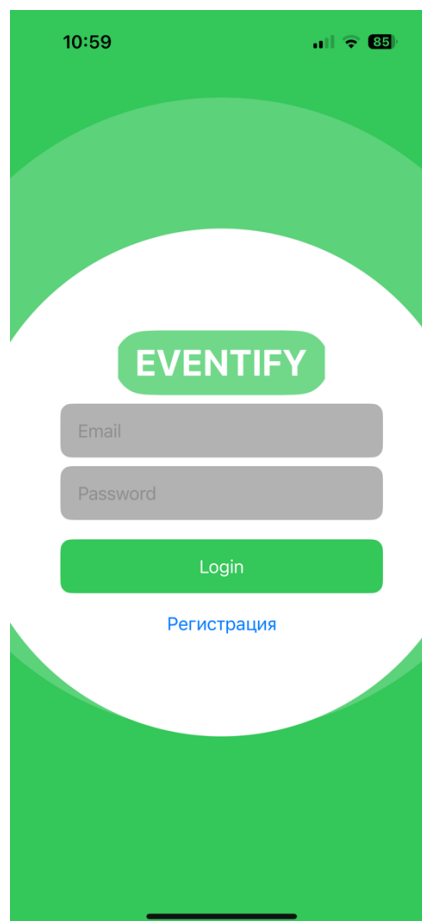


Рисунок 3.5 – Екран авторизації (рисунок виконано самостійно)

Перше, що зустрічає користувача після авторизації є мапа подій, на якій зображено логотип, аватар користувача, мапу та навігаційну панель у нижній частині екрана (див. рис. 3.6).

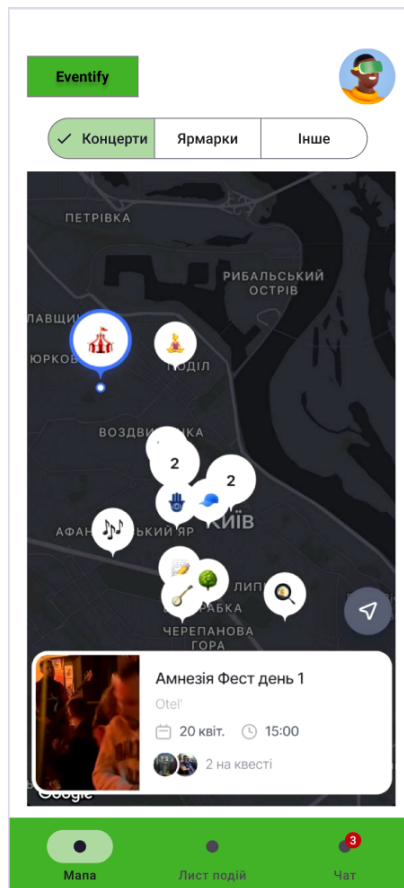


Рисунок 3.6 – Мапа подій (рисунок виконано самостійно)

На цьому дизайні ми бачимо інтуїтивний та зрозумілий інтерфейс, який одразу дозволяє користувачу знаходити події, у яких він зацікавлений. Також нижче знаходиться навігаційна панель, що дозволяє користувачу переходити до різних частин додатка. Далі користувач має змогу перейти до "Листу подій", де він зможе переглядати релевантні події у вигляді списку (див. рис. 3.7).

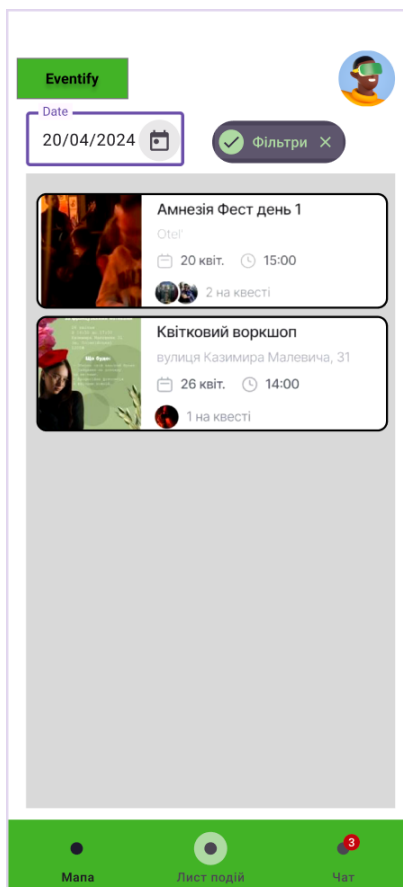


Рисунок 3.7 – Лист подій (рисунок виконано самостійно)

Цей дизайн містить у собі ряд обов'язкових елементів: лого, аватар користувача та навігаційну панель. Окрім цього він має простір, у якому з'являються події у вигляді списку, фільтр за часом проведення подій та додаткову кнопку фільтрації за іншими критеріями. Це робить дизайн зручним для використання користувачами.

У мобільних додатках UI/UX важливіші, ніж у вебверсіях, оскільки на мобільних пристроях обмежений простір екрана, а користувачам потрібно якісно взаємодіяти з додатком на невеликому просторі. Важливо враховувати зручність та простоту використання, адаптивність до різних розмірів екранів та можливість швидкої навігації між різними функціями додатку.

4 ОПИС ПРИЙНЯТИХ ПРОГРАМНИХ РІШЕНЬ

4.1 Інтеграція з Apple Maps для відображення подій

Для забезпечення інтерактивної карти в клієнтській частині використовується Apple Maps, що надає потужні інструменти для роботи з картографічними даними. Це дозволяє користувачам переглядати події на карті, шукати місця за допомогою автозаповнення, а також отримувати інформацію про події безпосередньо на карті. Інтеграція з Apple Maps забезпечується за допомогою фреймворку MapKit, який дозволяє завантажувати необхідні компоненти та використовувати їх у SwiftUI для відображення геолокаційних даних і взаємодії з ними (див. рис. 4.1).

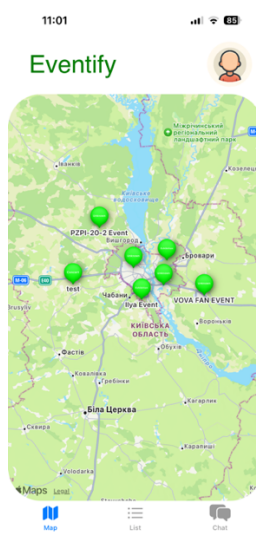


Рисунок 4.1 – Інтеграція Apple Maps для відображення подій на карті

Для більш інтерактивної взаємодії користувача з подією були створені реакції. Існує дві реакції: лайк(серце) та дизлайк(палець униз). Це дозволяє користувачу проявити свої побажання та інтереси, що робить цей функціонал важливим для користувачів. Ставлення реакцій здійснюється шляхом трьох запитів на Back-end, а саме: додавання реакції, оновлення реакції та видалення реакції (див. рис. 4.2).

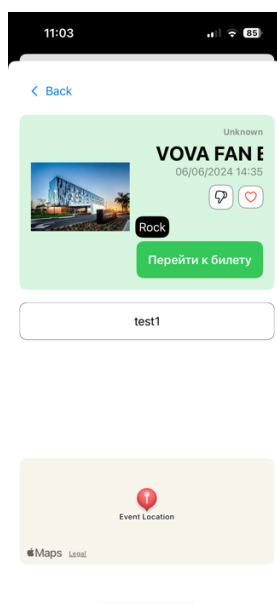


Рисунок 4.2 – Функціонал проставлення реакцій для інтерактивності користувачів.

Для покращення візуального сприйняття та ідентифікації різних типів подій використовуються кастомні іконки для маркерів на карті. Кожен маркер має власну іконку, що стилістично відповідає типу події. Наприклад, музика, театр, технології тощо. Це рішення забезпечує чітке розрізнення подій на карті, полегшуючи користувачам процес ідентифікації та вибору заходів (див. рис. 4.3).

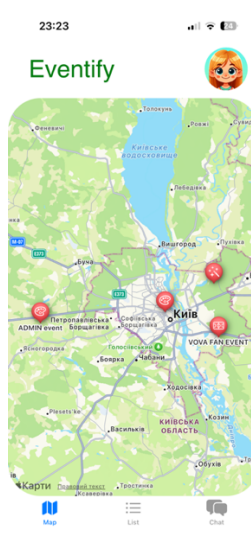


Рисунок 4.3 – Відображення маркерів з кастомними іконками на основі типу події.

Клієнтська частина підтримує інтерактивність маркерів, дозволяючи користувачам натискати на маркери для перегляду детальної інформації про подію. Це забезпечує зручний доступ до додаткової інформації через модальні вікна, що відкриваються при натисканні на маркер (див. рис. 4.4).

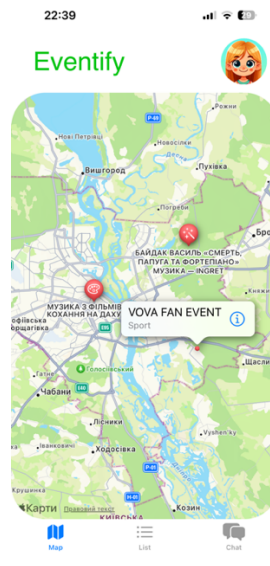


Рисунок 4.4 – Інтерактивність маркерів: відкриття модального вікна з деталями події

4.2 Інтеграція з Azure й OpenAI API для ШІ чат-бота

Для обробки природної мови та генерації відповідей на запити користувачів використовується Azure й OpenAI API. Це забезпечує підтримку розуміння природної мови та генерацію відповідей за допомогою моделі GPT-4, що значно підвищує ефективність і точність ШІ чат-бота. Інтеграція з Azure дозволяє зберігати історію розмов у пам'яті, що дозволяє ефективно управляти контекстом діалогу і підтримувати цілісність сесії (див. рис. 4.5).

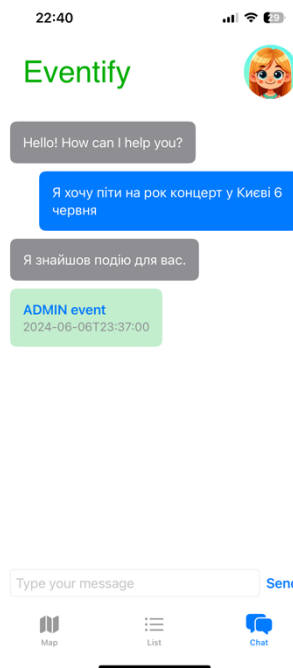


Рисунок 4.5 – Інтеграція з Azure та OpenAI API для обробки природної мови

4.3 Управління станом клієнтської частини за допомогою `@State` і `@ObservedObject`

Для управління станом клієнтської частини використовується `@State` та `@ObservedObject`, які забезпечують централізоване й ефективне управління станом користувача, токенами аутентифікації та іншими даними. Використання цих властивостей забезпечує централізоване зберігання стану додатка, спрощує керування станом, робить код більш організованим і сприяє ефективній роботі з даними.

Користувацькі дані, такі як профіль та токен аутентифікації, зберігаються у локальному сховищі. Якщо токен аутентифікації прострочений, він автоматично видаляється, забезпечуючи актуальність даних і безпеку. Аутентифікація здійснюється за допомогою JWT, які зберігаються в локальному сховищі. При вході, виході, зміні профілю і оновленні зображення профілю, стан оновлюється і синхронізується з локальним сховищем для забезпечення безперервності між сесіями. Цей підхід забезпечує надійне управління аутентифікацією і профілем користувача, підвищуючи безпеку і зручність використання додатка (див. рис. 4.6).

```
@State private var showResetPasswordModal = false
@State private var showRegistrationModal = false
@State private var pulsate = false
@ObservedObject var user = User(name: "", email: "", phone: "",
    password: "", userID: "")
```

Рисунок 4.6 – Конфігурація управління станом з @State і @ObservedObject

4.4 Вікно акаунту користувача для зміни аватару, редагування особистих даних та зміни локалізації.

Клієнтська частина додатку містить вікно акаунту користувача, де можна змінити аватарку, редагувати особисті дані та змінювати локалізацію. Це вікно надає користувачам можливість персоналізувати свій профіль, забезпечуючи зручний інтерфейс для оновлення інформації. Користувач може змінити аватарку, вибравши будь-який варіант з переліку, змінити своє ім'я, електронну адресу та номер мобільного телефону. Також є можливість обрати або змінити мову інтерфейсу, що дозволяє користувачам налаштувати додаток відповідно до своїх уподобань та потреб. Це рішення підвищує зручність використання та персоналізацію додатку (див. рис. 4.7).

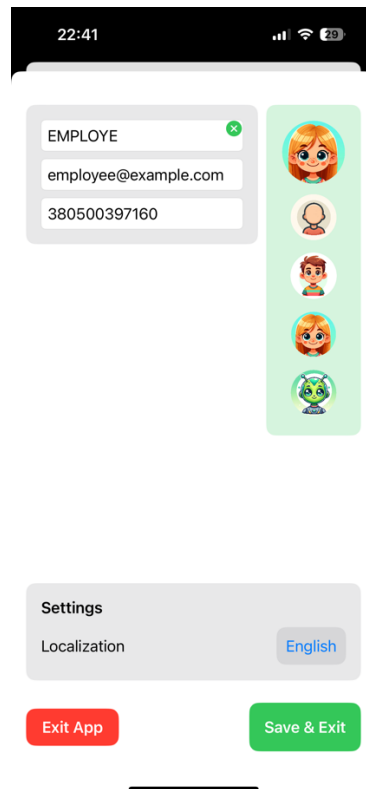


Рисунок 4.7 – Вікно акаунту користувача для зміни аватарки, редагування особистих даних та зміни локалізації.

Таким чином користувач має змогу оновити у будь-який момент свої дані та за бажанням змінити свій аватар, що робить сторінку профілю невід’ємною частиною мого додатку.

5 ТЕСТУВАННЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для виявлення помилок під час розробки мобільного додатка "Eventify" було застосовано мануальне тестування. Це означає, що перевірка програмного забезпечення на наявність помилок здійснювалася вручну, без використання автоматизованих тестів.

5.1 План тестування

Для ефективного тестування було складено план, де зазначено мету тестування та перелік функцій для перевірки на правильність роботи. Згідно з цим планом, було протестовано різні функціональні аспекти додатка, такі як створення, редагування та видалення даних, а також валідацію вхідних даних.

5.2 Тест-кейси

Тест-кейс №1 – Аутентифікація користувача

1. Послідовність роботи:

- Відкрити додаток.
- Ввести електронну адресу та пароль.
- Натиснути кнопку "Login".

2. Використані дані:

- Електронна адреса: testuser@example.com
- Пароль: password123

3. Результат, що очікується: Користувач успішно входить у систему, переходячи на головну сторінку.

4. Результат, який було отримано: Користувач успішно увійшов у систему, головна сторінка відобразилася.

5. Висновок: Тест пройдено.

Тест-кейс №2 – Оновлення профілю користувача

1. Послідовність роботи:

- Увійти під обліковим записом Користувача.
- Перейти на сторінку профілю, натиснувши на аватар користувача у навігаційному меню.

- Натиснути кнопку "Edit Profile".
- Ввести нові дані та натиснути "Save and Exit".

2. Використані дані:

- Ім'я: Vlad
- Телефон: +380500000000
- Зображення: profile_image.png
- Теги: Rock

3. Результат, що очікується: Після натискання на кнопку "Save", дані оновляться.

4. Результат, який було отримано: Після натискання на кнопку "Save", дані оновилися.

5. Висновок: Тест пройдено.

Тест-кейс №3 – Сортування, фільтрація та пошук заходів

1. Послідовність роботи:

- Увійти під акаунтом користувача.
- Перейти на сторінку заходів.
- Вибрати потрібні фільтри для сортування, пошуку та вибору категорій і типів.

2. Використані дані:

- Сортування за датою: За зростанням.
- Категорія: Концерт.
- Пошукова комірка: Fan event.

3. Результат, що очікується: Після введення даних події автоматично з'являться на сторінці.

4. Результат, який було отримано: Після введення даних події автоматично з'явилися на сторінці.

5. Висновок: Тест пройдено.

Тест-кейс №4 – Оцінка заходу

1. Послідовність роботи:

- Увійти під акаунтом користувача.
- Перейти на сторінку заходу, який хочеться оцінити.
- Натиснути кнопку "Like" або "Dislike".

2. Використані дані:

- Захід: Rock Concert

3. Результат, що очікується: Оцінка зберігається і відображається у списку оцінених заходів користувача.

4. Результат, який було отримано: Оцінка збереглася і відобразилася у списку оцінених заходів користувача.

5. Висновок: Тест пройдено.

Тест-кейс №5 – Спілкування з ШІ чат-ботом

1. Послідовність роботи:

- Увійти під акаунтом користувача.
- Перейти на сторінку чату.
- Написати повідомлення у полі для введення та натиснути кнопку "Send".

2. Використані дані:

- Повідомлення: Привіт, я хочу сходити на рок-концерт у Києві 20 червня.

3. Результат, що очікується: Відповідь від чат-бота з заходами, що відповідають запиту користувача.

4. Результат, який було отримано: Відповідь від чат-бота з заходами, що відповідають запиту користувача.

5. Висновок: Тест пройдено.

Тест-кейс №6 – Перегляд подій на мапі

1. Послідовність роботи:

- Увійти під акаунтом користувача.
- Перейти на вкладку "Мар".
- Переглянути події на інтерактивній мапі.

2. Використані дані:

- Місце проведення: Київ

3. Результат, що очікується: Відображення подій на мапі у вигляді маркерів.

4. Результат, який було отримано: Події відобразилися на мапі у вигляді маркерів.

5. Висновок: Тест пройдено.

Тест-кейс №7 – Перегляд детальної інформації про захід

1. Послідовність роботи:

- Увійти під акаунтом користувача.
- Перейти на сторінку заходу.
- Натиснути на захід для перегляду детальної інформації.

2. Використані дані:

- Захід: Rock Concert

3. Результат, що очікується: Відображення детальної інформації про захід.

4. Результат, який було отримано: Відображення детальної інформації про захід.

5. Висновок: Тест пройдено.

Зробивши мануальне тестування усіх ключових компонентів мобільного додатка, можна зробити висновок, що всі функції працюють правильно, і не було виявлено ніяких помилок.

ВИСНОВКИ

У ході виконання кваліфікаційної роботи бакалавра була розроблена мобільна клієнтська частина програмної системи "Eventify", яка спрямована на рекомендацію та підбір актуальних культурних подій, адаптованих до індивідуальних уподобань та геолокації користувачів.

Розробка здійснювалася на основі сучасних технологій, зокрема, мови програмування Swift та фреймворку SwiftUI для створення інтерфейсу користувача. У процесі роботи була реалізована інтерактивна карта подій за допомогою MapKit, що дозволяє користувачам зручно переглядати та фільтрувати події за місцем їх проведення.

Використання архітектури Model-View-ViewModel (MVVM) стало важливим елементом у забезпеченні чіткого розділення відповідальності між моделями даних, уявленнями та логікою управління станом. Це рішення дозволило спростити розробку та тестування окремих компонентів додатка, підвищуючи продуктивність та забезпечуючи зручність управління різними аспектами системи.

Інтеграція з серверною частиною здійснювалася за допомогою RESTful API, що забезпечує гнучкість та ефективність взаємодії між клієнтом та сервером. Використання HTTP-запитів для обміну даними дозволило забезпечити швидкий та надійний доступ до інформації про події, користувачів та їхні вподобання.

Чат-бот, інтегрований через OpenAI API, забезпечує високоякісну підтримку природної мови, дозволяючи надавати персоналізовані рекомендації та відповідати на запитання користувачів в реальному часі. Це підвищує рівень взаємодії з користувачами та забезпечує більш інтуїтивний досвід використання додатка.

Розроблена система "Eventify" не лише відповідає сучасним вимогам до програмного забезпечення, але й демонструє значний потенціал для подальшого розвитку та оптимізації. Дотримання принципів сучасної розробки, таких як компонентність і модульність, а також інтеграція передових технологій забезпечують гнучкість, масштабованість та продуктивність системи. Цей підхід дозволяє легко адаптувати систему до змін у потребах користувачів та розвитку

технологій, забезпечуючи її надійність та ефективність у вирішенні завдань з підбору культурних подій.

Загалом, виконана робота доводить, що "Eventify" є ефективним інструментом для покращення доступу до культурних заходів, враховуючи індивідуальні вподобання користувачів та забезпечуючи високий рівень персоналізації та зручності.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Афіша і квитки Kontramarka.ua [Електронний ресурс] – URL: <https://kontramarka.ua/uk> (дата звернення: 09.06.2024).
2. Афіша подій KARABAS [Електронний ресурс] – URL: <https://kyiv.karabas.com/ua/> (дата звернення: 09.06.2024).
3. Swift.org – Documentation. [Електронний ресурс] – URL: <https://www.swift.org/documentation/> (дата звернення: 13.05.2024).
4. SwiftUI | Apple Developer – Documentation. [Електронний ресурс] – URL: <https://developer.apple.com/documentation/swiftui/> (дата звернення: 13.05.2024).
5. ASP.NET Tutorial [Електронний ресурс] – URL: <https://dotnet.microsoft.com/en-us/learn/aspnet/hello-world-tutorial/intro> (дата звернення: 13.05.2024)
6. Introduction | LangChain. [Електронний ресурс] – URL: https://python.langchain.com/v0.1/docs/get_started/introduction/ (дата звернення: 13.05.2024).
7. Python 3.12.3 documentation. [Електронний ресурс] – URL: <https://docs.python.org/3/> (дата звернення: 13.05.2024).
8. Лавріщева К. М. Програмна інженерія / К. М. Лавріщева. – К. : Академперіодика, 2008. – 319 с.
9. Nguyen H. Single-page application and front-end testing methods: built with React and React Router, tested with Jest and Cypress. – 2022. [Електронний ресурс] – URL: https://www.theseus.fi/bitstream/handle/10024/745098/Nguyen_Huong.pdf (дата звернення: 13.05.2024).
10. Мазурова О. А., Широкопетлева М. С., Черепанова Ю. Ю. Информационные системы. [Електронний ресурс] – URL: https://dl.nure.ua/pluginfile.php/509/mod_resource/content/2/01.pdf (дата звернення: 13.05.2024).

ДОДАТОК А

Звіт результатів перевірки на унікальність тексту в базі ХНУРЕ



Ім'я користувача:
Олійник Олена Володимирівна каф. ПІ

ID перевірки:
1016348951

Дата перевірки:
11.06.2024 19:25:31 EEST

Тип перевірки:
Doc vs Library

Дата звіту:
11.06.2024 19:26:19 EEST

ID користувача:
100012353

Назва документа: 2024_Б_ПІ_ПЗПІ_20_2_Дегтяр_В_Е_скорочений

Кількість сторінок: 35 Кількість слів: 4838 Кількість символів: 37992 Розмір файлу: 3.28 MB ID файлу: 1016152198

Виявлено модифікації тексту (можуть впливати на відсоток схожості)

24.7%
Схожість

Найбільша схожість: 22.2% з джерелом з Бібліотеки (ID файлу: 1016143265)

Пошук збігів з Інтернетом не проводився

24.7% Джерела з Бібліотеки

136

Сторінка 37

0% Цитат

Вилучення цитат вимкнено

Вилучення списку бібліографічних посилань вимкнено

0%
Вилучень

Немає вилучених джерел


Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Підозріле форматування 7 сторінок

ДОДАТОК Б

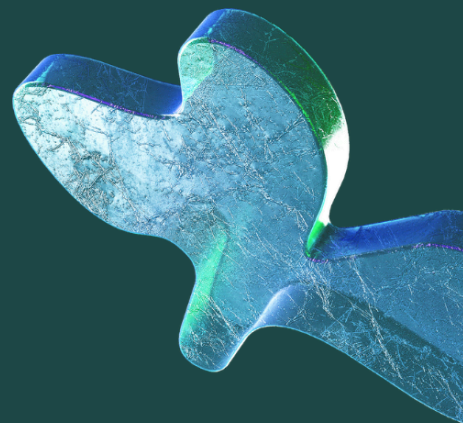
Слайди презентації



ПРОГРАМНА СИСТЕМА ДЛЯ РЕКОМЕНДАЦІЇ ТА ПІДБОРУ АКТУАЛЬНИХ КУЛЬТУРНИХ ПОДІЙ. MOBILE APPLICATION.

Виконав:
ст.гр.ПЗПІ-20-2
Дегтяр В.Е.

Керівник:
доц. кафедри ПІ Побіженко І.О.




Проблеми сучасних рішень у пошуку культурних заходів?



Зі збільшенням кількості культурних заходів виникає проблема вибору. Давайте подивимося на деякі існуючі рішення. Наприклад, Kontramarka та Karabas.

- Kontramarka.ua надає інформацію про події та можливість замовлення квитків, але не має інтерактивної карти, що ускладнює пошук заходів за місцеперебуванням.
- Karabas.com пропонує схожий функціонал, але також не забезпечує інтерактивні рекомендації, що робить процес вибору менш зручним для користувачів.

Також обидва ці сервіси мають мобільні додатки, але їх функціонал дублює веб-застосунок.



Цілі "Eventify"

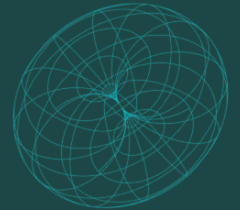
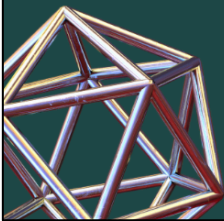
- Забезпечення високої точності рекомендацій
- Ефективне управління даними користувачів
- Дотримання сучасних стандартів безпеки та конфіденційності

Основні функції додатка

- Персоналізація
- Інтерактивна карта
- ШІ-асистент

Персоналізація дозволяє нам надавати рекомендації на основі ваших вподобань та історії відвідувань. Інтерактивна карта дозволяє зручно переглядати події поблизу вас. ШІ-асистент допомагає у виборі заходів, відповідає на ваші запитання та надає додаткову інформацію про події.

Технології та інструменти



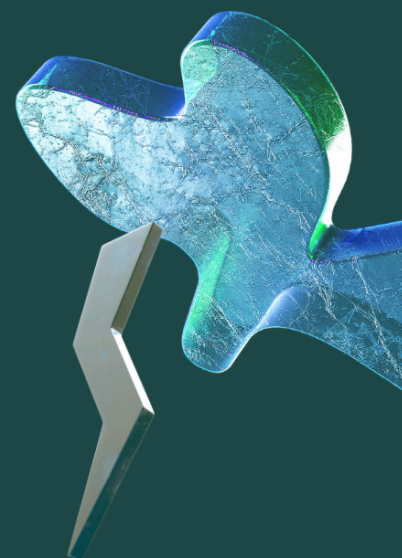
Архітектура системи



Архітектура "Eventify" базується на клієнт-серверній моделі, що забезпечує гнучкість та ефективність взаємодії між клієнтом та сервером. На фронтенді використовується React.js, що забезпечує інтерактивний користувацький інтерфейс. Мобільний додаток, реалізований на Swift, також взаємодіє з сервером через RESTful API.

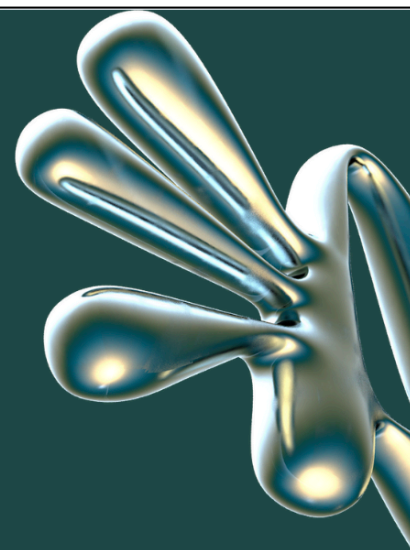
Тестування розробленого ПЗ

- Було проведено мануальне тестування основного функціоналу та можливостей мобільного додатку
- Основні тест-кейси: аутентифікація, оновлення профілю, фільтрація заходів, локалізація, реакції до подій, спілкування з ШІ



Апробація

Сафошин В. В., Хамінов І. О., Дегтяр В. Е., Побіженко І. О. Eventify: інноваційний підхід до організації та пошуку заходів з використанням ШІ. XXVIII міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті», конференція «Інформаційні інтелектуальні системи» - с. 544.

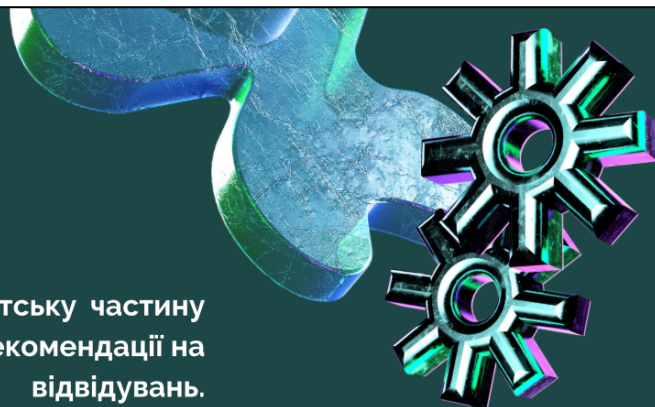


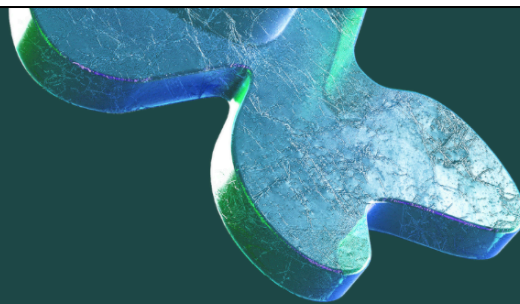
Потенціал для подальшого розвитку

"Eventify" має великий потенціал для подальшого розвитку. Ми плануємо продовжити оптимізацію додатку, щоб зробити його ще більш зручним та ефективним для користувачів. Також ми розглядаємо можливості розширення функціоналу, щоб задовольнити потреби ще більшої кількості користувачів та забезпечити ще кращу персоналізацію та взаємодію з культурними заходами.

Висновок

Ми успішно розробили мобільну клієнтську частину "Eventify", яка надає персоналізовані рекомендації на основі ваших інтересів та історії відвідувань. Інтерактивна карта дозволяє зручно переглядати події поблизу вас, а ШІ-асистент допомагає у виборі заходів та відповідає на ваші запитання в реальному часі. Це робить "Eventify" ефективним інструментом для покращення доступу до культурних заходів.





Дякую за увагу!

ДОДАТОК В

Специфікація вимог до програмного продукту

Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп'ютерних наук

Кафедра програмної інженерії

СПЕЦИФІКАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Програмна система для рекомендації та підбору актуальних культурних подій

Студент гр. ПЗПІ-20-2 Дегтяр Владислав Едуардович

Студент гр. ПЗПІ-20-3 Сафощин Володимир Володимирович

Студент гр. ПЗПІ-20-3 Хамінов Ілля Олександрович

Харків

2024р.

ЗМІСТ

Вступ	53
1.1 Огляд продукту.....	53
1.2 Мета.....	54
1.3 Межі	54
1.4 Посилання.....	55
1.5 Означення та аббревіатури.....	56
2. Загальний опис	58
2.1 Перспективи продукту.....	58
2.2 Функції продукту	59
2.3 Характеристика користувачів	59
2.4 Загальні обмеження	59
2.5 Припущення й залежності	60
3 Конкретні вимоги.....	61
3.1 Вимоги до зовнішніх інтерфейсів	61
3.1.1 Інтерфейс користувача	61
3.1.2 Апаратний інтерфейс.....	61
3.1.3 Програмний інтерфейс	62
3.1.4 Комунікаційний протокол	62
3.1.5 Обмеження пам'яті	63
3.1.6 Операції	63
3.2 Атрибути програмного продукту.....	63
3.2.1 Надійність.....	63
3.2.2 Доступність	64
3.2.3 Безпека	64
3.2.4 Супроводжуваність.....	65
3.2.5 Переносимість.....	65
3.2.6 Продуктивність	65
3.2.7 Вимоги бази даних.....	65

ВСТУП

1.1 Огляд продукту

Сервіс "Eventify" з інтегрованим back-end рішенням представляє собою комплексний інструмент для користувачів, зацікавлених у відвідуванні та організації різноманітних заходів, включаючи концерти, виставки, ярмарки, фестивалі. Цей продукт включає в себе широкий спектр функцій для ефективного пошуку, відбору та участі в заходах.

Автентифікація та авторизація користувачів є ключовими елементами безпеки сервісу, що гарантує захист особистої інформації та контроль доступу до функцій залежно від ролі користувача. Ці можливості дозволяють користувачам створювати персональні профілі, налаштовувати інтереси та отримувати персоналізовані рекомендації.

"Eventify" спрощує процес відкриття та реєстрації на заходи завдяки інтуїтивно зрозумілому інтерфейсу, де користувачі можуть переглядати детальну інформацію про події, включаючи час, дату, місце проведення, а також реєструватися на них в кілька кліків. Це забезпечує зручний доступ до культурних та соціальних заходів.

Система також має функцію керування подіями, де організатори можуть створювати нові заходи, редагувати існуючі, управляти реєстраціями та відгуками відвідувачів. Це дає змогу ефективно організовувати та планувати заходи, а також збирати зворотний зв'язок для покращення майбутніх подій.

Додатково, сервіс включає можливість спілкування з ШІ асистентом, який допомагає користувачам у виборі подій на основі їхніх переваг, історії пошуку та оцінок, підвищуючи особистісний досвід використання платформи.

Загалом, "Eventify" є інноваційним рішенням для тих, хто шукає активне культурне життя, надаючи зручні інструменти для відкриття, планування та участі в заходах, а також для організаторів, щоб ефективно управляти та просувати свої події.

1.2 Мета

Основною метою проекту "Eventify" є створення універсального сервісу, який спрощує процес пошуку, вибору та участі в різноманітних заходах, таких як концерти, виставки, ярмарки, фестивалі, для користувачів. Цей сервіс надає можливість перегляду актуальних подій на карті місцевості, їх фільтрації за інтересами та пошуку, а також рекомендації заходів, які можуть бути цікавими для конкретного користувача.

Додатково, "Eventify" забезпечує верифікацію користувачів, можливість для переходу на сторінку для замовлення квитків та запрошень на заходи, виставлення оцінок. Сервіс також включає інтеграцію зі штучним інтелектом, який виступає в ролі асистента для користувачів, допомагаючи у виборі заходів на основі їх переваг.

Таким чином, мета проекту "Eventify" полягає у забезпеченні зручного, інтуїтивно зрозумілого інструменту для відкриття нових можливостей дозвілля, сприяння культурному розвитку та соціалізації користувачів через участь у заходах, що відбуваються в їхній місцевості або в інших регіонах.

1.3 Межі

У контексті проекту "Eventify", межі визначаються функціональними та технічними аспектами системи. Проект зосереджений на наданні користувачам інструментів для пошуку, фільтрації, рекомендації та участі у заходах, а також на інтеграції з платформами для замовлення квитків та спілкуванні з ШІ асистентом. Важливими аспектами є верифікація користувачів, виставлення оцінок подіям.

Технічні обмеження системи включають обмеження щодо ресурсів сервера, які можуть вплинути на її масштабованість та продуктивність при великій кількості одночасних запитів. Це також стосується обмежень на кількість одночасних користувачів та подій, які можуть бути оброблені системою без зниження її ефективності.

З точки зору безпеки, проект має забезпечити захист даних користувачів, включаючи аутентифікацію, авторизацію, шифрування даних та захист від

несанкціонованого доступу. Це вимагає використання сучасних протоколів безпеки та методів шифрування, а також постійного моніторингу та оновлення системи безпеки.

Проект "Eventify" не займається безпосередньою організацією заходів, а лише надає платформу для їх пошуку та рекомендації. Тому, відповідальність за точність інформації про заходи, їх виконання та якість лежить на організаторах подій та представниках заходів.

1.4 Посилання

Проект "Eventify" прагне автоматизувати та оптимізувати процеси вибору та участі в культурних заходах, мінімізуючи необхідність ручного пошуку актуальних подій у різних джерелах. Інтегровані функції верифікації користувачів, бронювання квитків, виставлення оцінок, а також відображення подій на карті надають користувачам зручний і інтуїтивно зрозумілий доступ до необхідної інформації. Це значно спрощує процес взаємодії з подіями та їх відвідування, покращуючи загальний досвід користувача.

"Eventify" також забезпечує організаторам ефективний канал для залучення нової аудиторії та просування своїх заходів. Сервіс дозволяє їм швидко оновлювати інформацію про події, управляти реєстраціями та отримувати зворотний зв'язок від учасників. Це створює можливості для покращення організації заходів та підвищення їх якості.

Таким чином, "Eventify" пропонує цілісний підхід до взаємодії з культурними подіями, сприяючи культурному розвитку користувачів та ефективному управлінню подіями, що покращує досвід як відвідувачів, так і організаторів

1.5 Означення та аббревіатури

Backend – це серверна частина програмного забезпечення, яка відіграє ключову роль у обробці даних, виконанні бізнес-логіки та забезпеченні безпеки. Вона управляє взаємодією з базою даних, виконує процеси аутентифікації та авторизації користувачів і забезпечує виконання операцій, важливих для функціонування системи.

Frontend - це клієнтська частина програмного забезпечення, відповідальна за інтерфейс та взаємодію з користувачем. Вона включає в себе дизайн, розмітку та скрипти сторінок, які користувачі бачать та з якими взаємодіють в браузері або через мобільні додатки.

API (Application Programming Interface) – це інтерфейс програмування застосунків, що дозволяє різним програмним компонентам спілкуватися між собою. API виступає як місток, що визначає правила та методи, за допомогою яких можна доступатися до функціоналу або даних однієї програми з іншої.

ORM (Object-Relational Mapping) – технологія, що дозволяє мапінг реляційних баз даних на об'єктно-орієнтовану модель програми. Завдяки ORM, робота з базою даних стає більш інтуїтивно зрозумілою для розробників, оскільки вони можуть оперувати об'єктами замість SQL-запитів.

HTTP (HyperText Transfer Protocol) – основний протокол передачі даних у Всесвітній мережі, який використовується для завантаження веб-сторінок з сервера на клієнтський браузер. HTTP визначає спосіб, яким повинні бути сформовані та передані запити та відповіді між клієнтом та сервером.

CRUD (Create, Read, Update, Delete) – це концепція, що описує чотири основні дії, які використовуються при роботі з даними: створення нових записів, читання (вибірка) існуючих даних, оновлення (модифікація) даних та видалення записів. Ці операції є фундаментальними для будь-яких систем, які працюють з базами даних.

LLM (Large Language Model) – це велика мовна модель, що використовує алгоритми глибокого навчання для обробки та генерації природної мови, здатна

виконувати широкий спектр задач, від автоматичного перекладу до створення текстового контенту. LLM використовуються для покращення здатності комп'ютерних систем розуміти та взаємодіяти з людською мовою, надаючи можливості для створення більш продвинутих і зручних користувацьких інтерфейсів, а також для аналізу великих обсягів текстових даних.

2. ЗАГАЛЬНИЙ ОПИС

2.1 Перспективи продукту

Перспективи програмної системи “Eventify” – надання рекомендацій та підбору актуальних заходів (концертів, виставок, ярмарок, фестивалів) є обнадійливими і обіцяють багато можливостей для подальшого розвитку та розширення.

З урахуванням зростаючого інтересу до культурних подій та розваг, а також зростаючої потреби у персоналізованих та зручних інструментах для вибору та участі в заходах, "Eventify" може знайти своє місце в різних галузях та сприяти розвитку сфери розваг та культурно-масових заходів.

Перспективи продукту включають розширення функціональності та додавання нових модулів, щоб задовольнити специфічні потреби організацій та користувачів. Наприклад, можливості планування і розподілу ресурсів, система відгуків, інтеграція з іншими платформами або соціальними мережами, розширена аналітика та звітність.

Крім того, вдосконалення аналітичних засобів для оцінки успішності різних заходів та рекламних кампаній. Розробка розширених звітів для організаторів та партнерів для підвищення ефективності подій. Розширення географічного охоплення для підтримки подій у різних регіонах та країнах. Активне залучення до міжнародних культурних заходів та фестивалів для розширення аудиторії.

Загалом, "Eventify" має великий потенціал стати ключовим гравцем у сфері організації та участі в культурних та розважальних заходах. Розширення функціональності, посилення географічного охоплення та впровадження новітніх технологій можуть вивести продукт на новий рівень та забезпечити йому стабільне місце на ринку.

2.2 Функції продукту

Основні функції цього продукту включають:

- FE-1: управління користувачами;
- FE-2: управління заходами;
- FE-3: безпека та доступ до даних;
- FE-4: пошук і фільтрація заходів;
- FE-5: взаємодія з ШІ асистентом;
- FE-6: перегляд подій на карті.

2.3 Характеристика користувачів

Характеристики користувачів програмної системи “Eventify” – надання рекомендацій та підбору актуальних заходів (концертів, виставок, ярмарок, фестивалів) можуть бути різноманітними, оскільки система взаємодіє з різними типами користувачів. Основні характеристики користувачів включають:

- користувачі є основними в системі. Це люди будь-якого віку та статі. Їх характеристики можуть включати їхні інтереси, їх вік та локацію де вони мешкають;
- менеджери компаній відповідають за керування подіями. Вони мають розширені права доступу та можуть створювати, редагувати та видаляти події, керувати користувачами, аналізувати звітність та здійснювати інші адміністративні функції;
- адміністратори системи відповідають за керування користувачами та менеджерами.

2.4 Загальні обмеження

У програмній системі “Eventify” – надання рекомендацій та підбору актуальних заходів (концертів, виставок, ярмарок, фестивалів) існують загальні обмеження, які можуть впливати на її функціонування та використання. Деякі з цих обмежень включають.

Система повинна дотримуватися вимог безпеки, щоб захистити дані користувачів та запобігти несанкціонованому доступу. Обмеження безпеки можуть включати криптографічні вимоги, захист від атак, правила доступу до даних та контроль прав користувачів.

Також система повинна бути здатна масштабуватись для впорядкування зростаючого обсягу користувачів, проектів та завдань. Обмеження масштабованості можуть впливати на продуктивність та швидкодію системи при збільшенні навантаження.

2.5 Припущення й залежності

Наявність стабільного та надійного Інтернет-з'єднання є важливою умовою для оптимальної роботи системи. Система взаємодіє з мережею для обміну даними, автентифікації, надсилання сповіщень тощо, тому постійне з'єднання з Інтернетом є необхідним. Впевненість в стабільності та швидкості Інтернет-з'єднання є важливою умовою для ефективної роботи системи.

Доступ до бази даних: Система налагоджується від доступу до бази даних для зберігання та отримання інформації про користувачів, проекти, завдання та інші дані. Наявність та налаштування бази даних виконуються через Microsoft SQL Server. Ця база даних є критичною для функціонування системи, тому важливо впевнитися в її належному функціонуванні та оптимальному налаштуванні.

3 КОНКРЕТНІ ВИМОГИ

3.1 Вимоги до зовнішніх інтерфейсів

3.1.1 Інтерфейс користувача

Інтерфейс користувача програмної системи рекомендації та відбору актуальних подій з back-end реалізацією розроблений з метою забезпечення зручності, ефективності та легкості використання для користувачів. Він пропонує інтуїтивно зрозуміле та привабливе середовище, яке дозволяє взаємодіяти з системою та виконувати необхідні завдання.

Інтерфейс користувача може бути представлений у вигляді веб-додатку або мобільного додатку, залежно від потреб користувачів та доступних платформ. Він має чітку структуру, навігаційні елементи, що дозволяють користувачам легко орієнтуватися та виконувати різноманітні дії.

Основні характеристики інтерфейсу користувача включають:

- інтуїтивність: Забезпечення легкого розуміння та використання для користувачів без додаткового навчання.
- ефективність: Максимізація продуктивності користувачів та зменшення часу на виконання завдань.
- привабливість: Створення привабливого та естетичного вигляду інтерфейсу для залучення користувачів.
- чітка структура: Організація інформації та функціональності для легкого доступу та сприяння логічному розташуванню елементів.
- навігаційні можливості: Доступні та зрозумілі елементи навігації для швидкого переміщення користувачів по системі.

Ці аспекти сприяють позитивному враженню від використання системи та підвищують задоволення користувачів взаємодією з нею.

3.1.2 Апаратний інтерфейс

Апаратний інтерфейс програмній системі рекомендації та відбору актуальних подій з back-end реалізацією включає необхідні апаратні компоненти та засоби, які

використовуються для забезпечення функціональності системи. Оскільки ця програма зазвичай працює в онлайн-середовищі та доступна через веб-браузер або мобільний додаток, система вимагає наявності серверів для зберігання та обробки даних. Це можуть бути фізичні сервери, віртуальні сервери або хмарні платформи, які забезпечують необхідні обчислювальні та ресурси для зберігання роботи системи.

3.1.3 Програмний інтерфейс

Програмний інтерфейс (API) програмної системи рекомендації та відбору актуальних подій з back-end реалізацією визначає набір правил та протоколів, за допомогою яких різні компоненти програми можуть взаємодіяти між собою. Він дозволяє зовнішнім системам або розробникам використовувати функціональні можливості та отримувати доступ до даних програмної системи.

Програмний інтерфейс може бути реалізований у вигляді RESTful API , який базується на використанні HTTP-протоколу для передачі даних між клієнтами та сервером. Це дозволяє здійснювати стандартизовану та просту взаємодію з системою за допомогою HTTP-запитів (GET, POST, PUT, DELETE) та обміну даними у форматі JSON або XML.

3.1.4 Комунікаційний протокол

Комунікаційний протокол в програмної системи рекомендації та відбору актуальних подій з back-end реалізацією визначає правила та формати обміну даними між різними компонентами системи. Він гарантує, що передача інформації між цими компонентами відбувається швидко, надійно та безпомилково.

Один з найпоширеніших комунікаційних протоколів, який може бути використаний в такій системі, - це HTTP (Hypertext Transfer Protocol). HTTP використовується для передачі даних через мережу Інтернет та побудований на базі клієнт-серверної архітектури.

НТТР визначає формат запитів та відповідей між клієнтом та сервером. Клієнт (наприклад, веб-браузер або мобільний додаток) робить запит до сервера, а сервер обробляє цей запит та повертає відповідь з необхідними даними.

3.1.5 Обмеження пам'яті

Обсяг оперативної пам'яті, доступний для виконання програм та зберігання даних, може бути обмеженим. Залежно від розміру доступної пам'яті, система може вміщувати обмежену кількість користувачів та обробляти лише обмежену кількість даних.

Наявність дискового простору для зберігання даних також може бути обмеженою. Дисковий простір використовується для зберігання файлів, баз даних, резервних копій та інших ресурсів. Обмеження дискового простору може впливати на масштабність системи та можливість зберігання великих обсягів даних.

3.1.6 Операції

Користувачі можуть шукати події та заходи що їх цікавлять. Це включає визначення назви, типу, місця, часу та інших важливих атрибутів події.

Система може надавати можливості для створення компаніям або угрупованням, що мають права на заходи, нові події та заходи. Також система надає можливість утворення звітності для організаторів.

3.2 Атрибути програмного продукту

3.2.1 Надійність

Надійність в програмній системі рекомендації та відбору актуальних подій з back-end реалізацією є важливою характеристикою, оскільки вона забезпечує безперебійну та стабільну роботу системи. Надійність означає, що система працює так, як очікується, і надійно обробляє запити користувачів, запобігаючи можливим збоям чи відмовам.

Для досягнення надійності, система повинна мати вбудований механізм для обробки помилок та винятків, які можуть виникати під час роботи. Це допомагає

виявляти та вирішувати проблеми, що впливають на роботу системи, забезпечуючи її стабільну та надійну функціональність.

Також важливо мати механізми резервного копіювання, що забезпечують збереження та захист даних системи. Це може включати регулярне резервне копіювання даних, створення резервних копій баз даних та інших важливих ресурсів, що гарантує можливість відновлення системи в разі втрати чи пошкодження інформації.

3.2.2 Доступність

Система повинна мати реактивний дизайн, який автоматично адаптується до різних типів пристроїв, таких як комп'ютери, планшети або смартфони. Це дозволяє користувачам зручно взаємодіяти з системою на будь-якому пристрої, незалежно від їхнього розміру екрану. Такий підхід сприяє покращенню зручності використання та задоволенню користувачів, забезпечуючи оптимальний інтерфейс на різних пристроях.

3.2.3 Безпека

Система повинна володіти механізмами аутентифікації для перевірки ідентичності користувачів та авторизації їх доступу до різних функцій системи. Це може включати використання паролів, рівнів доступу та управління правами користувачів.

З метою захисту конфіденційності даних, особливо під час їх передачі між клієнтом та сервером, використовується шифрування. Використання протоколів шифрування, таких як SSL (Secure Sockets Layer) або TLS (Transport Layer Security), сприяє забезпеченню безпеки під час передачі даних.

Система також повинна мати заходи безпеки для захисту від зламу та зловмисницьких атак, таких як введення шкідливого коду, SQL-ін'єкції, переповнення буфера тощо. Це може включати застосування механізмів фільтрації введення користувачів, валідацію даних та захист від переповнення, що сприяє забезпеченню високого рівня безпеки системи.

3.2.4 Супроводжуваність

Система повинна мати належну документацію, яка точно описує її архітектуру, конфігурацію, процедури установки, налаштування та супроводження. Це сприяє швидкому розумінню та вирішенню проблем розробниками та адміністраторами системи, а також дозволяє внесення необхідних змін.

Крім того, система повинна мати механізми для регулярних оновлень та встановлення патчів з метою виправлення відомих помилок, усунення вразливостей та покращення функціональності. Це забезпечує надійність та стійкість системи, а також дозволяє впровадження нових функцій та вдосконалень.

3.2.5 Переносимість

Переносимість в програмній системі, щодо рекомендацій та відбору актуальних подій у back-end реалізації, передбачає здатність системи працювати на різних платформах та середовищах без значних змін. Це досягається шляхом використання платформи-незалежних технологій, стандартизованих фреймворків та уникання глибокої залежності від конкретних платформ або СКБД. Такий підхід гарантує гнучкість та ефективність у розгортанні системи на різних середовищах без великих зусиль.

3.2.6 Продуктивність

Висока продуктивність у програмній системі рекомендації та відбору актуальних подій з реалізацією back-end є суттєвим компонентом і включає в себе оптимізацію коду, масштабування та ефективне використання ресурсів. Розробка коду, який оптимізовано виконує завдання, разом з масштабуванням системи для ефективного розподілу навантаження та правильного використання ресурсів, сприяє високій продуктивності системи та задоволенню потреб користувачів.

3.2.7 Вимоги бази даних

Для програмної системи рекомендації та підбору актуальних подій із back-end реалізацією вимоги до бази даних включають в себе належну структуру, нормалізацію даних та підтримку необхідних запитів та операцій. Забезпечення безпеки даних є пріоритетом і вимагає використання механізмів авторизації, шифрування та контролю доступу. Оптимізація бази даних для досягнення високої швидкодії та ефективності виконання запитів також є ключовою вимогою.

ДОДАТОК Г

Тези XXVIII міжнародного молодіжного форуму «Радіоелектроніка та молодь у
XXI столітті»

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
РАДІОЕЛЕКТРОНІКИ

МАТЕРІАЛИ XXVIII МІЖНАРОДНОГО МОЛОДІЖНОГО
ФОРУМУ

**«РАДІОЕЛЕКТРОНІКА ТА МОЛОДЬ
У XXI СТОЛІТТІ»**

16 – 18 квітня 2024 р.

Том 6

**КОНФЕРЕНЦІЯ
«ІНФОРМАЦІЙНІ ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ»
INFORMATION INTELLIGENT SYSTEMS**

Харків 2024

28-й Міжнародний молодіжний форум «Радіоелектроніка та молодь у XXI столітті». Зб. матеріалів форуму. Т. 6., – Харків: ХНУРЕ. 2024. – 958 с.

У збірнику представлено матеріали доповідей учасників 28-го Міжнародного молодіжного форуму «Радіоелектроніка та молодь у XXI столітті».

Для науковців, викладачів, практичних працівників, студентів, а також широкого кола читачів, які цікавляться цією проблематикою.

Відповідальність за зміст поданого матеріалу несе його автор.

Видання підготовлено факультетом комп'ютерних наук Харківського національного університету радіоелектроніки

61166, Україна, Харків, просп. Науки, 14
тел./факс: (057) 7021397

E-mail: mref21@nure.ua

ISBN ISBN 978-966-659-396-5
DOI [10.30837/IYF.IIS.2024](https://doi.org/10.30837/IYF.IIS.2024)

© Харківський національний
університет радіоелектроніки
(ХНУРЕ), 2024

Програмний комітет конференції

Федорович О.Є.	д.т.н., проф., зав. каф. Комп'ютерних наук та інформаційних технологій (КНІТ), Національний аерокосмічний університет ім. М.Є. Жуковського "Харківський авіаційний інститут", Лауреат Державної премії України.
Субботін С.А.	д.т.н., проф., зав. каф. Програмних засобів, Запорізький національний технічний університет, Україна.
Петренко М.Г.	д.т.н., проф., Інститут кібернетики імені В.М. Глушкова НАН України.
Стасюк О.І.	д.т.н., проф. Державний економіко-технологічний університет транспорту, Україна.
Єрохін А.Л.	проф., декан ф-ту ХНУРЕ, м. Харків, Україна.
Філатов В.О.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Петров К.Е.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Дудар З.В.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Гребеннік І.В.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.
Дейнеко Ж.В.	проф., зав. каф. ХНУРЕ, м. Харків, Україна.

УДК 004.89

EVENTIFY: ІННОВАЦІЙНИЙ ПІДХІД ДО ОРГАНІЗАЦІЇ ТА ПОШУКУ ЗАХОДІВ З ВИКОРИСТАННЯМ ШІ

Сафощин В. В., Хамінов І. О., Дегтяр В. Е.

Науковий керівник – к.т.н., доц. Побіженко І. О.

Харківський національний університет радіоелектроніки, каф. ПІ
м. Харків, Україна

e-mail: volodymyr.safoshyn@nure.ua, illia.khaminov@nure.ua,

e-mail: vladyslav.dehtiar@nure.ua

In an era of rapid cultural evolution, the challenge of effectively organizing and selecting events that cater to individual preferences has become paramount. "Eventify" revolutionizes event organization in the digital age by leveraging artificial intelligence to tailor cultural event recommendations to individual preferences. This platform not only makes finding events effortless but also signifies a new chapter in personalized cultural experiences, ensuring users are matched with events that resonate with their interests. "Eventify" exemplifies the transformative potential of artificial intelligence [1] (AI) in enriching cultural engagement and streamlining the event selection process.

В епоху стрімкого розвитку культурного життя, актуальним стає питання ефективної організації та відбору заходів, які б відповідали особистим уподобанням кожної людини. Доступними рішеннями є різні телеграм канали, які повідомляють користувачів про різні події. В таких джерелах інформація про події не згрупована за видом та розташуванням, тож користувач може загубити цікаві йому події поміж багатьох інших. Також є різні сайти, що пропонують тільки специфічний спектр подій і не охоплюють всі вподобання користувача, що призводить до того що потрібно використовувати багато різних джерел. В цьому контексті сервіс "Eventify" пропонує революційний підхід, що базується на використанні передових технологій штучного інтелекту (ШІ) для аналізу великих обсягів даних про культурні події, їх учасників та переваги користувачів. Удосконалюючи свою місію щодо пошуку та організації культурних заходів, платформа "Eventify" залучатиме передові методи штучного інтелекту, які включають великі мовні моделі [2] (LLMs), ChatGPT[3].

Дослідження спрямоване на створення інноваційної програмної системи, яка за допомогою веб- та мобільного додатків пропонуватиме користувачам персоналізовані рекомендації щодо різноманітних заходів, виходячи з їхніх переваг. Це дозволить користуватися однією платформою для пошуку всіх видів культурних заходів, а не багатьма різними сторонніми ресурсами.

Впродовж дослідження аналогічний рішень нашого додатку, було виявлено те, що жоден з них не використовує ніякі засоби для оптимізації та комфорту у виборі культурних заходів. Враховуючи це, ми прийшли до

висновку, що використання штучного інтелекту стане ідеальним рішенням для покращення користувацького досвіду.

Великі мовні моделі представляють собою клас алгоритмів машинного навчання, здатних аналізувати та генерувати природну мову [4] з високим рівнем складності та точності. У контексті "Eventify" використовуються LLMs для обробки об'ємних текстових даних, пов'язаних із культурними заходами.

ChatGPT – це одна з передових LLMs, розроблена OpenAI. Ця модель базується на технології трансформерів і здатна генерувати природні відповіді в контексті діалогу з користувачем [5]. Під час розробки нашого додатку ми обрали цю LLM як ключовий компонент, оскільки він пропонує унікальні переваги для покращення користувацького досвіду у виборі культурних подій. ChatGPT відіграє ключову роль у нашому додатку, оскільки здатний до глибокого аналізу діалогів, що дозволяє вловлювати нюанси запитів користувачів та відповідати на них з надзвичайною точністю. Також він здатний до поглибленої взаємодії з користувачем, що буде сприяти більш детальному розумінню їхніх уподобань, що дозволяє надавати влучні рекомендації щодо заходів, які відповідають їхнім інтересам.

Отже, наш проект підкреслює значення інтеграції штучного інтелекту в області організації заходів, демонструючи, як сучасні технології можуть радикально змінити взаємодію між організаторами та учасниками. Також за результатами нашого дослідження було проаналізовано усі доступні рішення цієї проблеми та створено інноваційний підхід її вирішення шляхом розробки додатку "Eventify" з використанням передових технологій штучного інтелекту.

Список використаних джерел:

1. Barstow, David. "Artificial intelligence and software engineering." *Exploring artificial intelligence*. Morgan Kaufmann, 1988. 641-670.
2. Chang, Yupeng, et al. "A survey on evaluation of large language models." *ACM Transactions on Intelligent Systems and Technology* (2023).
3. Wu, Tianyu, et al. "A brief overview of ChatGPT: The history, status quo and potential future development." *IEEE/CAA Journal of Automatica Sinica* 10.5 (2023): 1122-1136.
4. Erdem, Erkut, et al. "Neural natural language generation: A survey on multilinguality, multimodality, controllability and learning." *Journal of Artificial Intelligence Research* 73 (2022): 1131-1207.
5. Sharonova, N., Kyrychenko, I., Gruzdo, I., Tereshchenko, G. (2022) Generalized Semantic Analysis Algorithm of Natural Language Texts for Various Functional Style Types *CEUR Workshop Proceedings*, 2022, 3171, pp. 16–26.

Гриб А. С., 714
 Григор'єв О. В., 945
 Гринишина С. О., 923
 Гриньов С. А., 121
 Гриньова О. Є., 13, 16, 24
 Гриньова О.Є., 5, 7
 Грiшаєва А. М., 64
 Громенко А. І., 158, 160
 Груздо І. В., 116, 502, 846
 Гулієв Н. Б., 508
 Гуркін В. С., 894

Д

Давиденко А. Л., 160
 Данилов А. Д., 456
 Данілейко С. І., 730
 Двугрошев А. О., 369
 Дегтяр В. Е., 544
 Дейнеко А. О., 35
 Дейнеко Ж. В., 948, 950
 Дем'яненко М. С., 487
 Демиденко С. О., 246
 Демченко М. О., 493
 Денисюк В. М., 336
 Дергачова Д. К., 140
 Деркач К. Ю., 481
 Дехадрай Д. Р., 796
 Дідусь О. П., 850
 Добудько А. М., 902
 Домніч Д. В., 840
 Донець Д. С., 338
 Драконова О. О., 348
 Дробицький Д. С., 124
 Дубок В. Ю., 425
 Дуванов А. К., 630
 Дудар З. В., 399
 Дудка М. В., 794
 Дукельська К.Б., 24
 Дюжев М. Л., 162

Є

Євланов М. В., 168, 275, 284
 Євменкін Д. К., 164
 Єгорова І. М., 925
 Єлтишев П. І., 861
 Єльчанинов Д. Б., 74, 80
 Ємельянов А. В., 695
 Єрохін А. Л., 326, 329
 Єрохін М. А., 166
 Єрошенко С. О., 317, 605

Ж

Жаркіх С. Є., 16
 Жемчужний Р. І., 869
 Женило К.О., 5
 Жирко К. В., 158, 160
 Жмур Д. М., 112

З

Забийворота М. А., 892
 Заворіна М.А., 30
 Загнойко І. Ю., 421
 Задніпровський Д. Б., 168
 Задорожний А. Ю., 568
 Запара О. С., 300
 Заполочний А. Д., 170
 Звєгінцев А. В., 532
 Златкін С. С., 173

І

Іванов В. Г., 581, 583, 618,
 650, 667, 673, 704, 800
 Іванов Є. О., 130
 Іванова А. І., 175
 Іванова О. С., 44
 Ігнатюк Є. О., 160
 Ільїн І. О., 810
 Імангулова З. А., 133, 644,
 663, 755, 796, 806, 812,
 848, 855
 Іпполітова В. Є., 920
 Ісаєнко С. С., 177
 Іткін Д. О., 180

К

Казимов Л. Б., 183
 Кайданюк Г. С., 450
 Калайда Н. С., 697, 734, 746,
 766, 784, 810, 892, 900
 Калита Н. І., 738, 836
 Калінін Д. В., 757
 Калінін Д.В., 10
 Калініченко О. Ю., 363
 Кальний С. А., 865
 Калюжний О. Д., 702
 Камсюк Д. О., 830
 Канінець А. А., 945
 Кардаш Д. М., 52
 Каряка В. В., 185
 Кастиркін Д. Р., 880
 Каук В. І., 353, 366, 428, 472,
 493
 Кащенко Ю. Є., 656
 Кириченко І. В., 112
 Кирсанов О. О., 59
 Кієнко Д. В., 187
 Кієу Куанг Хієп, 753
 Кікоть М. С., 189
 Кісельгова М. Є., 342
 Кітов А. В., 54
 Кіценко Ю. О., 435
 Клімова І. М., 205, 726, 882
 Клішов М. Р., 886
 Ключко Є. С., 654
 Клюванський Є. Г., 194

Коваленко А. І., 587, 589,
 591, 595, 599, 601, 608,
 620, 675, 871

Коваленко О. А., 635
 Коваленко О. О., 933
 Коваль О. О., 677
 Ковальов І. М., 768
 Ковальов М. М., 667
 Козирев А. Д., 437
 Козорог І. Г., 818
 Колендовська М. М., 933,
 938

Колесник Л. В., 699
 Коломощь К. В., 710
 Коломойцев П. А., 326
 Комзолов М. О., 786
 Комін А. С., 102
 Кондратьєв О. В., 914
 Коновалова М. Д., 898
 Константинов Б. С., 855
 Копейчиков І. Ю., 197
 Коптілов Н. С., 632
 Корзун В. Р., 741
 Коріненко В. Д., 842
 Косенко Б. А., 413
 Котелевець К. А., 107
 Котенко І. І., 571
 Кошарний Є. Ю., 612
 Кошель В.О., 42
 Кравець Н. С., 345, 387, 405,
 425, 440
 Кравцов Д. О., 526
 Кравченко В. Д., 744
 Кравченко Є. О., 396
 Кравченков Т. П., 583
 Кривенко С. А., 59
 Круц О. О., 900
 Крюкова М. М., 925
 Кубай Р. В., 475
 Кудрявський Д. А., 681
 Кудрявцева М. С., 10, 140,
 183, 248, 757
 Кузнецов Р. О., 340
 Кузьміна П. О., 896
 Куліш Є. І., 217
 Кулішова Н. Є., 905, 907, 920
 Кулішова Н.Є., 40
 Кульмінський Я. К., 479
 Купенко М. І., 200
 Кучеренко Д., 931

Л

Лавриненко Р. М., 91
 Лавриненко С. Р., 86, 91
 Лавріненко В. В., 712
 Лановий О. Ф., 303, 541, 565
 Ларченко Л. В., 290
 Ларченко С. О., 565
 Латішев О. О., 581
 Лахтін В. В., 105

ДОДАТОК Д

Код для управління авторизацією користувача

```

1. import SwiftUI

2. struct ContentView: View {
3. @State private var email = ""
4. @State private var password = ""
5. @State private var wrongEmail: Float = 0
6. @State private var wrongPassword: Float = 0
7. @State private var showingLoginScreen = false
8. @State private var isLoggedIn = false
9. @State private var showResetPasswordModal = false
10.     @State private var showRegistrationModal = false
11.     @State private var pulsate = false
12.     @ObservedObject var user = User(name: "", email: "", phone:
    "", password: "", userID: "")

13.     var body: some View {
14.     NavigationView {
15.     ZStack {
16.     Color.green
17.     .ignoresSafeArea()
18.     .onTapGesture {
19.     showRegistrationModal = false
20.     }
21.     Circle()
22.     .scale(pulsate ? 1.5 : 2.0)
23.     .foregroundColor(.white.opacity(0.2))
24.     .animation(Animation.easeInOut(duration:
    2.0).repeatForever(autoreverses: true))
25.     .onAppear {
26.     self.pulsate.toggle()
27.     }
28.     Circle()
29.     .scale(1.2)
30.     .foregroundColor(.white)

31.     VStack {
32.     Text("EVENTIFY")
33.     .font(.largeTitle)
34.     .bold()
35.     .padding()

```

```
36.     .foregroundColor(.white)
37.     .background(Color.green.opacity(0.7))
38.     .frame(width: 200, height: 60)
39.     .cornerRadius(30)

40.     TextField(NSLocalizedString("email_placeholder", comment:
    ""), text: $email)
41.     .padding()
42.     .keyboardType(.emailAddress)
43.     .frame(width: 300, height: 50)
44.     .background(Color.black.opacity(0.3))
45.     .cornerRadius(10)
46.     .border(.red, width: CGFloat(wrongEmail))

47.     SecureField(NSLocalizedString("password_placeholder",
    comment: ""), text: $password)
48.     .padding()
49.     .frame(width: 300, height: 50)
50.     .background(Color.black.opacity(0.3))
51.     .cornerRadius(10)
52.     .border(.red, width: CGFloat(wrongPassword))

53.     Button(NSLocalizedString("login_button", comment: "")) {
54.         authenticateUser(email: email, password: password)
55.     }
56.     .foregroundColor(.white)
57.     .frame(width: 300, height: 50)
58.     .background(.green)
59.     .cornerRadius(10)
60.     .padding(.top, 10)

61.     HStack {
62.         Button(NSLocalizedString("registration_button", comment:
    "")) {
63.             showRegistrationModal.toggle()
64.         }
65.         .foregroundColor(.blue)

66.     }
67.     .frame(width: 300)
68.     .padding(.top, 10)
```

```
69.     NavigationLink(destination: MainView(), isActive:
      $isLoggedIn) {
70.     EmptyView()
71.     }
72.     .hidden()
73.     }
74.     .padding()
75.     if showRegistrationModal {
76.     GeometryReader { geometry in
77.     Color.black.opacity(0.5)
78.     .ignoresSafeArea()
79.     .onTapGesture {
80.     showRegistrationModal = false
81.     }

82.     RegistrationModal(isPresented: $showRegistrationModal)
83.     .frame(width: 400, height: 500)
84.     .background(Color.white)
85.     .cornerRadius(20)
86.     .shadow(radius: 10)
87.     .overlay(
88.     RoundedRectangle(cornerRadius: 20)
89.     .stroke(Color.gray, lineWidth: 1)
90.     )
91.     .position(x: geometry.size.width / 2, y:
      geometry.size.height / 2)
92.     .onTapGesture {
93.     }
94.     }
95.     .transition(.scale)
96.     }
97.     }
98.     }
99.     .navigationBarHidden(true)
100.    .navigationBarBackButtonHidden(true)
101.    }

102.    func authenticateUser(email: String, password: String) {
103.    let loginData = LoginData(email: email, password: password)

104.    sendLoginRequest(loginData) { result in
105.    DispatchQueue.main.async {
106.    switch result {
107.    case .success(let response):
```

```
108.     print("Login successful: Token: \(response.token),
        Expiration: \(response.expiration)")
109.     user.updateUserID(from: response.token)
110.     fetchUserProfile(userID: user.userID)
111.     isLoggedIn = response.result
112.     case .failure(let error):
113.     print("Login failed: \(error.localizedDescription)")
114.     wrongEmail = 2
115.     wrongPassword = 2
116.     }
117.     }
118.     }
119.     }
120.     func fetchUserProfile(userID: String?) {
121.     guard let userID = userID else { return }
122.     let url = URL(string: "https://eventify-
        backend.azurewebsites.net/api/Profile/get-
        profile?id=\(userID)")!
123.     var request = URLRequest(url: url)
124.     request.httpMethod = "GET"

125.     let task = URLSession.shared.dataTask(with: request) {
        data, response, error in
126.     if let error = error {
127.     print("Failed to fetch user profile:
        \(error.localizedDescription)")
128.     return
129.     }

130.     if let response = response as? HTTPURLResponse {
131.     print("HTTP Status Code: \(response.statusCode)")
132.     }

133.     guard let data = data else {
134.     print("No data received.")
135.     return
136.     }

137.     if let jsonString = String(data: data, encoding: .utf8) {
138.     print("Raw JSON String: \(jsonString)")
139.     }

140.     do {
```

```
141.     if let json = try JSONSerialization.jsonObject(with: data,  
    options: []) as? [String: Any] {  
142.     DispatchQueue.main.async {  
143.     testUser.updateUserProfile(from: json)  
144.     }  
145.     print("User Profile Data: \(json)")  
146.     }  
147.     } catch {  
148.     print("Failed to parse user profile data:  
    \(error.localizedDescription)")  
149.     }  
150.     }  
  
151.     task.resume()  
152.     }  
153.     }
```