

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій

УДК

«ПОГОДЖЕНО»

Декан факультету
інформаційних технологій

Глазунова О.Г., д.п.н., професор

_____ 2024 р.

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Завідувач кафедри комп'ютерних наук

Голуб Б.Л., к.т.н., доцент

_____ 2024 р.

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему «Система підтримки прийняття рішень при організації наукових конференцій»

Спеціальність _____ 122 «Комп'ютерні науки» _____
(код і назва)

Освітня програма _____ «Інформаційні управляючі системи і технології» _____
(назва)

Орієнтація освітньої програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)

Гарант освітньої програми

_____ (науковий ступінь та вчене звання) _____ (підпис) _____ (ПІБ)

Керівник магістерської кваліфікаційної роботи

Кандидат економічних наук, ст. викладач _____ Ніколаєнко Дмитро Володимирович
(науковий ступінь та вчене звання) (підпис) (ПІБ)

Виконав

_____ (підпис) **Канашков Олександр Сергійович**
(ПІБ студента)

КИЇВ-2024

ЗМІСТ

ЗМІСТ	2
ВСТУП	3
Розділ 1. Системний аналіз предметної області.....	5
1.1 Опис та методології системного аналізу.....	6
1.2 Аналіз наявних рішень	10
1.3 Аналіз вимог.....	11
1.4 Структурно-функціональний аналіз.....	13
1.5. Об'єктно-орієнтований аналіз.....	20
РОЗДІЛ 2. Проектування інформаційної системи	22
2.1 Об'єктно-орієнтоване проектування системи.....	22
2.1.1 Діаграма класів	24
2.1.2 Діаграма послідовності.....	26
Розділ 3. Розробка системи.....	30
ВИСНОВКИ.....	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	40

ВСТУП

Наукова конференція - це захід, на якому дослідники (викладачі, студенти, наукові співробітники та інші) представляють та обговорюють свої роботи та результати досліджень. Кожне дослідження повинно мати певну теоретичну та/або практичну цінність для науки та суспільства. Наукові конференції є найважливішим елементом сучасної наукової комунікації, надаючи дослідникам платформу для обміну ідеями, обговорення результатів та формування наукових зв'язків для подальшої спільної роботи.

Насамперед, виступаючи на конференції, можна з'ясувати, наскільки актуальним та цінним є ваші дослідження для науки, господарства, економіки, конкретної галузі чи суспільства загалом. Беручи участь як слухач або аналізуючи вже видані збірки тез доповідей, можна з'ясувати сучасні тенденції та стан проблеми, що вирішується. Крім того, апробація на конференціях бакалаврських та магістерських досліджень є обов'язковою умовою допуску до захисту диплома.

Конференції можуть відбуватися:

Очно – учасники приїжджають безпосередньо на місце проведення конференції, роблять доповіді та беруть участь у різноманітних заходах, які можуть відбуватися у рамках конференції. Це можуть бути круглі столи, присвячені окремим питанням, майстер-класи, воркшопи та інше. На таких конференціях, як правило, спочатку проводиться пленарне засідання, на якому доповідають найкращі доповіді, а потім учасники працюють у рамках тематичних секцій. Приємним бонусом очної конференції є кава-брейки, під час яких можна у неформальній обстановці поспілкуватись із учасниками конференції.

Заочно – цей формат підходить для учасників, які з якихось причин не можуть приєднатися до конференції в момент її проведення або мають проблеми

з публічними виступами. Проте вони можуть отримати всі матеріали конференції та опублікувати свої тези у збірнику конференції.

Дистанційно – ця форма особливо актуальна за умов сучасних карантинних обмежень, але існувала задовго до пандемії. Під час проведення конференції у такому форматі учасники роблять доповіді, використовуючи для спілкування онлайн платформи, визначені організаторами. Перевагою такого формату є те, що список її учасників не залежить від їхнього територіального розташування або інших факторів, що заважають їм бути особисто. Однак у такому форматі, на жаль, майже зникає неформальне спілкування.

У нинішніх умовах, коли значна частина наукової діяльності вимушено перейшла до онлайн-формату, актуальною стає розробка систем, що спрощують процес організації та участі у конференціях. Зазвичай організація таких заходів супроводжується значними труднощами, пов'язаними із залученням учасників, управлінням процесами реєстрації, модерацією контенту та аналізом результатів. Відсутність сучасних цифрових інструментів часто призводить до недостатнього використання даних про минулі заходи та переваги учасників, що обмежує потенціал майбутніх наукових конференцій.

Таким чином, особливо актуальною стає розробка та впровадження системи підтримки прийняття рішень (СППР), здатної автоматизувати ключові процеси при організації наукових конференцій, що сприятиме цифровізації наукового середовища та підвищенню ефективності організації наукових заходів.

Розділ 1. Системний аналіз предметної області

У цьому розділі проведено аналіз предметної області і розглянуто сутність процесів організації конференцій, а також аналіз наявних рішень у цій сфері для виявлення їхніх переваг та недоліків з метою подальшого вдосконалення.

Об'єкт дослідження: процес організації та проведення наукових конференцій.

Предмет дослідження: система підтримки прийняття рішень для оптимізації процесів організації наукових конференцій.

Мета дослідження: оптимізувати організаційні процеси та зробити їх більш ефективними з використанням сучасних технологій та прогнозуючих алгоритмів.

1.1 Опис та методології системного аналізу

«Системний аналіз - це методологія теорії систем, що полягає в дослідженні будь-яких об'єктів, що представляються в якості систем, проведенні їх структуризації і подальшого аналізу. Головна особливість системного аналізу полягає в тому, що він включає в себе не тільки методи аналізу (від грец. analysis - розчленування об'єкта на елементи), а й методи синтезу (від грец. synthesis - з'єднання елементів в єдине ціле). [1] »

«Головна мета системного аналізу - виявити і усунути невизначеність при вирішенні складної проблеми на основі пошуку найкращого рішення з існуючих альтернатив. [1] »

В основі методології системного аналізу лежать операції кількісного порівняння і вибору альтернатив у процесі прийняття рішення, що підлягає реалізації. «Якщо вимогу критеріїв якості альтернатив виконано, то можуть бути отримані їх кількісні оцінки. [2] »

«Системний аналіз можна представити у вигляді сукупності основних логічних елементів:

- Мета дослідження - вирішення проблеми та отримання результату;
- Ресурси - наукові засоби вирішення проблеми (методи);
- Альтернативи - варіанти рішень і необхідність вибору одного з декількох рішень;
- Критерії - засіб (ознака) оцінки вирішуємості проблеми;
- Модель створення нової системи. [1] »

«Якщо розташовувані ресурси не можуть забезпечити реалізацію поставленої мети, то ми отримаємо не плановані результати. Мета - це і є бажаний результат. Тому для реалізації цілей повинні бути вибрані відповідні ресурси. Якщо ресурси обмежені, то треба коригувати мету, тобто планувати ті результати, які можна отримати при даному наборі ресурсів. Тому формулювання цілей в інноваційній діяльності повинна мати конкретні параметри. [2] »

«Характерні особливості системного аналізу такі:

- Оскільки здебільшого ухвалені рішення стосуються окремих елементів системи, то при вирішенні задач обов'язково враховуються взаємозв'язки цього елемента з іншими і загальна мета системи (системний підхід).
- Аналіз спочатку здійснюється для всього комплексу проблем і зводиться до рівня їх складових. Дослідження проводяться за допомогою методів, що використовувалися для їх вивчення;
- Першочергове значення належить таким чинникам, як вартість і якість праці;
- У багатьох випадках уже дані аналізу показують шлях до очевидного рішення, проте саме рішення доводиться ухвалювати;
- Системний аналіз не замінює логічних думок, а є лише допоміжним елементом. Він дає можливість визначити ті області, де може бути використана пропозиція, і визначити кожний із можливих варіантів рішення, які розглядає керівник;
- Передбачає обов'язкове використання ПК; у деяких випадках вони можуть бути використані як технічні засоби. [3] »

Необхідно:

- Провести системний аналіз процесів організації та проведення наукових конференцій;
- Сформулювати вимоги до системи підтримки прийняття рішень, яка спрямована на автоматизацію процесів організації наукових конференцій;
- Реалізувати алгоритми, що забезпечують прогнозування відвідуваності та рекомендації тегів для наукових конференцій, що створюються, а також персоналізовані рекомендації конференцій для користувачів;
- Розробити механізм генерації аналітичних звітів за підсумками минулих конференцій;
- Побудувати архітектуру системи, яка включатиме перелічені вище пункти;
- Оцінити ефективність розробленої системи;
- Сформулювати висновки та отримані результати для подальшого покращення процесів організації та проведення наукових конференцій.

Процес організації наукових конференцій складається з кількох ключових етапів, кожен з яких може бути значно оптимізований за допомогою сучасних інформаційних технологій:

- Планування конференції;
- Визначення тематики та основних напрямків заходу;
- Перевірка та модерація заявок;
- Створення розкладу;
- Реєстрація учасників та обробка їхніх даних;
- Інформування учасників про дату, час та формат проведення;
- Надсилання нагадувань про участь;
- Проведення конференції;
- Аналіз та оцінка результатів;
- Збір даних про відвідуваність, аналіз активності учасників;
- Генерація аналітичних звітів;

- Використання отриманих даних для подальшого вдосконалення майбутніх заходів.

Автоматизація цих процесів може значно підвищити ефективність організації та проведення наукових конференцій, а також полегшити взаємодію з підписниками.

1.2 Аналіз наявних рішень

Для досягнення цілей даного дослідження був проведений аналіз існуючих систем, які надають схожий функціонал. Було розглянуто комерційні платформи, що спеціалізуються на конференціях та плануванні заходів.

EasyChair [4] та ConfTool [5] надають інструменти для керування заявками на конференції, реєстрації учасників та публікації матеріалів, однак мають обмежену функціональність щодо прогнозування відвідуваності та персоналізованих рекомендацій.

Whova [6] пропонує інструменти для проведення віртуальних заходів, але основний акцент робиться на взаємодії з учасниками в режимі реального часу, без глибокого аналізу даних минулих конференцій.

У дослідженнях, присвячених автоматизаціям заходів, основна увага приділяється питанням оптимізації розкладу, автоматизації обробки заявок та системам підтримки прийняття рішень.

Дослідження в цій сфері здебільшого стосуються автоматизації планування заходів та роботі з системами підтримки прийняття рішень, але рідко включають функціонал прогнозування та персоналізованих рекомендацій на основі машинного навчання. «Традиційні підходи до вирішення завдань планування використовують імітаційні моделі, аналітичні моделі, евристичні підходи чи комбінацію цих методів. [7] »

Висновок: більшість рішень не мають функціоналу прогнозування кількості учасників на основі історичних даних, а також персоналізованих рекомендацій для користувачів. Крім того, відсутній механізм аналітики результатів конференцій в розрізі учасників заходу. Тому запропонована система має потенціал для покращення ефективності цих процесів.

1.3 Аналіз вимог

Одним з основних етапів проведення системного аналізу є початковий етап проведення досліджень внутрішнього та зовнішнього середовища для формування вимог до системи. На даному етапу основною задачею є сформулювати ряд вимог та тверджень до даної системи, які могли б максимально точно описати систему, визначити її роль та цілі, які мають бути досягнені за допомогою розробки даної системи та в цілому сформулювати вектор, в якому повинна вестися подальша розробка та подальше проектування. Даний етап є надзвичайно важливим, адже саме на основі нього надалі й буде будуватися проектування системи.

Користувач	Вимоги
Гість	Перегляд наявних конференцій, пошук конференцій, реєстрація
Зареєстрований користувач	Перегляд наявних конференцій, пошук конференцій, створення конференції, налаштування сповіщень, авторизація
Адміністратор	Модерація запитів, перегляд наявних конференцій, створення конференції, підтвердження конференцій, налаштування сповіщень, авторизація

Табл. 1. Групи користувачів

Функціональність:

- Можливість створення, редагування та видалення конференцій;
- Пошук конференцій за критеріями, що значно полегшує навігацію для користувачів.

Надійність:

- Доступ до створення та редагування конференцій можуть мати лише зареєстровані користувачі, що запобігає можливості неконтрольованого створення подій сторонніми особам;
- Підтвердження та публікація конференцій можливі лише для адміністраторів, що дозволить уникнути помилкових або небажаних публікацій.

Додаткові функції:

- Нагадування про заплановані події на електронну пошту.
- Інтерактивні підказки на головній сторінці, які рекомендують найпопулярніші та персоналізовані конференції на основі уподобань користувача.

Захист даних користувачів:

- Паролі користувачів мають зберігатись виключно у вигляді хешу;
- Уся інформація зберігається на сервері, що забезпечує захист від втрати даних.

Інтерфейс та сумісність:

- Система повинна бути реалізована як веб-ресурс та працювати в актуальних браузерах;
- Система повинна мати простий і зрозумілий інтерфейс для швидкого доступу до основних функцій.

1.4 Структурно-функціональний аналіз

«Структурно-функціональний метод — це підхід в описі і поясненні систем, при якому досліджуються їхні елементи і залежності між ними в рамках єдиного цілого; окремі соціальні явища виконують визначену функцію в підтримці і зміні соціальної системи. Кожен елемент цієї структури виконує визначені функції, що задовольняють потреби системи. Діяльність елементів системи програмується загальною структурною організацією, займаними ними позиціями і виконуваними ролями. Структурно-функціональний метод є однією з найважливіших форм застосування цього методу в дослідженні управлінських явищ і процесів. Сутність його полягає у розділенні складного об'єкта на складові частини, вивченні зв'язків між ними та у визначенні притаманних їм специфічних функцій (ролей), спрямованих на задоволення відповідних потреб системи управління персоналом з урахуванням цілісності останньої та її взаємодії із зовнішнім середовищем. [8] »

«Структурний аналіз пропонує логічну графічну модель потоку інформації, поділяючи системи на модулі, що показують рівні, які піддаються керуванню та деталізації. [9] »

Усі основні методології структурного підходу базуються на низці спільних принципів. Серед них особливу увагу приділяють двом фундаментальним:

- Принцип "розділяй і володарюй": полягає в розв'язанні складних завдань шляхом їх поділу на численні невеликі незалежні частини, які легко зрозуміти та вирішити;
- Принцип ієрархічної організації: передбачає структурування складових елементів задачі у вигляді ієрархічної, деревоподібної моделі, де кожен рівень додає нові деталі.

Виділення двох базових принципів не означає, що останні принципи є другорядними, оскільки ігнорування будь-яке з них може привести до непередбачуваних наслідків (у тому числі й до провалу всього проекту).

«Основними з цих принципів є такі:

- принцип абстрагування - полягає у виділенні істотних аспектів системи та ігнорування неістотних;
- принцип формалізації - полягає в необхідності строгого методичного підходу до розв'язання проблеми;
- принцип несуперечності - полягає в обґрунтованості й узгодженості елементів;
- принцип структуризації даних - полягає в тому, що дані мають бути структуровані і ієрархічно організовані; [10] »

«Логічна модель інформаційної системи повинна максимально повно показувати вимоги користувача і бути незалежною від подальшої фізичної реалізації. Вона призначена для розроблення сукупності вимог до програмного забезпечення. [9] »

FDD (Functional Decomposition Diagrams) — це одна з методологій розробки програмного забезпечення, що належить до гнучких підходів (agile). Вона спрямована на об'єднання найкращих практик, визнаних у сфері розробки програмного забезпечення, із фокусом на ключову функціональність (властивості), яка має значення для замовника. Основна мета цього підходу — створення працюючого програмного продукту в чітко визначені строки, дотримуючись систематичного підходу.

Основою цієї діаграми є метод функціональної декомпозиції, який використовується для аналізу складних процесів через розбиття їх на окремі елементи. У такому контексті функція розглядається як частина загального процесу, а декомпозиція дозволяє розкласти цей процес на простіші для розуміння компоненти. Ідея функціональної декомпозиції походить із математики, де вона застосовується для аналізу взаємозв'язків між компонентами, що утворюють функціональну залежність, із метою спрощення

вихідної функції та її реконструкції. Таким чином, метод допомагає перетворювати складні системи на зрозуміліші моделі.

Враховуючи основи FDD і функціональної декомпозиції, ці методи можна використати для моделювання системи. Для початку декомпозиції необхідно виділити функціонал системи. У цьому допомагає діаграма прецедентів, яка аналізує взаємодію користувачів із системою. Грамотно побудована діаграма дозволяє визначити групи користувачів (акторів) і дослідити їхню взаємодію із системою. Це є критично важливим на початкових етапах, оскільки допомагає зрозуміти, хто саме буде користувачем системи, які функціональні вимоги слід реалізувати, і які елементи необхідно врахувати під час декомпозиції для створення FDD-діаграми.

На цій діаграмі відображено декомпозицію основних процесів системи. Для даної інформаційної системи були визначені такі ключові процеси: управління користувачами, підбір конференцій та контроль підготовки. **Діаграма FDD** дозволяє деталізувати ці процеси на підпроцеси, що забезпечує більш глибоке розуміння функціоналу системи.

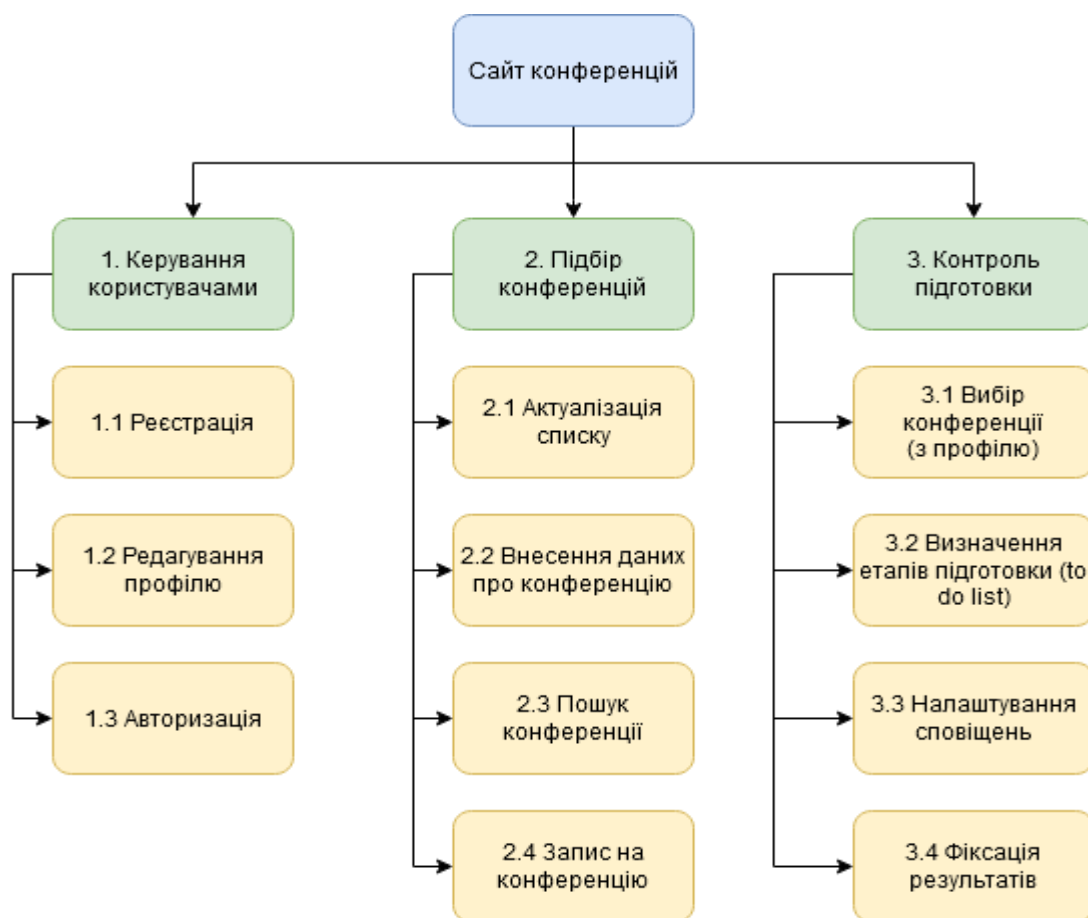


Рис. 1. FDD діаграма

Керування користувачами. Кожен користувач може створити власний акаунт, який дасть змогу налаштовувати сповіщення для конференцій, та де будуть зберігатися конференції, на які записаний користувач.

Підбір конференцій. Комплекс дій з оновлення списку конференцій, створення власної конференції для зареєстрованого користувача, пошук конференцій за критеріями (дата, місце, галузь).

Контроль підготовки. Можливість налаштовувати сповіщення для конференцій, на які записався користувач.

Діаграма потоків даних (DFD, Data Flow Diagram) схожа на діаграми FDD та SADT, оскільки її основними компонентами також є прецеденти і вона створюється за схожими принципами, такими як функціональна декомпозиція та підхід "згори донизу". Цей підхід починається з загального представлення системи і поступово переходить до більш детального опису, забезпечуючи поділ на рівні деталізації.

Основною відмінністю DFD є те, що вона зосереджується на аналізі інформаційних потоків у системі. Вона демонструє, звідки надходить інформація, де відбувається її обробка, як і де вона зберігається, а також яким чином передається між різними процесами. Ця діаграма відіграє важливу роль, оскільки допомагає зрозуміти, як система працює з даними, і дає попереднє уявлення про організацію фізичного зберігання даних.

Також необхідно відмітити, що в даній діаграмі окрім самих процесів з'явилися нові елементи, а саме:

- Зовнішні сутності. Дані елементи позначають об'єкт, який взаємодіє з системою і в процесу цих взаємодій надає системі певну інформацію, яка необхідна для забезпечення роботи процесів, а в результаті він отримує певні дані, в якості відповіді процесів;
- Сховища даних. Хоча даний об'єкт має назву «сховище даних», але фактичного він має декілька інше значення, і воно більш наближене до логічного структурування даних, аніж місце збереження. Якщо брати приклад із області кінцевої реалізації збереження даних, то даний об'єкт схожий на «Сутність» в реляційних базах даних.

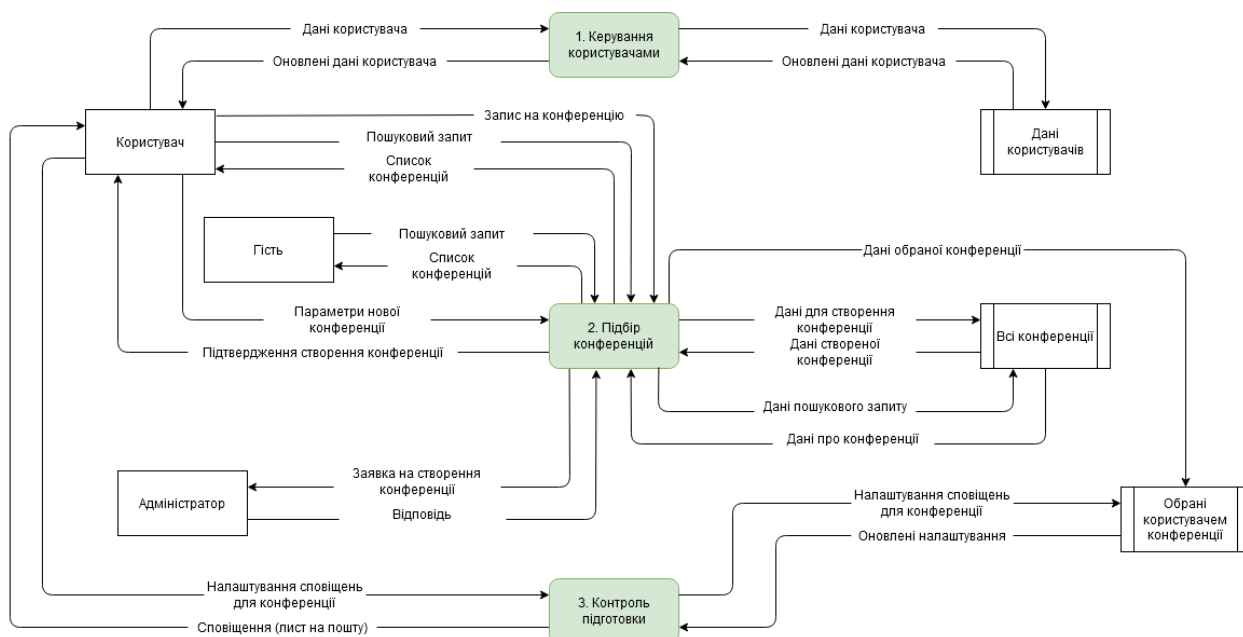


Рис. 2. DFD діаграма нульового рівня

Для даної діаграми було виділено наступні зовнішні сутності:

- Гість. Людина, що хоче потрапити на конференцію. Вводить дані для пошуку конференції, та отримує список конференцій у відповідь.
- Користувач. Зареєстрована людина. Вносить у профіль власні дані, у відповідь отримує оновлені дані профілю. Може створити конференцію, для цього відправляє параметри створюваної конференції, після підтвердження адміністратором у відповідь отримує повідомлення про створення.
- Адміністратор. Працівник, що наглядає за системою та підтверджує новостворені конференції. Отримує заявку на створення, у відповідь посилає своє рішення.

В якості сховищ було виділено:

- Дані користувачів. Даний об'єкт зберігає дані користувачів, а саме: ідентифікуючі дані, особисті дані, роль в системі.
- Всі конференції. Даний об'єкт зберігає дані конференцій (місце, тези, дати тощо).
- Обрані користувачем конференції. Даний об'єкт зберігає дані та налаштування сповіщень конференцій, на які записався користувач.

Формалізація вимог до системи

Сформуємо вимоги до системи відповідно до проведеного раніше аналізу:

Джерело	Вимоги
Попередній огляд	Було сформовано розуміння, хто є користувачами системи, сформовано вимоги до функціональних можливостей та описані деякі специфічні вимоги (наприклад, вимоги до надійності).
FDD	За допомогою підходу функціональної декомпозиції було проведено аналіз, які процеси мають бути реалізовані в системі: керування користувачами, підбір конференцій, контроль підготовки.
DFD	Було проведено аналіз, як мають проходити інформаційні потоки в системі, також було виокремлено сховища, які має реалізовувати система.

Табл. 2. Вимоги до системи.

1.5. Об'єктно-орієнтований аналіз

Діаграма прецедентів є одним із видів поведінкових діаграм UML, які часто застосовуються для аналізу роботи систем. Вони наочно показують різні ролі в системі та як ці ролі взаємодіють із її функціями.

Такі діаграми використовуються для збору вимог до роботи системи. Залежно від потреб, їх застосовують в різних ситуаціях:

- 1) Визначення функцій і взаємодії ролей. Головна мета діаграм – показати, які функції є в системі та як саме користувачі або інші ролі з ними працюють.
- 2) Зображення системи на загальному рівні. Це дуже корисно під час презентацій для керівників чи інших зацікавлених сторін. Діаграма показує ролі, які використовують систему, та її основні можливості, без деталізації внутрішніх процесів.
- 3) Визначення зовнішніх і внутрішніх факторів. У великих і складних проектах система може виступати як зовнішній елемент в іншому сценарії використання, що інколи потребує додаткового уточнення.

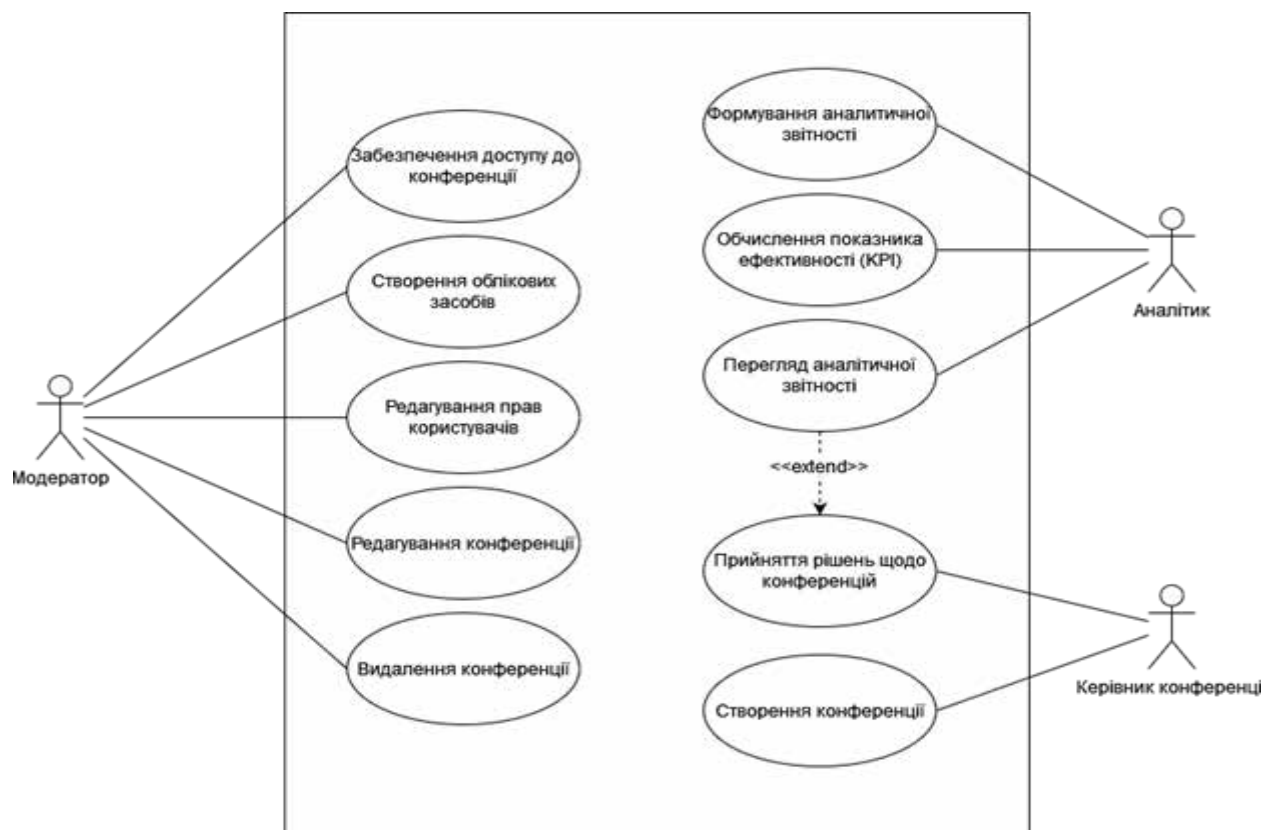


Рис. 3. Діаграма прецедентів

РОЗДІЛ 2. Проектування інформаційної системи

2.1 Об'єктно-орієнтоване проектування системи

Для виконання об'єктно-орієнтованого проектування буде використано UML (Unified Modeling Language). Ця мова є універсальним інструментом для візуального моделювання, розробленим для опису, візуалізації, проектування та документування компонентів програмного забезпечення, бізнес-процесів та інших систем.

Ефективне використання UML базується на розумінні загальних принципів моделювання складних систем та особливостей об'єктно-орієнтованого аналізу й проектування. Вибір засобів для побудови моделей визначає, які саме завдання можна вирішувати за їхньою допомогою.

При проектуванні системи з використанням UML важливо дотримуватися таких принципів:

- Принцип абстрагування: модель має включати лише ті аспекти системи, які безпосередньо стосуються її функціональності чи цільового призначення. Всі другорядні деталі варто опускати, щоб не ускладнювати аналіз і дослідження моделі.
- Принцип багатомодельності: жодна окрема модель не може повністю відобразити всі аспекти складної системи. Для цього використовуються взаємопов'язані представлення (views), кожне з яких описує певний аспект структури чи поведінки системи.

Принцип ієрархічної побудови: моделі створюються на різних рівнях деталізації або абстрагування в межах конкретних подань.

UML складається з двох взаємозалежних частин:

- Семантика UML: це мета-модель, яка визначає абстрактний синтаксис і смислове значення понять об'єктного моделювання.
- Нотація UML: графічний спосіб представлення семантики UML.

Семантика UML описується через мета-модель, що включає три аспекти: абстрактний синтаксис, правила побудови виразів і семантику. Нотація UML відображає графічні елементи на базові поняття семантики, доповнюючи її. Таким чином, ці дві частини працюють разом, забезпечуючи функціональність мови. Семантика UML викладається у напівформальному стилі, який поєднує природну мову з формальними правилами для опису базових понять і можливостей їхнього розширення.

Можемо виділити наступні сутності:

- Користувач. Описує користувачів системи.
- Адміністратор. Описує адміністратора.
- Зареєстрований користувач. Описує користувача, що зареєструвався.
- Конференції. Описує конференції. Пов'язаний з сутністю «зареєстрований користувач» зв'язком асоціація та з сутністю «сповіщення» зв'язком композиція.
- Сповіщення. Описує сповіщення, налаштовані користувачем для конференції. Пов'язаний с сутностями «зареєстрований користувач» та «конференції»
- Керуючі класи відповідають процесам з FDD, наявно 8 об'єктів-інтерфейсів. Детально розглянути діаграму класів можливо в Додатку А.

2.1.2 Діаграма послідовності

Діаграма послідовності представляє об'єкт у вигляді прямокутника, розташованого на вершині пунктирної вертикальної лінії, яка називається "лінією життя" об'єкта. Ця лінія відображає період існування об'єкта під час його участі у взаємодії.

Повідомлення між об'єктами показуються стрілками, які з'єднують їхні лінії життя, причому порядок повідомлень відповідає послідовності їхнього відправлення. На діаграмі відображаються лише ті об'єкти, які беруть безпосередню участь у взаємодії. Основна увага на таких діаграмах приділяється динамічному аспекту, тобто процесу обміну повідомленнями між об'єктами у часі.

Діаграма послідовності має два виміри. Вертикальний вимір показує час і відображає тривалість існування об'єктів у системі. Горизонтальний вимір відображає порядок розташування об'єктів, де крайній лівий об'єкт є ініціатором взаємодії, а наступні праворуч взаємодіють з попередніми.

Лінія життя об'єкта, представлена вертикальною пунктирною лінією, вказує час, протягом якого об'єкт існує у системі й може брати участь у взаємодії. Якщо об'єкт постійно присутній у системі, його лінія життя буде простягатися від верхньої до нижньої межі діаграми.

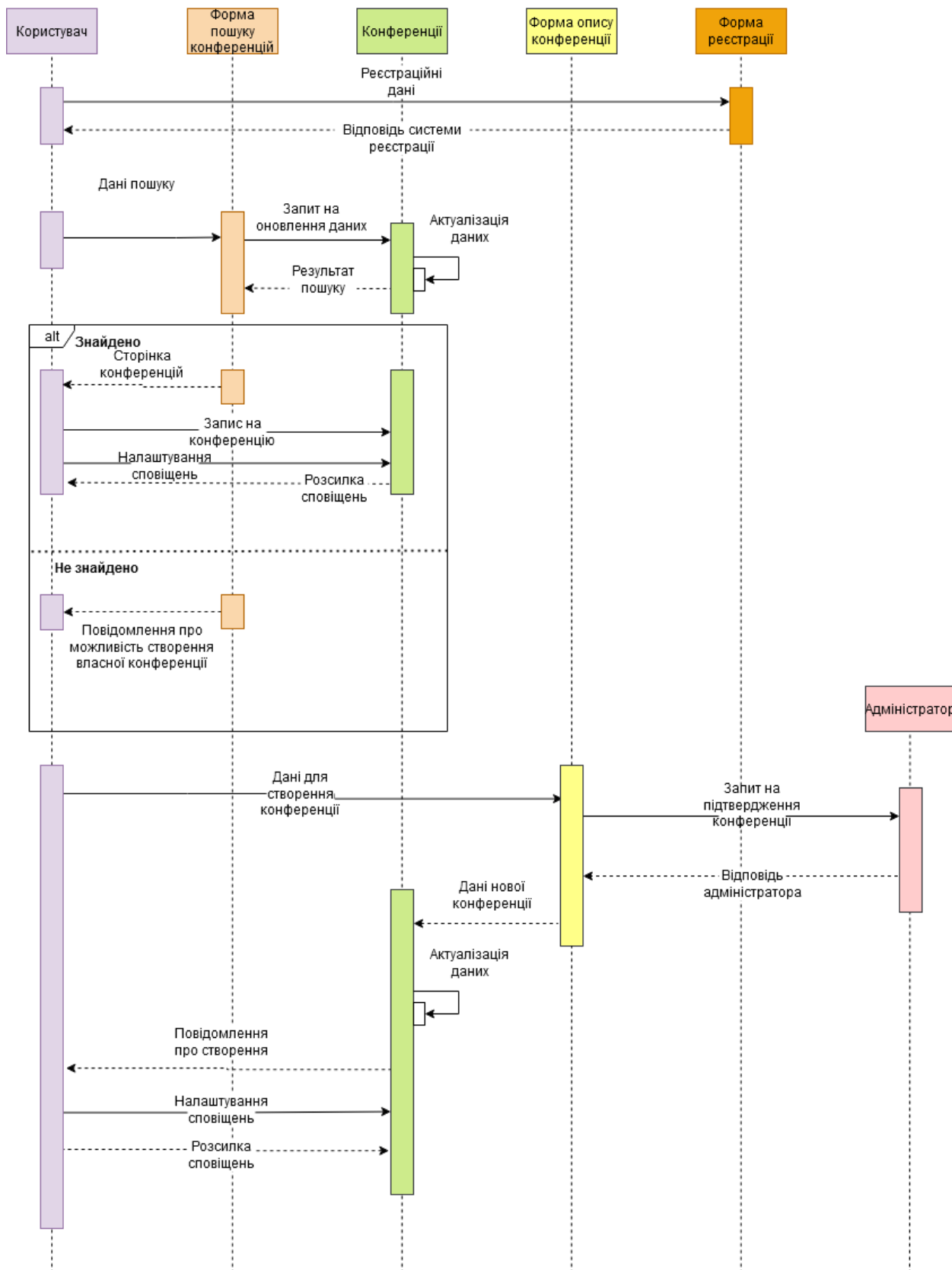


Рис. 5. Діаграма послідовностей.

База даних (БД) - це програма, яка дозволяє зберігати і обробляти інформацію у структурованому вигляді.

БД це окрема незалежна програма, яка не входить до складу мови програмування. У базі даних можна зберігати будь-яку інформацію, щоб пізніше отримувати до неї доступ.

Реляційна база даних - це набір даних з зумовленими зв'язками між ними. Ці дані організовані у вигляді набору таблиць, що складаються із стовпців і рядків. У таблицях зберігається інформація про об'єкти, представлених в базі даних. У кожному стовпчику таблиці зберігається певний тип даних, в кожному осередку - значення атрибута. Кожен рядок таблиці являє собою набір пов'язаних значень, що відносяться до одного об'єкту або сутності. Кожен рядок в таблиці може бути позначена унікальним ідентифікатором, зва- ним первинним ключем, а рядки з декількох таблиць можуть бути пов'язані з допомогою зовнішніх ключів. До цих даних можна отримати доступ багатьма способами, і при цьому реорганізувати таблиці БД не потрібно.

Побудована модель бази даних є приблизним відображенням стану таблиць в майбутній побудованій системі. Головне – відобразити інформацію про товари, їх категорію, також ролі користувачів, повідомлення, оцінки та транзакції. Можуть виникнути деякі інші таблиці у разі розширення функціоналу або задля досягнення ще більшої атомарності даних. Наприклад, ПІБ – це неатомарна одиниця. Реляційна БД вимагає приведення як мінімум до 3 Нормальної Форми.

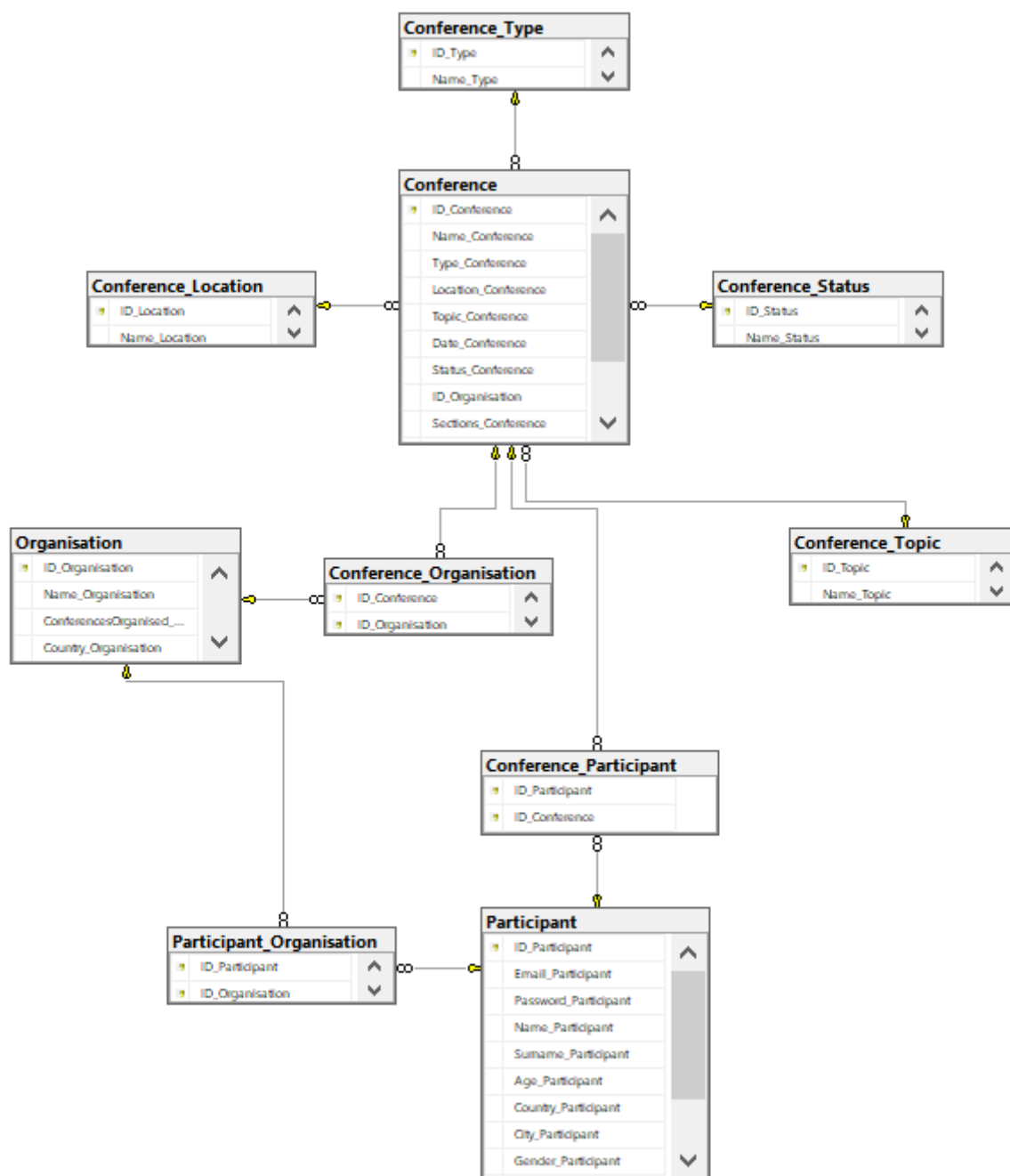


Рис. 6. Логічна модель БД

Розділ 3. Розробка системи

Розробка системи підтримки прийняття рішень для автоматизації організаційних процесів наукових конференцій була виконана з використанням сучасних веб-технологій та фреймворків, що забезпечили інтеграцію різноманітних функціональних модулів. У ході дослідження було застосовано такі методи та технології:

Системний аналіз:

- Проведено аналіз процесів організації наукових конференцій для виявлення вузьких місць і можливостей для оптимізації.
- Визначені функціональні вимоги до системи підтримки прийняття рішень на основі аналізу потреб користувачів та зацікавлених сторін.

Методи проєктування:

- **Модульний підхід** до проєктування архітектури системи був використаний для створення незалежних компонентів: ролі користувачів (адміністратор, модератор, користувач), система рекомендацій, механізми прогнозування та генерації звітів.
- Вибір архітектурного патерна **NestJS (MVC)** для чіткого розділення логіки та забезпечення масштабованості проєкту.

Технології та інструменти розробки:

- **NestJS** — фреймворк для розробки бекенд-застосунків на TypeScript, що забезпечує модульність, інкапсуляцію та використання декораторів для побудови REST API.
- **TypeORM** — ORM (Object-Relational Mapping) для роботи з базами даних, зокрема з **SQLite**, що була обрана для легкості налаштування та тестування.
- **RxJS** — бібліотека для асинхронного програмування, що забезпечує реактивне управління даними та обробку подій.

- **@nestjs-modules/mailer** — для реалізації механізму відправки нагадувань користувачам на електронну пошту.
- **JWT (JSON Web Token)** — використовується для автентифікації користувачів та управління доступом до різних частин системи.

Методи прогнозування та рекомендацій:

- **Машинне навчання:** для прогнозування кількості учасників конференцій використовувались алгоритми на основі аналізу попередніх даних про відвідуваність.
- **Рекомендаційні системи:** при створенні конференцій система пропонує теги на основі аналізу минулих заходів, а також підбирає персоналізовані рекомендації для користувачів на основі їх інтересів та вподобань.

Методи тестування та контролю якості:

- **Jest** — використовувався для модульного тестування системи та забезпечення стабільності коду.
- **Supertest** — застосовувався для тестування HTTP-запитів і перевірки коректності роботи REST API.
- **ESLint та Prettier** — для забезпечення стандартів якості коду та його форматування.

Методи збору та аналізу даних:

- Для аналізу результатів проведених конференцій використовувалась функція **генерації аналітичних звітів**
- Отримані дані використовувались для подальшого вдосконалення алгоритмів рекомендацій і прогнозування, що дозволило підвищити точність прогнозів та релевантність рекомендацій.

Застосування зазначених методів і технологій дозволило створити ефективну систему підтримки прийняття рішень для організації наукових

конференцій, що підвищує їхню результативність та автоматизує ключові процеси.

Наукова новизна: у рамках даного дослідження вперше було розроблено систему підтримки прийняття рішень для автоматизації процесів організації наукових конференцій, яка включає функціональні можливості на основі сучасних технологій та алгоритмів прогнозування. Основні елементи наукової новизни полягають у наступному:

- Реалізовано алгоритм обробки інформації для автоматичної генерації аналітичних звітів, що включають не лише кількісні показники, а й аналіз учасників за демографічними характеристиками (стать, країна тощо). Це сприяє більш глибокому розумінню аудиторії та підвищенню якості організації майбутніх заходів.
- Було розроблено інтегровану систему рекомендацій для користувачів на основі аналізу їхніх уподобань і участі у попередніх конференціях. Система автоматично підбирає актуальні конференції та пропонує релевантні теги для нових заходів.
- Впроваджено алгоритм прогнозування кількості учасників конференцій, заснований на аналізі даних про попередні заходи, що дозволило підвищити точність планування.
- Було досліджено вплив персоналізованих рекомендацій на участь користувачів у конференціях. Дослідження показало, що використання рекомендаційних алгоритмів збільшує залученість користувачів і підвищує відвідуваність заходів.

Таким чином, результати дослідження мають важливе значення для підвищення ефективності управління науковими конференціями та забезпечують нові підходи до автоматизації організаційних процесів із використанням сучасних технологій і методів аналізу даних.

ВИСНОВКИ

У ході цього дослідження було розроблено систему підтримки прийняття рішень при організації наукових конференцій з використанням сучасних технологій, таких як NestJS, TypeORM, SQLite та RxJS, що забезпечило гнучкість, масштабованість та надійність.

Програмні та апаратні вимоги:

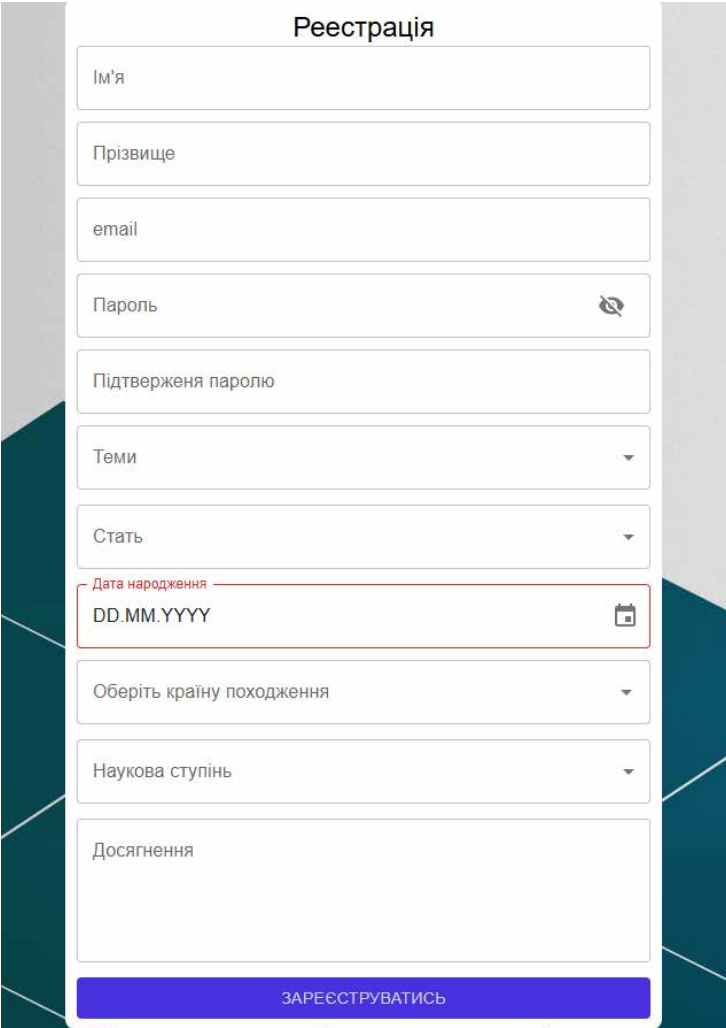
- Серверні вимоги: 4 ГБ ОЗУ, процесор із частотою 2.5 ГГц, 20 ГБ вільного місця на диску;
- Програмні залежності: Node.js (v16 та вище), NestJS, TypeORM, SQLite, JWT, Mailer;
- Середовище розробки: TypeScript, Postman (для тестування API), Docker (для контейнеризації).

Ключові результати роботи:

1. Реалізація системи ролей (користувач, модератор, адміністратор) дозволила керувати доступом до різних функцій системи, що забезпечує безпеку при користуванні системою.
2. Модуль реєстрації та запису на конференції дозволяє користувачам легко знаходити та приєднуватися до заходів, що сприяє збільшенню кількості учасників.
3. Функція модерації конференцій забезпечує перевірку перед публікацією, що підвищує якість та достовірність інформації.
4. Прогнозуючі алгоритми при створенні конференцій надають оцінку очікуваного числа підписантів та учасників на основі даних про минулі події, що допомагає організаторам при плануванні.
5. Модуль генерації звітів забезпечує аналіз проведених конференцій за такими показниками, як кількість учасників, гендерний склад, географічний розподіл тощо, що дозволяє організаторам планувати майбутні заходи на основі отриманих даних.

6. Система рекомендацій на основі вподобань та історії взаємодії користувачів дозволяє персоналізувати конференції під користувачів, що значно збільшує ймовірність участі у запропонованих заходах.
7. Надсилання електронних листів з нагадуванням про конференцію допомагає користувачам не пропускати важливі події та підтримує високий рівень залучення.

Нижче наведено деякі приклади реалізації системи:




Реєстрація


Ім'я


Прізвище


email


Пароль 


Підтверження паролю

Теми 

Стать 

Дата народження 
DD.MM.YYYY

Оберіть країну походження 

Наукова ступінь 

Досягнення

ЗАРЕЄСТРУВАТИСЬ

Рис. 7. Реєстрація акаунта

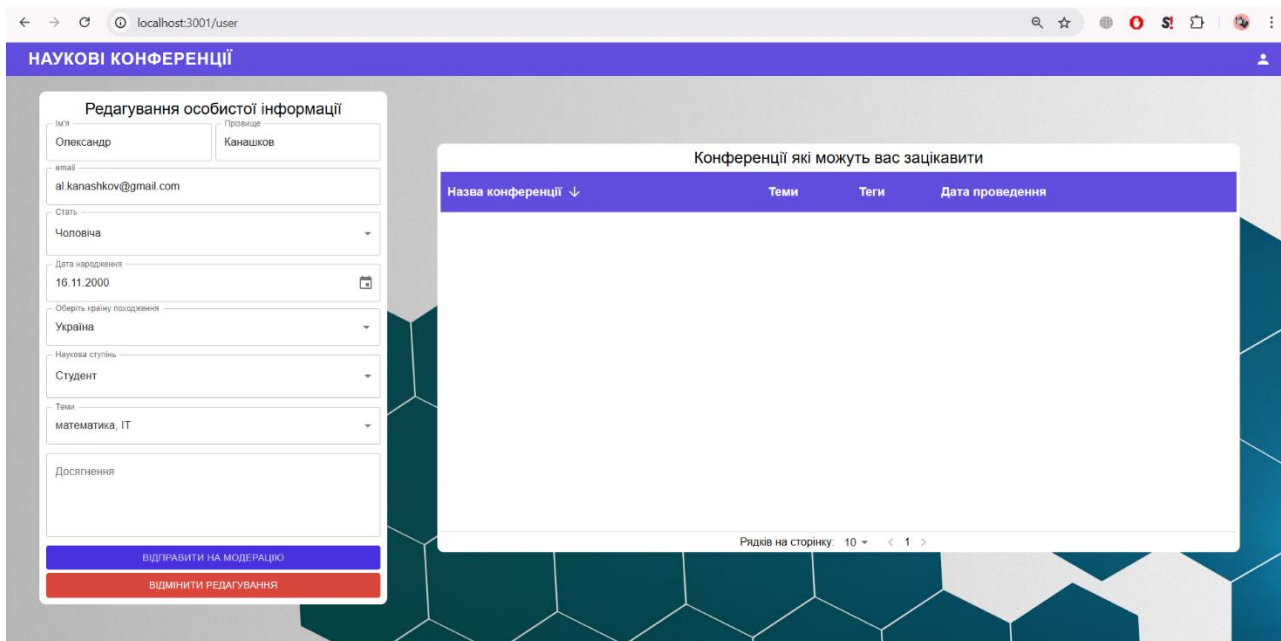


Рис. 8 . Профіль користувача

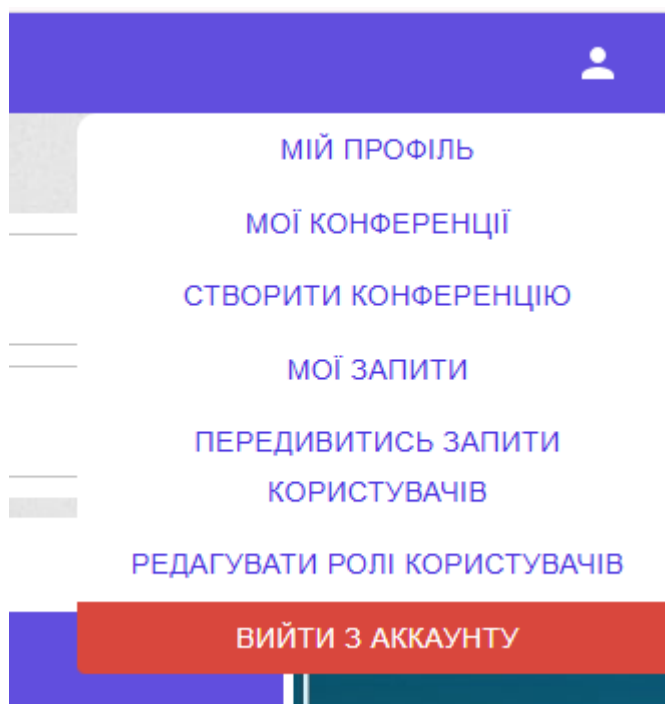


Рис. 9. Пункти меню в профілі (адміністратор)

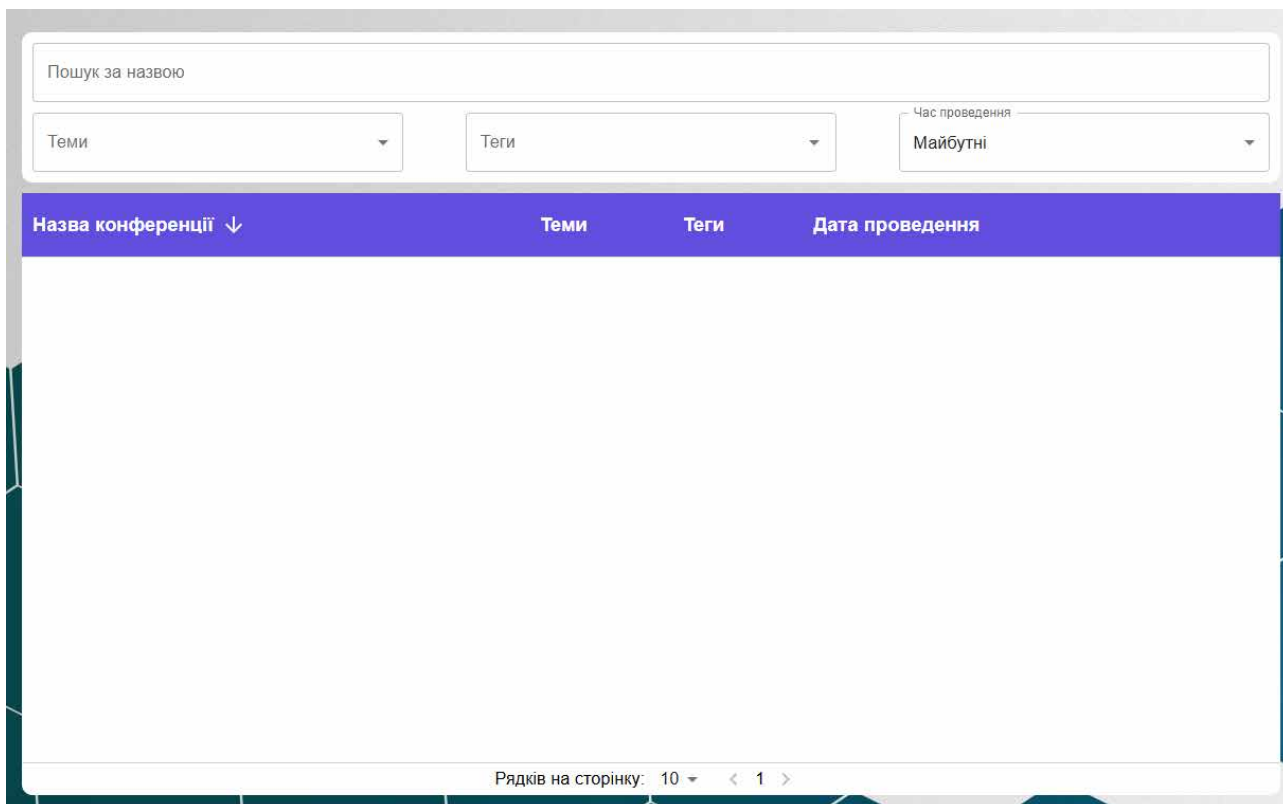


Рис. 10. Компонент з конференціями та фільтром

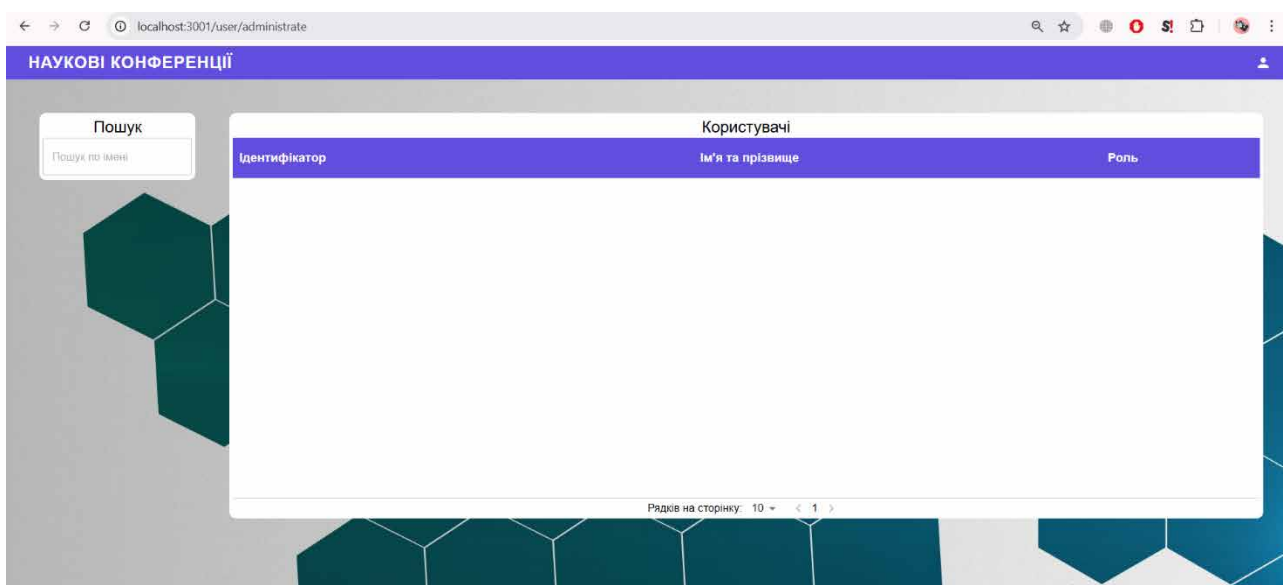


Рис. 11. Пошук користувачів

localhost:3001/request

НАУКОВІ КОНФЕРЕНЦІЇ

Запити на зміну профілю

Номер запиту ↓	Статус	Причина відмови
1	Затверджено	-
2	Затверджено	-
3	Затверджено	-

Рядків на сторінку: 10 < 1 > Вивести видалені

Запити на створення або зміну конференції

Номер запиту ↓	Статус	Тип запиту	Причина відмови
1	Затверджено	Створення	-

Рядків на сторінку: 10 < 1 > Вивести видалені

Рис. 12. Запити на зміни (адміністратор)

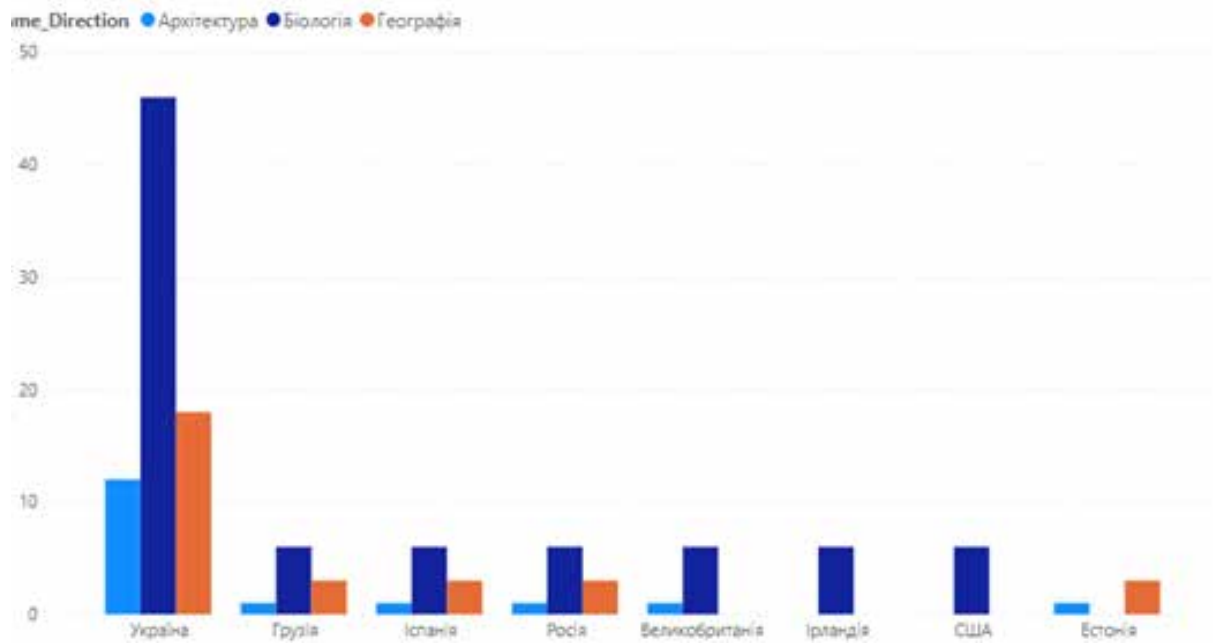


Рис. 13. Статистика за країнами в розрізі напрямку конференції

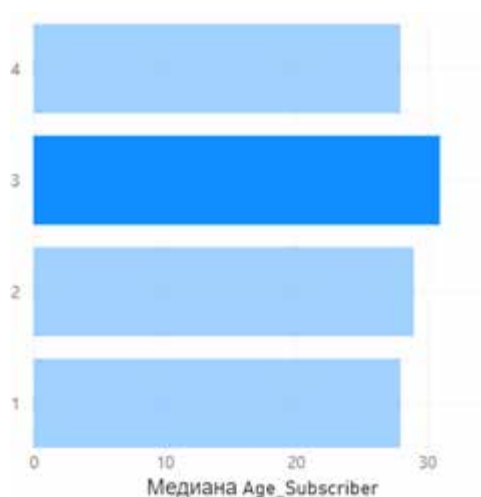


Рис. 14. Медіана вікових груп

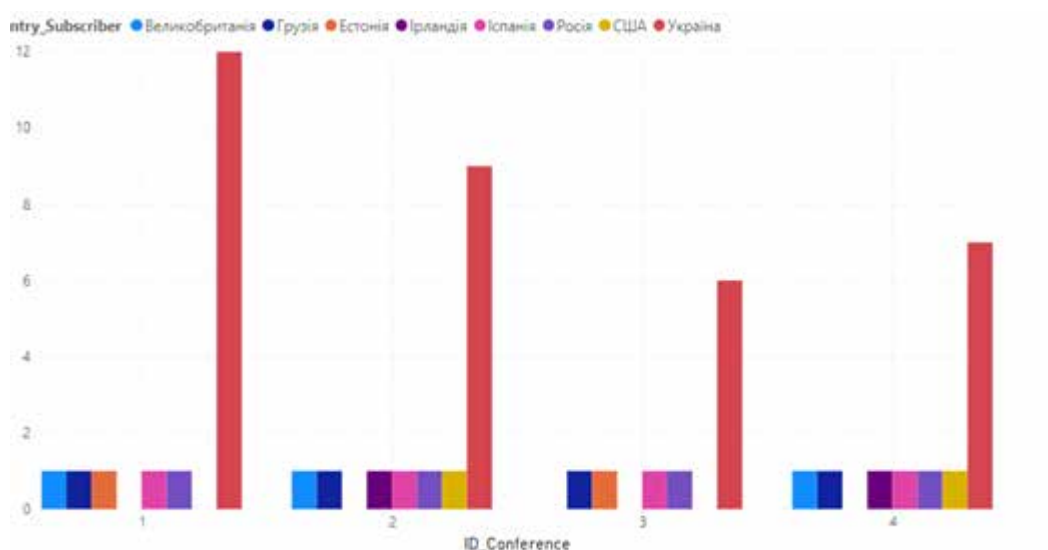


Рис. 15. Підписники за країнами

Розроблена система продемонструвала свою ефективність у автоматизації процесів при організації наукових конференцій. Використання сучасних алгоритмів прогнозування та персоналізованих рекомендацій дозволяє значно покращити взаємодію з користувачами та підвищити якість послуг. Впровадження такої системи може суттєво підвищити конкурентоспроможність організацій, які займаються проведенням наукових заходів, та підвищити їхню привабливість для користувачів.

Виходячи з отриманих результатів, можна надати наступні рекомендації щодо подальшого використання та розвитку цієї системи:

Інтеграція в університети та наукові установи:

- Система може бути використана в якості платформи для проведення конференцій, семінарів та наукових зустрічей у навчальних закладах та дослідницьких центрах. Це дозволить автоматизувати організаційні процеси та зменшити навантаження на адміністративний персонал.

Подальший розвиток алгоритмів машинного навчання:

- Для підвищення точності прогнозування відвідуваності та персоналізованих рекомендацій можна інтегрувати більш складні алгоритми машинного навчання та штучного інтелекту, зокрема моделі на основі нейронних мереж;
- Використання алгоритмів для аналізу поведінки користувачів на платформі дозволить ще більше підвищити рівень персоналізації.

Розробка мобільного застосунку:

- Для підвищення зручності користувачів варто розглянути можливість створення мобільного застосунку, який дозволить отримувати доступ до функцій платформи з будь-якого пристрою та отримувати сповіщення про майбутні події.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Навчальний посібник з дисципліни «Системний аналіз» для здобувачів спеціальності 122 – Комп'ютерні науки / В.М. Тонконогий, В.О. Вайсман, Л.В. Бовнегра, К.Г. Кіркопуло. – Одеса: Нац. ун-т «Одеська політехніка», 2022. – 84 с.
2. Суздаль Є. Г. Теорія систем і системний аналіз / *Принципи і методи системного аналізу* СПБ : СПБ Гуті, 2010. URL: https://stud.com.ua/45001/investuvannya/printsipi_metodi_sistemnogo_analizu#srcannot_1
3. Гаркуша Н.М. Моделі і методи прийняття рішень в аналізі та аудиті: підручник, 2012. URL: <https://westudents.com.ua/glavy/6674-osoblivost-sistemnogo-analzu.html>
4. URL: <https://easychair.org/> (дата звернення: 14.11.2024).
5. URL: <https://www.conftool.net/> (дата звернення: 14.11.2024).
6. URL: <https://whova.com/> (дата звернення: 14.11.2024).
7. Aytug, H., Bhattacharyya, S., Koehler, G., and Snowdon, J. (1994). A review of machine learning in scheduling. *IEEE Trans. Eng. Manag.* 41, 165–171. doi: 10.1109/17.293383
8. Підсистеми управління. URL: <https://studfile.net/preview/5602842/page:5/>
9. Литвин В.В., Шаховська Н.Б. Навчальний посібник з курсу «Проектування інформаційних систем» для студентів базового «Консолідована інформація» : Львів – 2010, 103 с. URL: <https://studfile.net/preview/7367578/page:11/>
10. О. Г. Харченко, І. Е. Райчев, В. В. Замковий МЕТОДОЛОГІЯ ПРОЕКТУВАННЯ ІНФОРМАЦІЙНИХ СИСТЕМ: навчальний посібник. Одеса: Нац. ав. ун-т, 2010. - 19 с.
11. Neubert, S., & Casel, K. (2024). Incremental Ordering for Scheduling Problems. *Proceedings of the International Conference on Automated Planning and Scheduling*, 34(1), 405-413. <https://doi.org/10.1609/icaps.v34i1.31500>
12. Zhang, Y., Deekshith, U., Wang, J., & Boedecker, J. (2024). Improving the Efficiency and Efficacy of Multi-Agent Reinforcement Learning on Complex Railway Networks with a Local-Critic Approach. *Proceedings of the International Conference on Automated Planning and Scheduling*, 34(1), 698-706. <https://doi.org/10.1609/icaps.v34i1.31533>