

CIT₂₀₂₅

WORKSHOP

ЗИМОВА ШКОЛА-СЕМІНАР

молодих вчених і студентів

2 грудня
2025 року

КОМП'ЮТЕРНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ



м. Тернопіль
вул. О. Теліги, 8

fcit.wunu.edu.ua

ОРГАНІЗАТОРИ

- Західноукраїнський національний університет
- Факультет комп'ютерних інформаційних технологій
- Асоціація фахівців комп'ютерних інформаційних технологій

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Асоціація фахівців комп'ютерних інформаційних технологій

МАТЕРІАЛИ ЗИМОВОЇ ШКОЛИ-СЕМІНАРУ
МОЛОДИХ ВЧЕНИХ І СТУДЕНТІВ

КОМП'ЮТЕРНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

COMPUTER INFORMATION TECHNOLOGIES

2 грудня 2025 року

CIT'2025

Тернопіль
ЗУНУ
2025

ББК 32.97

УДК 004.2-3+004.9+51.7+519.6-8

Організатори школи-семінару:

Західноукраїнський національний університет

Факультет комп'ютерних інформаційних технологій

Асоціація фахівців комп'ютерних інформаційних технологій

32.97 *Комп'ютерні інформаційні технології: Матеріали зимової школи-семінару молодих вчених і студентів СІТ'2025. – Тернопіль: ЗУНУ, 2025.*

У матеріалах семінару опубліковані результати наукових досліджень і розробок науковців та студентів факультету комп'ютерних інформаційних технологій ЗУНУ з таких напрямків: математичні моделі об'єктів та процесів, комп'ютерні мережеві технології; спеціалізовані комп'ютерні системи; системи штучного інтелекту; інженерія програмного забезпечення; комп'ютерні технології інформаційної безпеки та управління проектами. Матеріали призначені для наукових співробітників, викладачів, інженерно-технічних працівників, аспірантів та студентів.

Відповідальний за випуск:

Лукас А.В., д. т. н., професор, завідувач кафедри комп'ютерних наук

Відповідальність за достовірність, стиль викладення та зміст надрукованих матеріалів несуть автори.

©ЗУНУ, 2025

© колектив авторів, 2025

ГОЛОВНИЙ РЕДАКТОР:

КРЕПИЧ Світлана Ярославівна *к.т.н., доцент*

ЗАСТУПНИК ГОЛОВНОГО РЕДАКТОРА:

СПИВАК Ірина Ярославівна *к.т.н., доцент*

ЧЛЕНИ РЕДАКЦІЙНОЇ КОЛЕГІЇ

ВОЙТЮК Ірина Федорівна *к.т.н., доцент*

ГОНЧАР Людмила Іванівна *к.е.н., доцент*

КРЕПИЧ Світлана Ярославівна *к.т.н., доцент*

МАНЖУЛА Володимир Іванович *д.т.н., професор*

МАРЦЕНЮК Євгенія Олексіївна *к.т.н., доцент*

ПОРПЛИЦЯ Наталія Петрівна *к.т.н., доцент*

ПУКАС Андрій Васильович *д.т.н., професор*

СПИВАК Ірина Ярославівна *к.т.н., доцент*

СТАСІВ Ірина Степанівна *к.т.н., доцент*

ТИМЧИШИН Володимир Степанович *д.філ., ст.викладач*

ШПНТАЛЬ Михайло Ярославович *к.т.н., доцент*

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

ПУКАС Андрій Васильович *д.т.н., професор*

КРЕПИЧ Світлана Ярославівна *к.т.н., доцент*

СПИВАК Ірина Ярославівна *к.т.н., доцент*

ЗМІСТ

СИСТЕМА АНАЛІЗУ ПРОДУКТИВНОСТІ КОМАНД РОЗРОБНИКІВ НА ОСНОВІ МЕТРИК З GIT.....	1
Гончар Л.І., Трачук О.О.	
ПІДХІД ДО СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДОГЛЯДУ ЗА РОСЛИНАМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ.....	3
Сворінь Д.В., Співак І.Я., Крепич С.Я.	
МАТЕМАТИЧНА МОДЕЛЬ ПРОГНОЗУВАННЯ РИЗИКІВ ПРОГРАМНИХ ПРОЄКТІВ НА ОСНОВІ АНАЛІЗУ КРІ.....	5
Гончар Л.І., Гук Ю.Р.	
ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ СОЦІАЛЬНОЇ МЕРЕЖІ З АДАПТИВНИМ МЕХАНІЗМОМ ОБМЕЖЕННЯ СОЦІАЛЬНИХ ЗВ'ЯЗКІВ.....	7
Співак І.Я., Устиченко О.О., Крепич С.Я.	
ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ, ЩО ПОКРАЩУЮТЬ ПРИВАТНІСТЬ ДЛЯ АНОМАЛІЗАЦІЇ ТЕКСТОВИХ ДАНИХ.....	9
Заєць Т.В.	
ВЕБ-ДОДАТОК АУДИОПІДСУМКІВ МАТЕРІАЛІВ НА ОСНОВІ ШІ ДЛЯ КОРИСТУВАЧІВ З ПОРУШЕННЯМ ЗОРУ.....	11
Крепич С.Я., Дячок С.І., Співак І.Я., Крепич Р.В.	
ПРОГРАМНІ ЗАСОБИ ДЛЯ ОЦІНЮВАННЯ ВЕБ-ДОСТУПНОСТІ ЧЕРЕЗ СИМУЛЯЦІЮ ВАД ЗОРУ.....	13
Стасів І.С., Порплиця Н.П., Чир Д.М.	
АРХІТЕКТУРА ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ ПЗ НАУКОВОГО ПРИЗНАЧЕННЯ НА ОСНОВІ ПАТЕРНІВ ADAPTER TAVISITOR.....	15
Сеньків Ю.М., Манжула В.І.	
МЕТОДОЛОГІЧНІ ПІДХОДИ ДО БЕЗПЕЧНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	19
Гераскін М.А.	
ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДИНАМІЧНОГО ЦІНОУТВОРЕННЯ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ.....	21
Тимчишин В.С., Плавуцький С.М.	
СИСТЕМА ОПТИМІЗАЦІЇ ХМАРНИХ ВИТРАТ НА ОСНОВІ АНАЛІЗУ МЕТРИК СПОЖИВАННЯ.....	23
Марценюк Є.О., Бойко А.А.	
АРХІТЕКТУРА ВЕБ-СИСТЕМИ УПРАВЛІННЯ СТУДЕНТСЬКИМИ НАУКОВИМИ ПРОЄКТАМИ.....	25
Матвійчук Ю.В., Тимчишин В.С.	
ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ РОЗПОДІЛУ РЕСУРСІВ З УРАХУВАННЯМ ПРОФЕСІЙНОГО ВИГОРАННЯ.....	27
Осадчук С.Б., Шпінталь М.Я.	
МАТЕМАТИЧНА МОДЕЛЬ ОПТИМАЛЬНОГО РОЗМІЩЕННЯ ТОЧОК ЗВ'ЯЗКУ У ПРИМІЩЕННЯХ.....	29
Марценюк Є.О., Корман М.К.	
ПРОГРАМНА СИСТЕМА ДЛЯ ОРГАНІЗАЦІЇ КОМАНДНОЇ РОБОТИ.....	31
Крепич С.Я., Гида І.Р., Співак І.Я.	
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ ТЕХНІЧНОГО БОРГУ ПРИ МІГРАЦІЇ LEGACY JAVA-ДОДАТКІВ.....	33
Войтюк І.Ф., Косаренко П.К.	
ІНТЕЛЕКТУАЛЬНА СИСТЕМА SPARK.....	35
Крепич С.Я., Маркевич Д.В., Співак І.Я., Крепич Р.В.	
АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ СХОДІВ У ЖИТЛОВИХ ПРИМІЩЕННЯХ.....	38
Войтюк І.Ф., Кузьмяк Б.В.	
ВЕБ-ПЛАТФОРМА ДЛЯ СОЦІАЛЬНОЇ ІНТЕГРАЦІЇ ВНУТРІШНЬО ПЕРЕМІЩЕНИХ ОСІБ В УКРАЇНІ.....	41
Кліщ С.С., Крепич С.Я., Співак І.Я.	
ПРОЄКТУВАННЯ СТРУКТУРИ БАЗИ ДАНИХ СИСТЕМИ ОНЛАЙН-ЗАПИСУ ДО ЛІКАРЯ.....	43
Могильська І.М., Тимчишин В.С.	
ВЕБ-ПЛАТФОРМА ДЛЯ ОРГАНІЗАЦІЇ ТА АГРЕГАЦІЇ МІСЬКИХ ПОДІЙ.....	45
Крепич С.Я., Молдавчук А.В., Співак І.Я.	
ВЕБ-СИСТЕМА ДЛЯ ОРГАНІЗАЦІЇ ДІЯЛЬНОСТІ ТАНЦЮВАЛЬНОГО КОЛЕКТИВУ.....	47
Мелько І.М., Тимчишин В.С.	

ВЕБ-ЗАСТОСУНОК ДЛЯ ПІДТРИМКИ ПРОЦЕСІВ ОРЕНДИ ЖИТЛА.....	49
Балух В.І., Крепич С.Я., Співак І.Я.	
ІНТЕРАКТИВНИЙ СИМУЛЯТОР ЕЛЕКТРОННИХ СХЕМ ДЛЯ НАВЧАЛЬНИХ ЗАВДАНЬ.....	51
Фанта В.В.	
ПРОЄКТУВАННЯ БАЗИ ДАНИХ СОЦІАЛЬНОЇ ПЛАТФОРМИ З АІ-МОДЕРАЦІЄЮ ДЛЯ ВИЯВЛЕННЯ ТА ФІЛЬТРАЦІЇ ТОКСИЧНИХ ПОВІДОМЛЕНЬ У СОЦІАЛЬНИХ МЕРЕЖАХ.....	53
Тимчишин В.С., Сіماشко І.В.	
ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ДЛЯ УПРАВЛІННЯ ОСОБИСТИМИ ФІНАНСАМИ.....	55
Палійчук Х.М., Тимчишин В.С.	
АРХІТЕКТУРА ПРОГРАМНОЇ СИСТЕМИ ПОЄДНАННЯ АРТ-ТЕРАПІЇ ТА ФІЗИЧНОЇ РЕАБІЛІТАЦІЇ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ.....	57
Цапів Я.А.	

СИСТЕМА АНАЛІЗУ ПРОДУКТИВНОСТІ КОМАНД РОЗРОБНИКІВ НА ОСНОВІ МЕТРИК З GIT

Гончар Л.І.¹⁾, Трачук О.О.²⁾
Західноукраїнський національний університет
^{1)к.е.н., доцент; 2) магістрант}

I. Постановка проблеми

У сучасній індустрії розробки програмного забезпечення зростає потреба у точному, об'єктивному та автоматизованому оцінюванні продуктивності команд. Якщо попереднє дослідження було спрямоване на створення системи збору та аналізу базових Git-метрик, то подальший розвиток роботи вимагає переходу до глибшої інтерпретації отриманих даних, поєднання їх з інтелектуальними методами аналізу та формування більш комплексних моделей оцінювання. Стрімке зростання масштабів проєктів, поява розподілених команд, зростання кількості паралельних гілок розробки та ускладнення процесів релізного циклу створюють додаткові виклики для менеджерів і технічних лідерів.

Саме тому виникає потреба у створенні системи, здатної не лише збирати метрики, але й автоматично виявляти тенденції, аналізувати аномалії, прогнозувати поведінку команди та забезпечувати прозору, детальну і достовірну аналітику для прийняття управлінських рішень.

II. Мета роботи

Мета публікації – представити розширений підхід до аналізу продуктивності команд розробників, що використовує зібрані Git-метрики, алгоритми штучного інтелекту та моделі прогнозування для отримання більш глибоких висновків щодо стану, динаміки та потенційних ризиків розвитку команди

III. Основна частина

Розширення системи аналізу продуктивності команд розробників ґрунтується на інтеграції інтелектуальних методів обробки даних із механізмами збору Git-метрик, що були реалізовані на попередньому етапі дослідження. Основна увага приділяється автоматизації, масштабованості та здатності системи обробляти великі обсяги інформації у реальному часі, забезпечуючи при цьому максимально точні та корисні аналітичні висновки для менеджерів і технічних лідерів.

Для розширеного збору даних система використовує REST API платформ GitHub, GitLab і Bitbucket, які дозволяють отримувати більше проміжних і допоміжних метрик, ніж у базовій конфігурації. Зокрема, крім традиційних показників (кількість комітів або частота зміни файлів), система фіксує часові інтервали між створенням задачі і першим комітом розробника, середню тривалість review-процесів, затримки під час об'єднання гілок, інтенсивність “гарячих” змін перед релізами, а також відстежує повторювані модифікації одних і тих самих файлів, що може вказувати на накопичення технічного боргу або проблемні ділянки системи. На цьому етапі система формує розширені профілі активності кожного розробника та проєкту в цілому, що створює підґрунтя для глибшого аналізу.

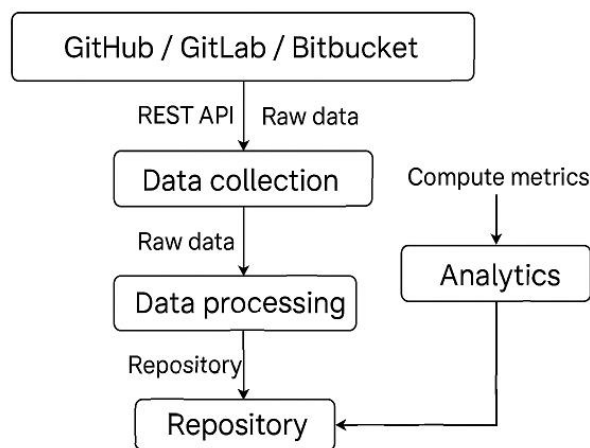


Рисунок 1 – Архітектурна схема системи аналізу продуктивності команд розробників

Сирі дані формуються у форматі JSON, після чого проходять попередню обробку та трансформацію. Система автоматично нормалізує часові позначки, конвертує часові пояси, усуває дублікати подій і виконує узгодження даних між різними джерелами. Це дозволяє звести метрики з GitHub і GitLab до єдиного формату та забезпечити коректне агрегування на рівні проекту або окремої команди. Після цього інформація зберігається у хмарній базі даних Firebase або Firestore, що забезпечує високу швидкість доступу, автоматичну синхронізацію та можливість паралельного читання великими обсягами.

Основний модуль аналітики включає обчислення інтегральних показників продуктивності, які враховують кілька типів даних одночасно. Система виявляє тренди у коміт-активності, визначає часові піки, будує діаграми навантаження на команду та окремих учасників. За допомогою статистичних алгоритмів оцінюються середні та медіанні значення часу обробки pull request, інтенсивність змін перед релізами, закономірності повторних змін у тих самих компонентах коду. Особлива увага приділяється аналізу аномалій: система автоматично позначає атипові сплески комітів або необґрунтовано великі зміни коду, що можуть свідчити про стресові ситуації, погано сплановані задачі або загрозу технічного боргу.

На більш просунутому рівні система інтегрує алгоритми машинного навчання для виявлення закономірностей. Вона може кластеризувати розробників за стилем роботи, виявляти поведінкові патерни команд, прогнозувати рівень активності на наступні дні або тижні залежно від історії попередніх розробок. Моделі машинного навчання аналізують взаємозв'язки між кількістю комітів, стабільністю коду, швидкістю review-процесів та ефективністю спринтів. Це дозволяє виявляти «вузькі місця» в команді та знаходити оптимальні точки для її покращення.

Для візуалізації результатів аналітики система пропонує інтерактивний веб-інтерфейс, побудований на основі сучасних фреймворків React або Vue.js. У ньому передбачено: графіки активності комітів за останні 7, 14 або 30 днів; діаграми залежності навантаження від часу доби; структури взаємодії між учасниками команди на основі pull request; таблиці з рейтинговими показниками розробників; інтегровані теплові карти продуктивності. Користувачі отримують можливість експорту даних у форматах PDF, CSV та JSON для подальшого використання.

У разі потреби система формує дашборди, які об'єднують ключові метрики в інтуїтивно зрозумілі панелі. Менеджер команди може миттєво переглянути зменшення активності, збільшення часу review або підвищення кількості змін у проблемних файлах. Додатково реалізовано модуль автоматичних сповіщень, який надсилає попередження у разі виявлення критичних аномалій чи ризиків: різке падіння активності, тривалі блокування merge-процесів або затримки у релізних циклах.

Система проходить комплексне тестування на реальних відкритих репозиторіях, що дозволяє перевірити надійність алгоритмів обробки та точність метрик. Таке тестування слугує основою для подальшої оптимізації продуктивності, удосконалення інструментів візуалізації та підвищення точності прогнозних моделей. Завдяки цьому система може бути впроваджена як внутрішній інструмент у команді або як частина корпоративної інфраструктури DevOps.

Висновок

Розширена система аналізу продуктивності команд розробників демонструє ефективність поєднання Git-метрик, автоматизованого збору даних, інтелектуальної обробки та інтерактивної візуалізації, що забезпечує глибше розуміння командної динаміки та виявлення ключових тенденцій у роботі. Завдяки інтеграції алгоритмів машинного навчання, виявленню аномалій та побудові прогнозів система дозволяє своєчасно виявляти потенційні ризики, оптимізувати робочі процеси та підвищувати ефективність команди. Отримані результати підтверджують можливість практичного впровадження розробленого підходу у реальних проектах і створюють підґрунтя для подальших досліджень у сфері DevOps-аналітики.

Список використаних джерел

1. Bird C., Nagappan N. *Empirical Software Engineering and Git Analytics*. IEEE Press, 2020.
2. Spinellis D. *Code Reading: The Open Source Perspective*. Addison-Wesley, 2016.
3. Vasilescu B., Filkov V., Devanbu P. *Developer Productivity and Coordination in Software Teams*. ACM Transactions on Software Engineering, 2019.
4. Forsgren N., Humble J., Kim G. *Accelerate: The Science of Lean Software and DevOps*. IT Revolution Press, 2018.
5. Google Engineering Research. *Developer Productivity, Measurement and Insights*. 2022.
6. GitHub Engineering Blog. *Machine Learning for Code Review and Collaboration*. 2021.
7. GitLab Documentation. *REST API Reference*. 2023.

ПІДХІД ДО СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ ДОГЛЯДУ ЗА РОСЛИНАМИ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ РОЗПІЗНАВАННЯ ЗОБРАЖЕНЬ

Сворінь Д.В.¹⁾, Співак І.Я.²⁾, Крепич С.Я.³⁾

Західноукраїнський національний університет

¹⁾магістр; ²⁻³⁾к.т.н., доцент

I. Постановка проблеми

Догляд за кімнатними рослинами потребує систематичності, уважності та розуміння особливостей кожного виду. Проте більшість користувачів, особливо початківців, не мають достатніх знань про оптимальні умови утримання рослин, їх потреби у волозі, світлі чи ґрунті. Це часто призводить до помилок у догляді — надмірного або недостатнього поливу, неправильного освітлення чи запізнілої пересадки, що негативно впливає на стан рослин.

Існуючі мобільні застосунки для догляду за рослинами частково вирішують цю проблему, проте більшість сучасних систем розпізнавання рослин працюють лише за умови постійного доступу до мережі, оскільки використовують хмарні моделі чи онлайн-бази даних. Це ускладнює використання застосунку у місцях із нестабільним сигналом або за відсутності інтернету[1]. Значна частина таких застосунків пропонує лише базові можливості безкоштовно, тоді як розпізнавання рослин або рекомендації щодо догляду потребують оформлення підписки. В таких застосунках зазвичай користувач може лише переглядати інформацію про рослину, не маючи можливості створювати власну базу з фото, описами та сповіщеннями про догляд.

II. Мета роботи

Метою роботи є визначення підходу до створення застосунку з розпізнавання рослин по фото, їх ідентифікація та надання рекомендацій на основі отриманої інформації.

III. Основна частина

Розробка мобільного застосунку для розпізнавання рослин за фотографією та отримання рекомендацій щодо догляду має на меті вирішення низки практичних і наукових завдань. Основним завданням системи є полегшення процесу догляду за кімнатними рослинами шляхом автоматизації процесів ідентифікації виду рослини та формування індивідуальних порад щодо її утримання. Такий застосунок дозволить користувачам швидко дізнатися назву рослини, частоту поливу, рівень освітлення, температуру, тип ґрунту чи інші параметри, необхідні для підтримання її здоров'я.

Мобільна програма зможе не лише розпізнавати рослину за фото, але й формувати рекомендації, які враховують біологічні особливості виду. Крім цього, система передбачатиме можливість часткової роботи в офлайн-режимі, що є важливою перевагою порівняно з більшістю аналогічних рішень. Буде реалізовано функцію додавання власних рослин вручну — користувач зможе вносити назву, опис, дати поливу та іншу інформацію без необхідності звернення до мережі.

Розроблюваний застосунок спрямований на підвищення ефективності догляду за кімнатними рослинами, особливо для користувачів без спеціальних знань. Основою його роботи є інтеграція з віддаленим сервером, що здійснює ідентифікацію рослин на основі завантаженого зображення[2]. Для цього використовується спеціалізоване API, наприклад PlantNet, яке базується на глибоких нейронних мережах, навчених на великому наборі ботанічних даних. Отриманий результат — точна наукова назва рослини — використовується для подальшого пошуку інформації про догляд та рекомендації з утримання[3].

З наукової точки зору, розроблений підхід демонструє практичне застосування технологій комп'ютерного зору та хмарних обчислень [4] у межах мобільного середовища. Використання серверної моделі класифікації є прикладом ефективного розподілу обчислювального навантаження між клієнтом і сервером, що забезпечує баланс між продуктивністю, точністю та зручністю використання. Такий підхід дозволяє створювати гібридні системи, у яких складні операції з обробки зображень виконуються на потужних віддалених машинах, а мобільний клієнт відповідає за відображення результатів, кешування даних і взаємодію з користувачем[5].

Процес розробки передбачає кілька основних етапів. На початковому етапі здійснюється проектування архітектури системи, у межах якої виділено такі ключові компоненти:

- модуль мобільного застосунку, який забезпечує інтерфейс користувача, управління локальною базою даних і формування запитів до сервера;
- модуль ідентифікації рослини по фото;
- модуль бази даних рослин, у якій зберігається структурована інформація про види рослин та їхні характеристики.

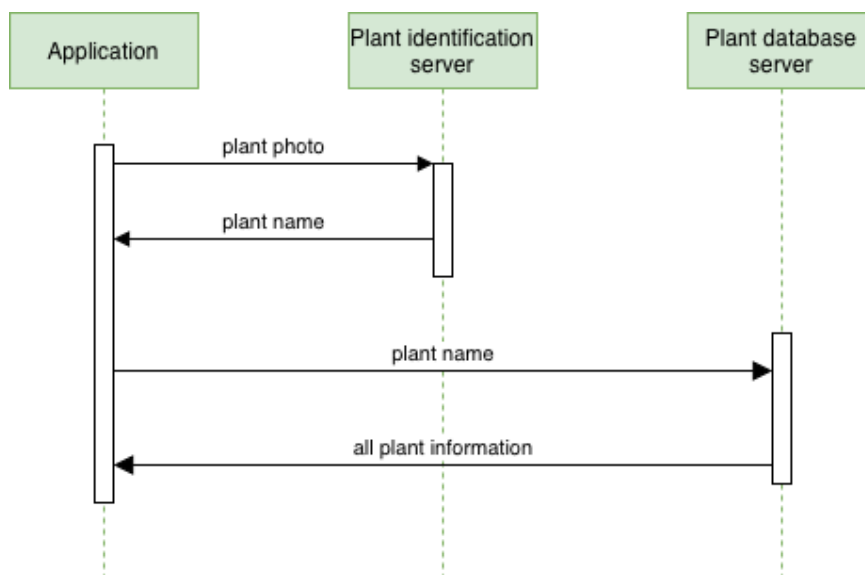


Рисунок 1 – Загальна схема послідовності системи

Взаємодія зазначених компонентів подана у вигляді блок-схеми на рисунку 1. Алгоритм роботи системи передбачає такі етапи: користувач здійснює фотографування рослини, система надсилає фото на сервер, де відбувається розпізнавання та результат розпізнавання надсилається в застосунок; після цього виконується запит до бази даних з відповідними даними і отримується вся доступна інформація про рослину включно з рекомендаціями, які відображаються в інтерфейсі користувача. У разі відсутності доступу до Інтернету користувач не зможе працювати з функціоналом автоматичного розпізнавання рослини та надання інформації та рекомендацій, проте буде мати змогу переглядати раніше збережені результати, додавати власні записи та отримувати нагадування про полив. Така поведінка вирішує проблему коли при відсутності доступу до Інтернету застосунок не працює взагалі.

Таким чином, запропонована система об'єднує наукові підходи та практичну цінність, забезпечуючи користувачеві сучасний інструмент для ефективного догляду за рослинами. Такий застосунок дозволить людям використовувати його навіть без попередніх знань про рослини, адже вся необхідна інформація буде надаватися користувачу. Це, у свою чергу, покращить користувацький досвід та сприятиме зростанню кількості активних користувачів застосунку.

Висновок

У роботі було здійснено опис функціоналу мобільного застосунку для розпізнавання рослин та надання рекомендацій щодо догляду. Було описано та схематично показано компоненти програми та обґрунтовано ефективність впровадження методів машинного навчання та комп'ютерного зору у мобільному застосунку. Такий мобільний застосунок спростить догляд за рослинами, особливо для користувачів без спеціальних знань. Реалізація запропонованої системи є доцільним кроком у напрямі інтеграції розпізнавання зображень у повсякденні мобільні застосунки, демонструючи, як технології машинного зору можуть робити побутові завдання більш зручними та ефективними.

Список використаних джерел

1. Best Plant Care Apps in 2025. [Електронний ресурс] – Режим доступу: <https://myplantin.com/blog/best-plant-care-apps>
2. Spivak I., Krepych S., Fedorov O., Spivak S. Approach to recognizing of visualized human emotions for marketing decision making systems. CEUR Workshop Proceedings, 2021, 2870, pp. 1292–1301
3. PlantNet API Documentation. [Електронний ресурс] – Режим доступу: <https://my.plantnet.org>
4. Співак І.Я., Крепич С.Я., Горішний В.І. Організація CLOUD-архітектури для систем забезпечення функціональної придатності статичних систем. Науковий журнал «Сучасні інформаційні системи», Харків, Том 3, №2, 2019. – с.35-39
5. Cloud-based or On-device: An Empirical Study of Mobile Deep Inference. [Електронний ресурс] – Режим доступу: <https://arxiv.org/abs/1707.04610>

МАТЕМАТИЧНА МОДЕЛЬ ПРОГНОЗУВАННЯ РИЗИКІВ ПРОГРАМНИХ ПРОЄКТІВ НА ОСНОВІ АНАЛІЗУ КРІ

Гончар Л.І.¹⁾, Гук Ю.Р.²⁾
Західноукраїнський національний університет
¹⁾ к.е.н., доцент; ²⁾ магістрант

I. Постановка проблеми

Сучасна індустрія розробки програмного забезпечення характеризується високим рівнем невизначеності та динамічністю змін вимог. Згідно з даними міжнародних досліджень, значна частка ІТ-проектів завершується із перевищенням бюджету або порушенням дедлайнів, а близько 30% проектів не досягають фінальної стадії реалізації взагалі [1]. Ключовим фактором, що впливає на успішність проекту, є ефективне управління ризиками (Project Risk Management). Відсутність зручних інструментів для кількісної оцінки ймовірності виникнення ризиків у реальному часі призводить до реактивного, а не проактивного стилю управління. Актуальність дослідження зумовлена необхідністю створення легковагої аналітичної системи, здатної інтегруватися в існуючі процеси розробки (SDLC) та надавати об'єктивну оцінку ризиків на основі вимірюваних показників ефективності команди (KPI), таких як швидкість закриття задач, кількість дефектів та стабільність вимог.

II. Мета роботи

Метою роботи є розробка математичної моделі для підвищення ефективності управління ІТ-проектами шляхом розробки методу та програмного засобу для автоматизованого виявлення та прогнозування ризиків на основі аналізу метрик проектної діяльності.

III. Математична модель оцінювання ризиків

Для формалізації процесу виявлення ризиків пропонується використовувати модель зваженої суми показників, що базується на аналізі відхилень ключових метрик проекту (KPI). Відповідно до стандартів управління проектами, ризик визначається як функція від ймовірності події та її впливу на результати [1,2].

Інтегральний показник ризику R_{total} розраховується за формулою (1), яка дозволяє агрегувати різномірні метрики (часові, бюджетні, якісні) в єдиний індикатор стану проекту.

$$R_{total} = \sum_{i=1}^n (K_i \cdot w_i \cdot N_i) \quad (1)$$

де n – загальна кількість метрик, що відстежуються системою, K_i – значення i -го нормалізованого KPI (наприклад, відсоток прострочених задач, де 0 – норма, 1 – критичний стан), w_i – ваговий коефіцієнт важливості i -го показника для конкретного типу проекту (визначається експертним шляхом, $\sum w_i = 1$), N_i – коефіцієнт нелінійності, що дозволяє врахувати різке зростання ризику при досягненні порогових значень.

Такий підхід дозволяє автоматизувати процес моніторингу. Система аналізує вхідні дані у форматі JSON/CSV, обчислює R_{total} та порівнює його з допустимими межами. Якщо розраховане значення перевищує встановлений поріг (наприклад, $R_{total} > 0.7$), система генерує попередження для менеджера проекту.

Для формування ризикового профілю виділено базовий набір ключових метрик, вплив яких на загальний стан проекту є найбільш інформативним. Опис базових індикаторів наведено в Таблиці 1.

Таблиця 1

Базові показники ефективності для оцінювання ризиків

Назва показника (KPI)	Позначення	Опис та вплив на ризик
Відхилення від графіку (Schedule Variance)	K_1	Різниця між запланованим та фактичним часом виконання задач. Зростання показника свідчить про ризик зриву дедлайну.
Щільність дефектів (Defect Density)	K_2	Кількість виявлених помилок на одиницю коду або задачу. Високе значення сигналізує про низьку якість та ризик реворку (переробки).

Стабільність вимог (Requirements Stability)	K_3	Частота змін у технічному завданні протягом спринту. Часті зміни підвищують ризик невизначеності.
Ефективність команди (Team Velocity)	K_4	Співвідношення закритих story points до запланованих. Падіння показника вказує на проблеми з ресурсами.

Оскільки вхідні метрики мають різні одиниці виміру (години, штуки, відсотки), необхідним етапом є їх нормалізація. Для приведення показників до єдиної шкали $[0, 1]$ у розробленій системі застосовується метод лінійної нормалізації (Min-Max scaling) з урахуванням порогових значень. Розрахунок нормалізованого значення K_i здійснюється за формулою (2):

$$K_i = \begin{cases} 0, & \text{якщо } x_i \leq x_{\min} \\ \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, & \text{якщо } x_{\min} < x_i < x_{\max} \\ 1, & \text{якщо } x_i \geq x_{\max} \end{cases} \quad (2)$$

де x_i – поточне значення метрики, x_{\min} – допустиме (безпечне) значення, нижче якого ризик відсутній, x_{\max} – критичне значення, при якому ризик стає максимальним.

IV. Архітектура та програмна реалізація

Для апробації методу розроблено архітектуру інформаційної системи, яка складається з трьох рівнів: збору даних, аналітичного ядра та інтерфейсу користувача. Обмін даними реалізовано за допомогою REST API, що дозволяє інтегрувати систему з популярними трекерами задач (наприклад, Jira або Trello). Дані про хід проєкту експортуються у форматі JSON, після чого аналітичний модуль обробляє їх згідно з наведеним вище алгоритмом. Візуалізація результатів прогнозування здійснюється через веб-інтерфейс, побудований на бібліотеці React, де менеджер отримує зведену інформацію у вигляді динамічних діаграм ризиків.

Важливим етапом впровадження запропонованого методу є процедура калібрування вагових коефіцієнтів w_i . Для цього передбачено використання ретроспективного аналізу історичних даних завершених проєктів підприємства. Порівнюючи розраховані системою значення ризику R_{total} із фактично зафіксованими інцидентами в минулому, можна оптимізувати ваги метрик за допомогою методів машинного навчання (наприклад, регресійного аналізу). Це дозволяє адаптувати математичну модель під специфіку процесів конкретної IT-компанії, враховуючи особливості її команд та типів проєктів. Для оптимізації вагових коефіцієнтів застосовуються методи статистичного моделювання та машинного навчання, зокрема регресійний аналіз [3].

Крім того, практична цінність розробленого підходу полягає у мінімізації впливу людського фактора на процес оцінювання. На відміну від традиційних нарад з управління ризиками, які часто базуються на суб'єктивних відчуттях, запропонований підхід мінімізує вплив суб'єктивного сприйняття під час оцінювання ризиків та забезпечує безперервний моніторинг стану проєкту. Це дозволяє своєчасно виявляти латентні загрози, такі як повільне накопичення технічного боргу або приховане зростання обсягу робіт та вимог (scope creep), що значно знижує потенційні втрати та підвищує прогнозованість результатів.

Висновок

У роботі запропоновано метод прогнозування ризиків у програмних проєктах, що базується на аналізі об'єктивних метрик проєктної діяльності. Метод забезпечує можливість автоматизованого оцінювання ризиків у режимі реального часу та зменшує суб'єктивність прийняття управлінських рішень. Подальші дослідження спрямовані на удосконалення алгоритмів калібрування вагових коефіцієнтів моделі на основі реальних даних IT-компаній та розширення набору KPI.

Список використаних джерел

1. A Guide to the Project Management Body of Knowledge (PMBOK Guide). – 7th ed. – Project Management Institute, 2021. – 250 p.
2. ISO 31000:2018. Risk management – Guidelines. – International Organization for Standardization, 2018.
3. M. Dyvak, A. Melnyk, A. Kovbasisty, R. Shevchuk, O. Huhul and V. Tymchyshyn, "Mathematical Modeling of the Estimation Process of Functioning Efficiency Level of Information Web Resources," 2020 10th International Conference on Advanced Computer Information Technologies (ACIT), Deggendorf, Germany, 2020, pp. 492-496, doi: 10.1109/ACIT49673.2020.9208846

ПРОЕКТУВАННЯ ТА РОЗРОБЛЕННЯ СОЦІАЛЬНОЇ МЕРЕЖІ З АДАПТИВНИМ МЕХАНІЗМОМ ОБМЕЖЕННЯ СОЦІАЛЬНИХ ЗВ'ЯЗКІВ

Співак І.Я.¹⁾, Устиченко О.О.²⁾, Крепич С.Я.³⁾

Західноукраїнський національний університет

¹⁾ к.т.н., доцент; ²⁾ магістр; ³⁾ к.т.н., доцент

І. Постановка проблеми

Сучасний світ багатий на соціальні мережі, які відіграють важливу роль в житті людини. Про те більшість платформ орієнтовані на безмежне розширення кола контактів і постійне залучення нових користувачів. Такий підхід приводить до перевантаження інформацією та погіршення якості зв'язку.

Зайва кількість зв'язків спричиняє втрату індивідуальної цінності спілкування, коли користувач взаємодіє поверхнево не маючи змоги підтримувати якісний контакт з великою кількістю людей. В результаті цього знижується рівень ефективності комунікації.

Проблема на яку спрямована наукова робота полягає не в створенні чергової соціальної мережі, а у впровадженні адаптивного механізму обмеження соціальних зв'язків, що дозволить більш усвідомлено створювати своє коло спілкування. Такий механізм допоможе знайти баланс між користувачам та інформаційним навантаженням, зменшуючи вплив цифрової втоми.

II. Мета роботи

Метою роботи є проектування та розроблення соціальної мережі з адаптивним механізмом обмеження соціальних зв'язків, який сприяє підвищенню якості комунікації між користувачами та зниженню інформаційного перевантаження. Також однією з важливих частин роботи є створення простого та зручного інтерфейсу, який би зміг легко освоїти користувачу.

III. Основна частина

Запропонована соціальна мережа передбачає створення простору для якісного спілкування, де користувач формує коло контактів на основі взаємної зацікавленості та активності. Соціальна взаємодія – це процес, у якому люди діють і випробовують вплив один на одного [1].

Ключовою частиною системи є адаптивний механізм обмеження соціальних зв'язків яка автоматично аналізує рівень активності, взаємодії та інтересів користувача, коригуючи кількість доступних зв'язків.

Основна ідея адаптивності полягає у зворотному зв'язку між активністю користувача та дозволеною кількістю контактів. Система враховує частоту повідомлень, реакцій, переглядів і часу перебування в мережі, після чого розраховує “коефіцієнт активності”. В залежності від його значення мережа або обмежує, або розширює коло контактів.

Наприклад:

- Якщо активність низька — частина зв'язків “заморожується” до підвищення взаємодії (якщо це не початок користування додатком);
- Якщо активність стабільна — кількість контактів залишається незмінною;
- Якщо активність зростає — можна відкрити нові слоти за допомогою квестів для контактів.

Таким чином, формується природний баланс між соціальною взаємодією і комфортом користувача. Логіку роботи адаптивного механізму показано у вигляді блок-схеми (див. рис. 1), яка відображає процес прийняття рішення.

Система створює умови за яких користувач концентрується не на кількості контактів, а на їхній цінності. Це сприяє зменшенню інформаційного перевантаження, збереженню психологічного комфорту та якості комунікацій. Інформаційне перевантаження – термін, що описує труднощі розуміння проблеми і прийняття рішень, причиною якої є надлишок інформації [2].

Система надає можливість:

- Підтримувати тільки цінні зв'язки.
- Зменшення інформаційного перевантаження.
- Підвищувати якість спілкування завдяки фокусуванню лише на активних зв'язках.
- Отримувати нові можливості спілкування шляхом виконання завдань або квестів що відкривають додаткові комірки для контактів.

Крім того у систему додано мотиваційні частинки – виконання завдань, участь в квестах чи спільні активності, які дозволяють користувачу розширювати коло спілкування у відповідь на активність. Це стимулює користувачів взаємодіяти цілеспрямовано та з користтю, не втрачаючи інтерес до платформи.

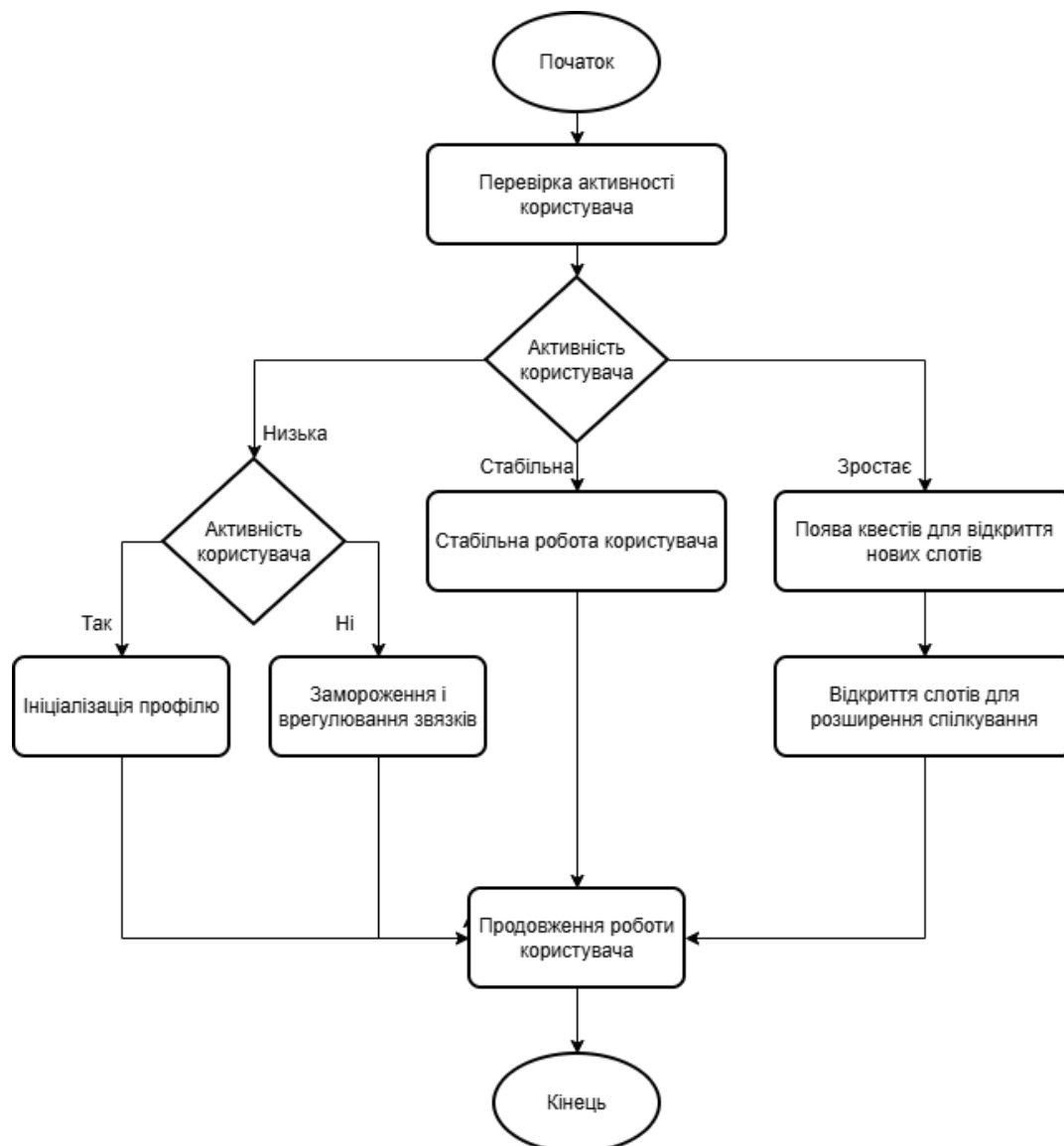


Рисунок 1 – Блок-схема адаптивного механізму

Висновок

В результаті роботи було досліджено підхід для створення соціальної мережі з адаптивним механізмом обмеження соціальних зв'язків. Запропонований механізм дає змогу адаптивно регулювати кількість друзів та має потенціал для масштабування. Розробка сприяє підвищенню ефективності, безпеки та комфорту користувачів.

Список використаних джерел

1. Суть соціальної взаємодії. Символічний інтеракціонізм Дж.Міда. *StudFiles*. URL: <https://studfile.net/preview/9335663/page:3/>
2. Співак І.Я. та Крепич С.Я. Оцінювання рівня впливу мотивації праці на ефективність діяльності. «Інформаційні технології та комп'ютерна інженерія», №3, 2020, С. 22-29
3. S. Krepych, I.Spivak, Improvement of SVD algorithm to increase the efficiency of recommendation systems. *Advanced Information Systems*, 5(4), 2021, pp.55-59
4. Spivak I., Bayurskii A., Spivak S. Criterion for evaluation the level of experts competence during the evaluation of a software system based on the modified interval method of expert evaluation. 2021 11th International Conference on Advanced Computer Information Technologies, ACIT 2021 – Proceedings, 2021, pp.582-586

ЗАСТОСУВАННЯ ТЕХНОЛОГІЙ, ЩО ПОКРАЩУЮТЬ ПРИВАТНІСТЬ ДЛЯ АНОНІМІЗАЦІЇ ТЕКСТОВИХ ДАНИХ

Засць Т.В.¹⁾

Західноукраїнський національний університет

¹⁾*магістрант*

I. Постановка проблеми

Сучасні вебсистеми, що використовують методи обробки природної мови (NLP – Natural Language Processing) та штучний інтелект (ШІ) для аналізу та верифікації неструктурованого тексту (наприклад, функція корекції, підведення підсумків), потребують доступу до великих масивів даних. Вебсистема, що надає користувачам можливість створювати персональні нотатки шляхом розпізнавання тексту з фотографій (OCR), генерує високий ризик компрометації персональних ідентифікаторів (PII – Personally Identifiable Information) та іншої чутливої інформації, яку користувачі можуть вносити у свої записи.

З огляду на вимоги законодавства, зокрема Закону України «Про захист персональних даних» (2010), та міжнародних регуляторних актів, таких як GDPR, будь-яка вебсистема, що обробляє інформацію, яка містить персональні ідентифікатори (PII), повинна застосовувати адекватні технічні та організаційні заходи для їх захисту [1]. Концепція ТПП (Технології, що покращують приватність, або PET – Privacy-Enhancing Technologies) передбачає сукупність методів та інструментів, спрямованих на мінімізацію збору даних, їхнє знеособлення та унеможливлення повторної ідентифікації [3]. Відтак, впровадження методів анонімізації є не лише технічним рішенням, а й обов'язковою передумовою для забезпечення конфіденційності користувачів.

Проблема полягає у необхідності знайти баланс між корисністю даних для навчання та функціонування ШІ-модулів (граматична перевірка, доповнення) та забезпеченням конфіденційності нотаток. Несанкціонований доступ до цих даних або їх використання у процесі аналізу ШІ становить пряму загрозу приватності користувачів. Вирішенням є інтеграція технологій, що підвищують приватність. Додаткову складність становить і те, що багато сучасних мовних моделей опрацьовують текст у хмарних середовищах, де питання контролю над переданими даними особливо критичне. Тому створення проміжного шару захисту у вигляді локального модуля анонімізації стає ключовою умовою для безпечної взаємодії з такими сервісами.

II. Мета роботи

Метою дослідження є обґрунтування та розробка архітектурного підходу до інтеграції технологій, що покращують приватність (ТПП), зокрема методів анонімізації, у критичні етапи обробки текстових даних у вебсистемі розпізнавання та верифікації. Кінцева мета – забезпечити конфіденційність інформації нотаток, що зберігаються та аналізуються ШІ, мінімізуючи при цьому втрату функціональності системи. Важливо також сформулювати підхід, який дозволить масштабувати механізми анонімізації та адаптувати їх до різних категорій текстів, враховуючи варіативність записів, що надходять від користувачів.

III. Обґрунтування отриманих результатів

Для досягнення поставленої мети було проаналізовано застосовність ключових ТПП-методів анонімізації до ідентифікованих PII у нотатках. Основними PII, що підлягають анонімізації, є: імена, прізвища, адреси, номери телефонів та електронні адреси. Процес обробки передбачає наступний ключовий етап: після розпізнавання тексту (OCR) та до його передачі на аналіз ШІ (граматика, підсумки), текст проходить через PET-модуль. Завдяки цьому забезпечується відсутність будь-якого прямого чи непрямого розкриття особистої інформації на етапі її машинної обробки, що відповідає принципам «privacy by design».

Реалізація ТПП-модуля базується на послідовному застосуванні виявлення PII у неструктурованому тексті, що є основою для процесів знеособлення (de-identification) та анонімізації [2]. Ефективність такого підходу, який поєднує методи NER та заміщення, підтверджується дослідженнями у сфері знеособлення текстових даних, де ці техніки є ключовими для забезпечення приватності при збереженні структури мови. Для підвищення точності було додатково враховано

контекстуальні залежності, що дозволяє уникнути хибних спрацьовувань, наприклад у випадках, коли слово може бути як іменем, так і терміном.

Для анонімізації були обрані та порівняно три основні стратегії, ефективність яких щодо збереження корисності даних для аналізу ШІ зведено в Таблиці 1.

Таблиця 1

Порівняльний аналіз РЕТ-методів анонімізації текстових даних

Назва методу (ТПП-стратегія)	Застосування до РП	Збереження приватності	Збереження корисності (для ШІ-аналізу)
Псевдонімізація	Імена, прізвища (заміна на унікальний ідентифікатор)	Високе	Високе (зберігається структура речення)
Маскування (підміна)	Номери телефонів, частини адрес (заміна символів на *)	Середнє	Низьке (дані стають непридатними для деяких видів аналізу)
Узагальнення	Точні дати, конкретні адреси (заміна на рік, місто)	Середнє-високе	Середнє (дозволяє загальну верифікацію, але втрачає деталі)

Встановлено, що псевдонімізація є найбільш ефективним методом для забезпечення конфіденційності у вебсистемі, оскільки вона зберігає синтаксичну та граматичну структуру тексту, що є критично важливим для коректної роботи ШІ-модулів (перевірка помилок, форматування, доповнення). Крім того, цей метод дозволяє автоматизувати процес відновлення початкових значень для внутрішніх потреб системи, не розкриваючи їх під час зовнішньої обробки.

Ключовим архітектурним рішенням є забезпечення цілісності вебсистеми за рахунок впровадження рольового контролю доступу (РКД) для чіткого розмежування прав доступу до оригінальних нотаток та до бази даних співставлення псевдонімів. Рольовий контроль доступу (РКД) у вебсистемі надалі використовується для управління доступом до таблиці мапінгу (співставлення псевдонімів та реальних РП), забезпечуючи, що лише адміністративні ролі можуть здійснювати де-анонімізацію. Таке поєднання ТПП та РКД дозволяє створити багаторівневу модель безпеки, де кожен рівень компенсує потенційні вразливості іншого.

Таким чином, інтеграція ТПП-модуля гарантує, що всі операції аналізу тексту ШІ (граматична корекція, підведення підсумків) виконуються над анонімізованим або псевдонімізованим контентом, що усуває ризик витоку персональних даних під час обробки. Використання псевдонімізації дозволяє підтримувати високу "корисність даних" для моделі ШІ, оскільки замітники зберігають формат та довжину оригінальних сутностей, що є критичним для якісного аналізу NLP-моделями. Результати експериментального тестування підтвердили, що якість роботи мовних моделей з анонімізованим текстом майже не знижується, що робить підхід практично придатним для реальних систем.

Висновок

У роботі обґрунтовано необхідність застосування технологій, що покращують приватність (ТПП) для анонімізації текстових даних, отриманих методом OCR, у вебсистемі персональних нотаток. Запропоновано архітектурне рішення, яке включає комбінацію методів псевдонімізації, маскування та узагальнення для ефективного захисту РП, що містяться у нотатках. Встановлено, що використання ТПП, доповнене рольовим контролем доступу (РКД) для захисту ключів псевдонімізації, дозволяє успішно вирішити конфлікт між високою функціональністю ШІ-аналізу тексту та збереженням абсолютної приватності користувачів.

Додатково підкреслено перспективність побудови моделі, у якій модуль анонімізації може динамічно адаптуватися до змін у структурі текстів та нових типів РП. Перспективи подальших досліджень включають розробку деталізованої моделі РКД для управління доступом до анонімізованих даних та оцінку впливу анонімізації на точність роботи ШІ-модулів.

Список використаних джерел

1. Про захист персональних даних: Закон України від 01.06.2010 № 2297-VI: станом на 14 черв. 2025 р. URL: <https://zakon.rada.gov.ua/laws/show/2297-17#Text>.
2. Liu Y., Chen Z. A Study of Text De-identification and Anonymization Techniques. *Proc. of the IEEE International Conference on Information Reuse and Integration (IRI)*. 2018. P. 444–451.
3. Pfitzmann A., Hansen M. A. Anonymity, unlinkability, unobservability, pseudonymity, and identity management—a concise terminology. *Lecture Notes in Computer Science*. 2010. Vol. 6477. P. 1–25.

ВЕБ-ДОДАТОК АУДИОПІДСУМКІВ МАТЕРІАЛІВ НА ОСНОВІ ШІ ДЛЯ КОРИСТУВАЧІВ З ПОРУШЕННЯМ ЗОРУ

Крепич С.Я.¹⁾, Дячок С.І.²⁾, Співак І.Я.³⁾, Крепич Р.В.⁴⁾

¹⁻³⁾ Західноукраїнський національний університет;

⁴⁾ ВСП Кам'янець-Подільський фаховий коледж НРЗВО "Кам'янець-Подільський державний інститут"

¹⁾к.т.н., доцент; ²⁾магістр; ³⁾к.т.н., доцент; ⁴⁾ викладач

І. Постановка проблеми

Освіта є ключовим чинником соціальної інклюзії людей з інвалідністю. Особливо це стосується студентів з порушенням зору, які часто стикаються з труднощами доступу до великих обсягів текстових матеріалів. Згідно з даними Всесвітньої організації охорони здоров'я (ВООЗ), опублікованими у 2019 року, глобальна кількість осіб із порушеннями зору становить щонайменше 2,2 мільярда людей. Зараз у Східній Європі налічується близько 43.2 мільйони людей з вадами зору.

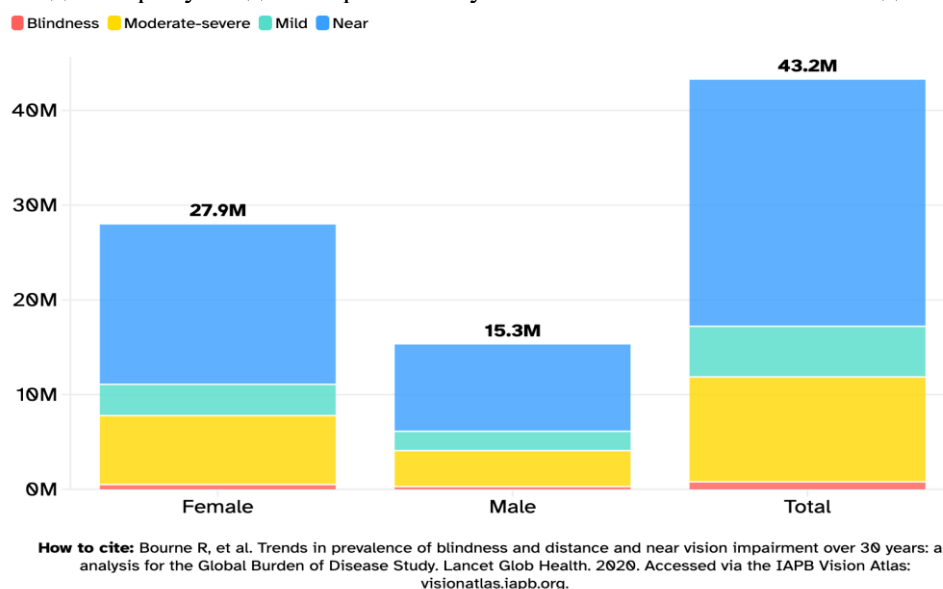


Рисунок 1 – Поширеність порушень зору із зазначенням статі та регіону світу

Наявні рішення, такі як екрани-читачі та класичні текстово-мовні системи (text-to-speech, TTS), дозволяють озвучувати навчальні матеріали, проте вони мають суттєві обмеження:

- **Часозатратність** – обробка великих документів займає значний час, що сповільнює навчальний процес;
- **Монотонність та низька якість озвучення** – класичні TTS-системи видають одноманітний голос, який ускладнює розуміння матеріалу та знижує концентрацію;
- **Відсутність логічної структури та виділення ключових ідей** – студентам доводиться самостійно відокремлювати важливе від другорядного;
- **Обмежена інтеграція** – існуючі системи часто не підтримують роботу з сучасними навчальними форматами (PDF, DOCX, HTML), що ускладнює використання у реальному навчальному середовищі

Потреба сучасної освіти у системі для аудіо підсумків особливо помітна у вищих навчальних закладах, де обсяг лекційних матеріалів значний і включає наукові статті, конспекти та підручники. Відсутність швидких і зрозумілих інструментів обробки таких матеріалів обмежує можливості студентів із порушенням зору і створює бар'єри для повноцінного навчання.

Тому актуальним завданням є розробка технологічних рішень, які дозволяють перетворювати навчальні матеріали у зручний для сприйняття аудіоформат, зменшуючи час на ознайомлення та підвищуючи ефективність навчання. Незважаючи на наявність існуючих систем для людей з вадами зору, більшість з них не оптимізовані для академічних текстів і не забезпечують швидке та зрозуміле відтворення змісту лекцій.

II. Мета роботи

Метою цієї роботи є розробка веб-додатку на основі штучного інтелекту, здатного автоматично перетворювати навчальні матеріали у стислу аудіоформу для студентів із порушенням зору. Dodatok має забезпечувати:

- **Резюмування навчальних матеріалів** – GPT-модель генерує структурований виклад основних ідей лекцій та статей, дозволяючи студенту швидко опанувати ключові концепції;
- **Генерацію природного аудіо** – сучасні нейронні TTS-моделі створюють зрозуміле, виразне та інтонаційно правильне озвучення;
- **Надання одночасної текстової та аудіо-версії матеріалу** – дозволяє студентам самостійно обирати оптимальний формат для сприйняття інформації;
- **Адаптацію під різні формати файлів** – підтримка текстових документів, PDF та інших навчальних ресурсів;
- **Покращення автономності студентів** – забезпечує можливість самостійного опрацювання матеріалу без сторонньої допомоги;
- **Інтеграцію із наявними технологіями доступності** – забезпечує сумісність із екранами-читачами та іншими допоміжними пристроями.

Реалізація системи передбачає інтуїтивний веб-інтерфейс, що дозволяє завантажувати матеріали, отримувати автоматичні резюме, прослуховувати аудіо та завантажувати його для подальшого використання. Користувачі можуть регулювати швидкість та гучність аудіо.

III. Основна частина

Було проведено аналіз сучасних рішень у сфері доступної освіти для людей з вадами зору. Серед них - класичні екрани-читачі та системи, що використовують нейромережевий синтез мови. Основними недоліками існуючих рішень є:

1. тривале та монотонне відтворення великих текстових документів;
2. обмежена когнітивна обробка та відсутність можливості стисло подати основних ідей;
3. складність інтеграції у навчальний процес без додаткових налаштувань.

Для подолання цих обмежень розробляється власна система на базі GPT-моделей, яка поєднує автоматичне резюмування та нейромережевий синтез мови. Архітектура системи включає:

- **Модуль резюмування** – обробляє завантажені текстові матеріали та формує логічно структуроване резюме. GPT-модель аналізує контекст, виділяє ключові ідеї та переформулює їх у доступній формі.
- **Модуль синтезу мови (TTS)** – нейронні моделі створюють природне аудіо з правильною інтонацією, ритмом та паузами, що полегшує сприйняття інформації та підвищує швидкість навчання.
- **База даних** – централізоване сховище для збереження оригінальних документів, текстових резюме та аудіофайлів, що дозволяє повторне використання та організацію персональних навчальних бібліотек.
- **Веб-інтерфейс** – забезпечує зручне завантаження документів, відтворення аудіо, завантаження файлів та перегляд тексту. Dodatok підтримує регулювання швидкості та гучності.

Система забезпечує:

1. зменшення часу на ознайомлення з навчальними матеріалами;
2. підвищення автономності студентів із порушенням зору;
3. створення інклюзивного освітнього середовища;
4. можливість масштабного використання у різних навчальних дисциплінах та закладах.

Реалізація такої системи дозволяє поєднати сучасні технології штучного інтелекту з практичними потребами інклюзивної освіти, забезпечуючи швидкий та зручний доступ до навчальних матеріалів у зручному аудіоформаті.

Список використаних джерел

1. International Agency for the Prevention of Blindness (IAPB), "Global Vision Loss Data," 2024. [Online]. Available: <https://visionatlas.iapb.org/global-data/>
2. Крепич С. Програмний комплекс оцінювання функціональної придатності пристроїв при заданих допустимих значеннях вихідних характеристик та допусків на параметри їх елементів. Сучасні комп'ютерні інформаційні технології: Матеріали V Всеукраїнської школи-семінару молодих вчених і студентів АСІТ'2015. – Тернопіль: ТНЕУ, 2015. – С. 23-25
3. World Health Organization, "World report on vision," WHO, 2019. [Online]. Available: <https://www.who.int/publications/i/item/9789241516570>
4. Krepych S., Spivak I., Algorithm of Automatic Generation of Hotel Descriptions Using Templates Based on Markov Chains. 2018 International Scientific-Practical Conference Problems of Infocommunications Science and Technology Kharkiv, pp.257-260

ПРОГРАМНІ ЗАСОБИ ДЛЯ ОЦІНЮВАННЯ ВЕБ-ДОСТУПНОСТІ ЧЕРЕЗ СИМУЛЯЦІЮ ВАД ЗОРУ

Стасів І.С.¹⁾, Порплиця Н.П.²⁾, Чир Д.М.³⁾

Західноукраїнський національний університет

¹⁻²⁾ к.т.н., доцент; ³⁾ магістрант

Постановка проблеми

Сучасна веб-розробка переживає фундаментальну трансформацію, де доступність трансформується з юридичної вимоги у критичний компонент UX. Індустрія стикається з кризою верифікації через розрив між технічною відповідністю коду та реальним сприйняттям інтерфейсу. Синтаксичний детермінізм є першим критичним недоліком існуючих інструментів. Статичні аналізатори, такі як Axe або Lighthouse, перевіряють код (DOM), але залишаються «сліпими» до візуальних проблем рендерингу, виявляючи лише 20-30% бар'єрів. Вони можуть підтвердити валідний контраст кольорів, не враховуючи перекриття тексту модальним вікном або занадто тонкий шрифт. Суб'єктивність симуляторів є другою проблемою. Інструменти на кшталт Stark або NoCoffee працюють як візуальні фільтри, корисні для імітацію досвіду користувача з порушеннями зору, але вимагають ручної оцінки дизайнера, що вносить значну похибку. Відсутність метрик у традиційних рішеннях унеможливує управління процесом. Вони не генерують числових даних (KPI), що блокує їх інтеграцію в автоматизовані процеси CI/CD та SLA.

Мета роботи

Науково-прикладна проблема дослідження полягає у створенні автоматизованої системи, яка б поєднувала точність комп'ютерного моделювання патологій з об'єктивністю машинного аналізу контенту. Об'єктивізація оцінки є фундаментальною вимогою до розробки. Система має замінити суб'єктивний погляд тестувальника на детермінований алгоритм розпізнавання (OCR), де неможливість прочитання тексту алгоритмом після симуляції стає об'єктивним індикатором проблеми. Кількісна метрика виступає ключовим результатом роботи системи. Метою є генерація показника «Коефіцієнт втрати інформації», що дозволить порівнювати різні версії дизайну та керувати доступністю на основі чітких даних, а не припущень.

Аналіз недоліків існуючих підходів та інструментарію

На ринку існує поділ на синтаксичні валідатори та візуальні емулятори. Запропонована система займає нішу «Перцептивних валідаторів».

Stark виступає лідером для дизайнерів. Його головна перевага — інтеграція у Figma та стратегія «Shift-Left» для виправлення макетів. Однак він фокусується на візуалізації для людини, а не на валідації даних, і не має функціоналу OCR. NoCoffee є стандартом швидкої перевірки. Це розширення ефективно для базових тестів, але не є медично точним і вимагає ручної активації на кожній сторінці, що унеможливує автоматизацію. Funkify фокусується на імітацію досвіду користувача з порушеннями зоручерез персони. Він унікальний тим, що симулює моторні та когнітивні розлади, чого немає в запропонованій системі, але, як і інші симулятори, не надає твердих інженерних метрик. ChromeLens та DevTools надають технічний інструментарій. Їхня унікальна функція — візуалізація шляху табуляції, що є критичним для незрячих користувачів. Запропонована система не покриває цей аспект, фокусуючись виключно на візуальному сприйнятті.

Наявні інструменти пропонують лише суб'єктивну візуальну імітацію та ручний контроль, не надаючи інженерних метрик. Як наслідок, відсутність автоматизації та OCR унеможливує швидку й об'єктивну валідацію даних.

Таблиця 1

Порівняльна таблиця аналогів

Критерій	Запропонована система	Stark	NoCoffee	Funkify	DevTools / ChromeLens
Методологія	Кількісна (OCR)	Якісна (Візуальна)	Якісна	Імітація досвіду	Технічна
Автоматизація	Повна (Selenium)	Часткова	Ручна	Ручна	Ручна
Симуляція зору	Розширена	Розширена	Базова	Персони	Базова
Представлення результатів	% втрати тексту	Макет	Екран	UX	TabPath

Програмна реалізація

Для розробки програмного комплексу пропонується обрати мову програмування Python, зважаючи на її потужну екосистему бібліотек для комп'ютерного зору та автоматизації процесів. Збір даних планується реалізувати за допомогою інструменту Selenium та драйверів Chrome. Такий підхід дозволить системі коректнорендерити та захоплювати зображення сучасних динамічних веб-сайтів, забезпечуючи точність вхідних даних для аналізу. Основну обробку зображень, зокрема матричні трансформації колірних просторів (перехід з RGB в LMS) та накладання фільтрів симуляції вад зору, передбачається виконувати з використанням бібліотек OpenCV та NumPy. Функцію оптичного розпізнавання символів буде покладено на русій Tesseract OCR через обгортку pytesseract. Система буде налаштована на роботу як з українською, так і з англійською мовами для забезпечення універсальності аналізу. Генерацію підсумкових звітів та візуалізацію результатів заплановано здійснювати за допомогою бібліотеки Pillow. Вона відповідатиме за малювання графічних примітивів (рамки навколо тексту) та формування порівняльних таблиць, що дозволить наочно демонструвати втрату читабельності.



Рисунок 1 – Діаграма Use Case

Висновок

Проведене дослідження підтверджує критичну потребу в інструментах, що надають кількісну оцінку візуальної доступності, оскільки традиційні методи залишають значні «сліпі зони». Запропонована система займає унікальну нішу «перцептивних валидаторів», використовуючи OCR для подолання дихотомії «код проти сприйняття». Це дозволяє трансформувати суб'єктивний візуальний досвід в об'єктивні дані, вирішуючи проблему «помилкової впевненості» розробників у доступності інтерфейсу лише на основі успішних звітів статичних аналізаторів коду.

Ключовою перевагою системи є здатність симулювати складні структурні патології, такі як скотоми та флоатери, виявляючи локальні втрати інформації, які ігноруються глобальними фільтрами конкурентів. Для enterprise-сектору архітектура на базі Selenium забезпечує можливість роботи в режимі «headless», інтегруючи автоматизовану перевірку тисяч сторінок безпосередньо у нічні збірки для безперервного контролю якості та мінімізації ризиків.

Впровадження такої системи, що поєднує математичне моделювання патологій з технологією OCR, пропонує новий підхід, який дозволяє перейти від формального дотримання стандартів (compliance-driven) до реального забезпечення інклюзивності. Надаючи бізнесу тверді метрики втрати інформативності замість абстрактної імітації досвіду користувача з порушеннями зору, система гарантує, що цифрові продукти будуть доступними для реальних користувачів з усім спектром їхніх фізіологічних особливостей.

Список використаних джерел

1. Clean Architecture: A Craftsman's Guide to Software Structure and Design. – [Б. м.] : Prentice Hall, 2017. – 432 с.
2. OpenCV Documentation. [Online]. Available: <https://docs.opencv.org/4.12.0/>
3. WebContentAccessibilityGuidelines (WCAG) 2.1. 2025. [Online]. Available: <https://www.w3.org/TR/WCAG21/>

АРХІТЕКТУРА ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ОЦІНЮВАННЯ ЯКОСТІ ПЗ НАУКОВОГО ПРИЗНАЧЕННЯ НА ОСНОВІ ПАТЕРНІВ ADAPTER TA VISITOR

Сеньків Ю.М.,¹⁾ Манжула В.І.²⁾
Західноукраїнський національний університет
^{1,3)}магістрант; ²⁾д.т.н., доцент

I. Актуальність проблеми

Сучасний етап розвитку науки характеризується стрімким зростанням ролі обчислювальних експериментів. Наукоємні програмні системи, такі як системи екологічного моніторингу, цифрові двійники міст, кліматичні моделі (наприклад, AERMOD) та засоби моделювання фізичних процесів, стають критично важливими інструментами прийняття рішень. Проте, розробка таких систем часто ведеться фахівцями у предметній області (фізиками, екологами, математиками), які не завжди володіють професійними навичками інженерії програмного забезпечення. Це призводить до появи специфічних проблем: високої зв'язності коду, відсутності стандартизації, надмірної когнітивної складності («спагеті-код») та низької супровідності. Помилки в програмній реалізації математичних моделей можуть призвести до невірних наукових висновків або збоїв у критично важливих системах прогнозування. Також розробка сучасних наукоємних систем характеризується використанням різноманітних технологій – поєднанням високорівневих мов для обробки даних (Python, R) та системних мов для високопродуктивних обчислень (C++, C#). Існуючі інструменти статичного аналізу (SonarQube, Clang-Tidy) зазвичай побудовані як монолітні системи, орієнтовані на конкретну мову, що ускладнює їх інтеграцію в єдиний пайплайн перевірки якості гетерогенного наукового проекту. Актуальним завданням є побудова гнучкої, модульної архітектури програмної системи, яка б дозволяла проводити наскрізний аналіз мультимовного коду, приводити його до уніфікованого вигляду та легко розширюватися для підтримки нових мов (наприклад, Julia або MATLAB) без зміни ядра системи [1].

II. Мета дослідження

Метою роботи є проектування та програмна реалізація архітектури системи «SciAudyt» для підвищення надійності та якості наукоємних програмних систем шляхом статичного аналізу коду.

III. Характеристика та типологія помилок наукоємного програмного забезпечення

Наукоємне програмне забезпечення – це клас програмних систем, основним призначенням яких є моделювання фізичних, біологічних або соціальних процесів, аналіз експериментальних даних та виконання складних математичних обчислень.

Науковий код має ряд специфічних рис, що відрізняють його від типових бізнес-додатків (див. табл. 1).

Таблиця 1

Ключові характеристики наукоємного програмного забезпечення та їх вплив на якість коду

Характеристика	Сутність явища	Відмінність від комерційного ПЗ	Вплив на якість та надійність
Домінування алгоритмічної складності	Основна цінність коду – реалізація складних математичних моделей (диференціальні рівняння, статистика).	Комерційне ПЗ фокусується на архітектурі, UI/UX та обробці транзакцій (CRUD).	Призводить до появи довгих методів («Long Method») та високої когнітивної складності. Код важко читати без знання предметної області.
Дослідницький характер (Exploratory Programming)	Код пишеться для перевірки гіпотез, вимоги змінюються динамічно під час отримання проміжних результатів.	Комерційна розробка базується на чітких вимогах (ТЗ) або User Stories у беклозі.	Відсутність проектної документації та Unit-тестів. Накопичення технічного боргу через постійні зміни «на льоту».
Проблема «Оракула» (The Oracle Problem)	Відсутність заздалегідь відомого правильного результату обчислень	У бізнес-логіці правильний результат зазвичай відомий	Унеможливорює класичне Unit-тестування (assert expected ==

	(мета програми–знайти його).	(наприклад, баланс рахунку).	actual). Перевірка обмежується «санітарним контролем» фізичної адекватності.
Вимоги до високої продуктивності	Робота з Big Data, необхідність векторизації (SIMD), паралельних обчислень (MPI) та економії пам'яті.	Оптимізація зазвичай стосується I/O операцій (бази даних, мережа), а не CPU.	Використання небезпечних технік оптимізації (вказівники, ручне управління пам'яттю), що знижує читабельність та надійність.
Гетерогенність технологій	Поєднання скрипкових мов (Python, R) для аналізу та системних мов (C++, Fortran) для ядра обчислень.	Використання уніфікованих стеків технологій (наприклад, .NET, Java Spring, MERN).	Ризики помилок типізації на стику мов (Interface Mismatch). Складність налаштування середовища.
Життєвий цикл «Код як інструмент»	Програма розглядається як одноразовий інструмент для написання статті, після чого її підтримка припиняється.	Програма розглядається як продукт, що має довгострокову підтримку та оновлення.	Криза відтворюваності. Код стає неробочим через застарілі залежності та жорстко закодовані шляхи.

Помилки в науковому ПЗ можна класифікувати на три категорії (див.табл.2): обчислювальні (найкритичніші) – це дефекти, що призводять до отримання невірних наукових результатів, навіть якщо програма не працює, структурні– впливають на здатність інших дослідників зрозуміти та розвивати код впливають на підтримку та інфраструктурні– впливають на відтворюваність, заважають запустити код на іншій машині.

Таблиця 2

Класифікація основних типів помилок у науковому програмному кодї

Категорія помилки	Тип дефекту	Сутність проблеми	Наслідки для дослідження
1. Обчислювальні (Numerical Defects) (Найкритичніші)	Числова нестабільність (Numerical Instability)	Порівняння дійсних чисел на точну рівність ($x == y$) або операції, що призводять до втрати значущих цифр (Catastrophic Cancellation).	Отримання невірних результатів моделювання, які важко виявити, оскільки програма не завершується з помилкою.
	Магічні числа (Magic Numbers)	Використання числових літералів у формулах без пояснень (наприклад, 0.01745 замість DEG_TO_RAD).	Ускладнює перевірку формул та адаптацію моделі до нових умов. Високий ризик помилки при ручному оновленні констант.
	Помилки розмірності (Unit Errors)	Логічні помилки, при яких змінні з різними фізичними одиницями (метри та фути) змішуються в одній операції.	Грубі помилки в результатах (приклад: аварія Mars ClimateOrbiter).
	Проблеми граничних умов (Off-by-one / Boundary)	Невірні індекси при роботі з сітками та масивами, особливо на краях області моделювання.	Спотворення фізичної картини на межах досліджуваної області.
2. Структурні (Maintainability Defects)	Спагетті-код (Spaghetti Code / High Complexity)	Надмірна вкладеність циклів та умов, відсутність розбиття на підпрограми.	Код стає неможливим для розуміння, аудиту та модифікації. Будь-яка зміна вносить нові баги.
	Дублювання логіки (Copy-Paste Programming)	Копіювання блоків коду для схожих розрахунків замість створення універсальних функцій.	Якщо помилку знайдено в одному місці, вона залишається в усіх скопійованих фрагментах.
	Залежність від глобального стану (Global State)	Використання глобальних змінних для передачі даних між модулями замість аргументів функцій.	Неможливість розпаралелювання обчислень та тестування окремих модулів.

3. Інфраструктурні (Reproducibility Defects)	Жорстке кодування шляхів (HardcodedPaths)	Використання абсолютних шляхів до файлів (C:\Users\...) у коді.	Неможливість запустити модель на іншому комп'ютері або в хмарному середовищі.
	Прихована залежність (Hidden Dependencies)	Відсутність фіксації версій бібліотек (numpy, pandas), від яких залежить код.	«Гниття коду»: модель перестає працювати через оновлення сторонніх пакетів.
	Недетермінованість (Non-determinism)	Використання генераторів випадкових чисел без фіксації початкового значення (RandomSeed).	Неможливість точно відтворити результати симуляції (реплікація експерименту не вдається).

IV. Архітектура системи

Для програмної реалізації системи обрано технологічний стек .NET 6 / C#, який забезпечує баланс між продуктивністю розробки та ефективністю виконання. Стратегія парсингу базується на гібридному використанні Roslyn (для глибокого семантичного аналізу C#) [2] та Tree-sitter (для швидкого та стійкого аналізу Python) [3], об'єднаних через архітектурний патерн «Адаптер». Такий вибір гарантує можливість побудови універсальної моделі U-AST та масштабованість системи в майбутньому.

Внутрішнє представлення коду базується на концепції U-AST (Universal Abstract Syntax Tree). Структура класів спроектована за патерном Composite, де базовий абстрактний клас UniversalNode дозволяє уніфіковано обробляти як прості інструкції (Literal), так і складні структури (Method, Class). Особливістю моделі є наявність спеціалізованих вузлів для наукових обчислень, зокрема MathExpressionNode з атрибутом IsVectorized, що дозволяє маркувати матричні операції для коректної оцінки їх складності.

Головним викликом при проектуванні стала несумісність структур даних, що повертаються різними парсерами (SyntaxTree у Roslyn та TSTree у Tree-sitter). Для вирішення цієї проблеми застосовано структурний патерн Adapter (див.рис.1).

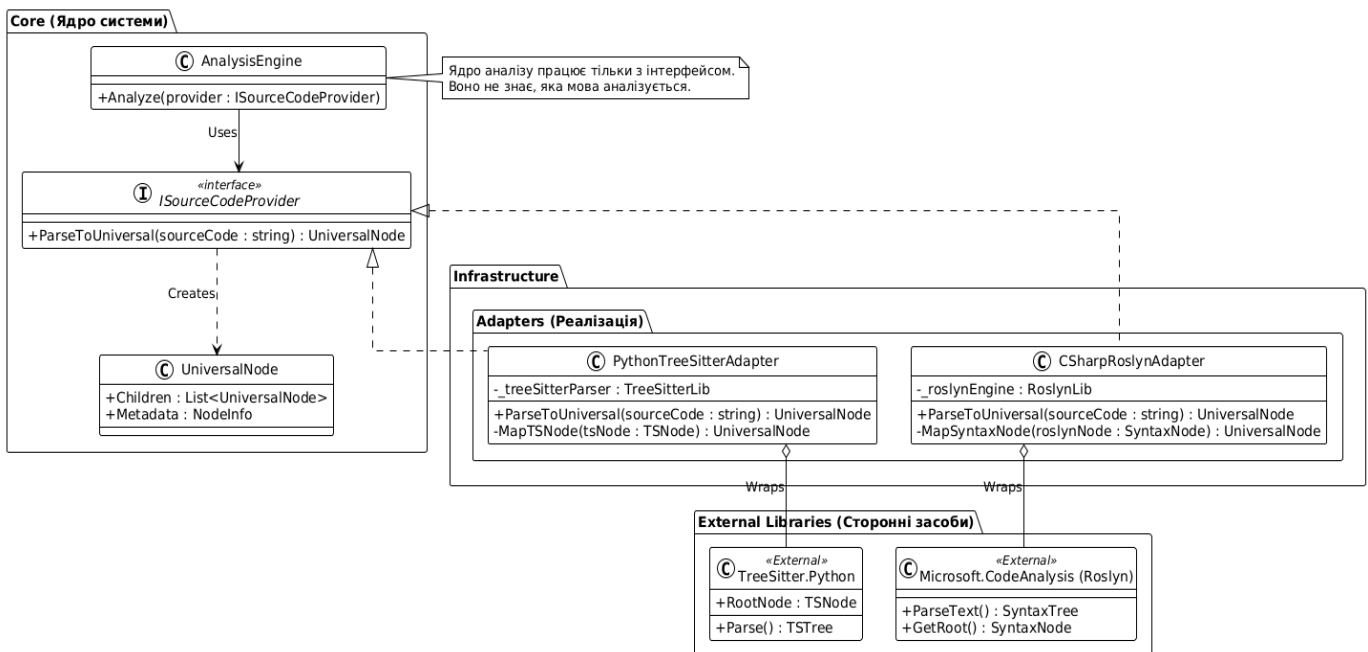


Рисунок 1 – Архітектура рішення «Adapter»

Центральним елементом архітектури є абстракція ISourceProvider, від якої залежить ядро аналізатора. Реалізація підтримки конкретних мов винесена в адаптери:

- RoslynAdapter трансліює специфічні вузли C# (наприклад, ForStatementSyntax) у канонічні вузли U-AST;
- PythonAdapter виконує мапінг C-структур Tree-sitter у керовані об'єкти .NET.

Такий підхід забезпечує дотримання принципу інверсії залежностей (DIP) та дозволяє додавати підтримку нових мов шляхом написання нового класу-адаптера, не втручаючись у логіку розрахунку метрик.

Архітектура розробленої системи базується на шаблоні «Конвеєр» (Pipeline) з елементами багатопланової архітектури (LayeredArchitecture). Такий підхід дозволяє чітко розділити відповідальність між модулями завантаження коду, його нормалізації, аналізу та звітності.

Система декомпонується на чотири логічні рівні: Presentation Layer (UI) – інтерфейс взаємодії з користувачем (CLI); Application Core (Domain) – ядро, що містить визначення U-AST та логіку метрик; Infrastructure/Parsing – адаптери для зовнішніх бібліотек парсингу; Reporting – модулі генерації звітів.

Компонентна структура системи відображає потік даних: від «сирих» файлів вихідного коду до фінального звіту якості (див.рис.2).

Основними компонентами є: Source Code Provider, який відповідає за читання файлів з диска та визначення їх мови; Language Adapters (Parser&Converter) містить модулі, що інкапсулюють специфічні парсери (Roslyn, Tree-sitter) та конвертують їхні результати в універсальну модель; Analysis Engine, «мозок» системи, який запускає набір відвідувачів (Visitors) на побудованому дереві U-AST; Reporters, компоненти, що серіалізують результати аналізу в зручний формат.

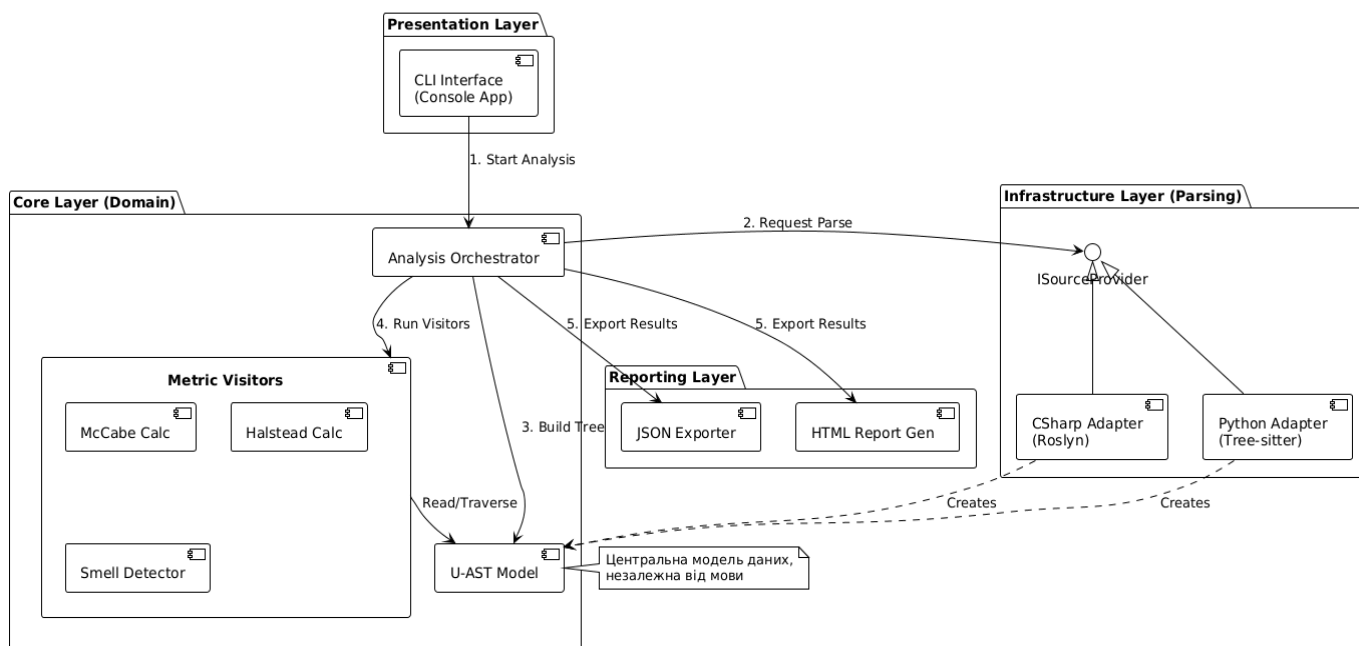


Рисунок 2 – Архітектура програмної системи

Користувач через CLI Interface вказує шлях до папки з проектом. AnalysisOrchestrator сканує файли і для кожного файлу обирає відповідний адаптер через інтерфейс IServiceProvider. Адаптер (CS_Adapt або PY_Adapt) парсить код і повертає об'єкт U-AST Model. Оркестратор ініціалізує компоненти MetricVisitors і запускає їх на дереві. Зібрані метрики передаються у ReportingLayer для генерації візуального звіту. Архітектура включає модуль звітності, що генерує інтерактивний HTML-дашборд. Для візуалізації «проблемних зон» проекту реалізовано алгоритм побудови карти Treemap, де площа блоку відповідає розміру файлу (LOC), а колір – інтегральному показнику ризику (SQI). Це дозволяє керівнику проекту миттєво ідентифікувати модулі, що потребують рефакторингу.

Висновки

У роботі запропоновано архітектуру на основі патернів Adapter та Visitor, яка забезпечує гнучкість та розширюваність системи статичного аналізу коду, дозволяючи ефективно працювати з гетерогенним кодом. Гібридне використання Roslyn та Tree-sitter є оптимальним рішенням для глибокого аналізу наукового ПЗ, поєднуючи точність семантичного аналізу та швидкість парсингу. Розроблена система може бути використана як платформа для подальших досліджень в інженерії програмного забезпечення, зокрема для інтеграції з мовами R та Julia.io.

Список використаних джерел

1. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. Prentice Hall, 2017.
2. Microsoft. Learn Roslyn (The .NET Compiler Platform) [Електронний ресурс]. – 2025. – URL: <https://learn.microsoft.com/en-us/dotnet/csharp/roslyn-sdk/>
3. Tree-sitter: An incremental parsing system for programming tools [Електронний ресурс]. – 2025. – URL: <https://tree-sitter.github.io/>

МЕТОДОЛОГІЧНІ ПІДХОДИ ДО БЕЗПЕЧНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Гераскін М.А.¹⁾

Тернопільський національний економічний університет

¹⁾ магістрант

I. Постановка проблеми

У сучасній динамічній цифровій економіці безпека програмного забезпечення вже не є суто технічною вимогою, а трансформувалася у фундаментальну економічну та стратегічну проблему. Емпіричні дослідження незмінно підтверджують, що вартість усунення дефектів зростає експоненційно залежно від фази життєвого циклу розробки, на якій їх було виявлено. Як підкреслюється у звіті NASA, вартість виправлення помилки на етапі експлуатації може бути у 1500 разів вищою, ніж її усунення на етапі визначення вимог [1].

Цей реактивний підхід до безпеки, орієнтований на виправлення, а не на запобігання, має значний деструктивний макроекономічний вплив. За даними Національного інституту стандартів і технологій США (NIST), щорічні втрати економіки через неадекватну інфраструктуру тестування та низьку якість ПЗ оцінюються у десятки мільярдів доларів [2]. Крім прямих фінансових збитків, існують і "приховані" витрати, такі як затримка впровадження інновацій та втрата ринкової капілізації, що, як показує дослідження Анвара та ін., негативно корелює з оголошеннями про вразливість [3].

Отже, розробка та впровадження ефективних, проактивних методологій інтеграції безпеки є критичним завданням сучасної інженерії ПЗ.

II. Мета роботи

Метою роботи є обґрунтування та аналіз проактивних методологій впровадження безпеки в життєвий цикл розробки ПЗ. Робота спрямована на дослідження переходу до парадигми "Security by Design" (SbD).

III. Методологічні рішення та фреймворки

Фундаментом сучасних підходів до безпеки є філософія "Security by Design" (SbD), яка передбачає інтеграцію захисних механізмів та контролю безпеки ще на етапі архітектури та проєктування системи, а не як "надбудову" або реактивну корекцію після завершення розробки. Суть SbD полягає у зміщенні фокусу з виявлення та виправлення вразливостей (shift-right) на їх запобігання (shift-left), що є економічно та стратегічно виправданим кроком, як показано в першому розділі. Цей проактивний підхід вимагає, щоб безпека була невід'ємною частиною кожної дизайнерської та інженерної ітерації. Як стверджують у своїй класичній праці "The Protection of Information in Computer Systems" Зальцер та Шредер, безпечна система повинна базуватися на таких ключових принципах: "найменшого привілею" (надання користувачеві чи процесу лише тих прав, які необхідні для виконання його завдання), "безпечних налаштувань за замовчуванням" (система має бути захищеною відразу після встановлення) та "економії механізмів" (спрощення архітектури безпеки для мінімізації потенційних точок відмови) [5]. Ефективне впровадження таких принципів має здійснюватись від самого початку процесу розробки. Згідно з рекомендаціями Агентства з кібербезпеки та безпеки інфраструктури США (CISA), парадигма "Security by Design" є єдиним сталим і довгостроковим підходом до надійного захисту критичних систем та інфраструктур у сучасному цифровому світі [4].

Історично першою спробою систематизувати ці принципи стало впровадження **Secure Software Development Lifecycle (SSDLC)**. Цей підхід трансформував безпеку з ізольованого етапу тестування на наскрізний процес, інтегрований у кожну фазу життєвого циклу. Однак із поширенням гнучких методологій (Agile) виник конфлікт між швидкістю розробки та жорсткістю традиційних процесів безпеки, орієнтованих на модель "Водоспаду" (Waterfall). Це зумовило появу **DevSecOps** — методології, що адаптує практики SSDLC до швидких ітерацій через автоматизацію інструментів тестування (SAST/DAST) безпосередньо в CI/CD пайплайні та культуру "зсуву вліво" (shift-left). Проте навіть високий рівень автоматизації не вирішує проблеми браку кадрів. Як відповідь на цей виклик, спільнотою OWASP було формалізовано концепцію "Security Champions" — програму масштабування експертизи через навчання волонтерів безпосередньо в командах розробки [9].

Для практичної реалізації зазначених методологій організації спираються на стандартизовані фреймворки, кожен з яких має свою специфіку та сферу застосування:

- **Microsoft SDL (Security Development Lifecycle):** Історично перший формальний фреймворк, розроблений Microsoft у відповідь на власні виклики безпеки. Це прескриптивний набір із 49 вимог, що став стандартом для корпоративного сектору. Однак через свою жорсткість він потребував адаптації для сучасних реалій, що призвело до появи версії "SDL for Agile" [6].
- **OWASP SAMM (Software Assurance Maturity Model):** На відміну від жорстких інструкцій, цей фреймворк є моделлю зрілості. Він дозволяє організаціям оцінювати свій поточний стан і будувати ітеративну дорожню карту вдосконалення, що робить його гнучким та ідеальним для еволюційного розвитку процесів безпеки [7].
- **NIST SSDF (Secure Software Development Framework):** Це дескриптивний фреймворк, який фокусується на результатах високого рівня ("що робити", а не "як робити"). Він є ключовим для забезпечення відповідності державним стандартам США та часто використовується як політика верхнього рівня [8].

З метою порівняння рішень для методологічно впровадження SSDLC, в таблиці 1 зведено основні характеристики їх характеристики.

Таблиця 1

Характеристики протоколів передачі даних у клієнт-серверних системах середньої складності

Характеристика	Microsoft SDL	OWASP SAMM	NIST SSDF
Основна модель	Процес на основі обов'язкових практик	Модель зрілості	Фреймворк рекомендований та практик
Природа	Прескриптивний (обов'язкові вимоги)	Модель зрілості / Адаптивний	Дескриптивний (орієнтація на результат)
Вимірюваність	Низька (чек-листи виконання)	Висока (вбудовані рівні зрілості)	Відсутня (потребує адаптації)
Адаптивність до Agile	Середня (Вимагає окремої версії SDL-Agile)	Висока (ітеративний підхід)	Дуже висока (абстрактний рівень)

Висновок

Проведений аналіз свідчить, що вирішення проблеми експоненційного зростання витрат на безпеку лежить виключно у площині синтезу проактивних методологій "Security by Design". Саме тому ефективна стратегія управління повинна інтегрувати ці методології в багаторівневу структуру. Зокрема, NIST SSDF має бути використаний як політика верхнього рівня для управління та забезпечення необхідної відповідності державним і міжнародним стандартам. OWASP SAMM, завдяки своїй моделі зрілості, є ідеальним інструментом для об'єктивного вимірювання поточного стану та прогресу безпеки. Для технічної реалізації конкретних захисних механізмів слід застосовувати адаптовані практики з Microsoft SDL. Такий синтетичний підхід забезпечує не лише надійність ПЗ, але й створює масштабовану та гнучку систему постійного вдосконалення безпеки.

Список використаних джерел

1. Stecklein J. et al. Error Cost Escalation Through the Project Life Cycle. NASA Technical Report, 2010. URL: <https://ntrs.nasa.gov>
2. The Economic Impacts of Inadequate Software Testing Infrastructure. NIST Planning Report 02-3, 2002. URL: <https://www.nist.gov>
3. Anwar M. et al. Measuring the Cost of Software Vulnerabilities. Semantic Scholar, 2020.
4. Secure by Design: Building Security into Technology Products. CISA, Cybersecurity and Infrastructure Security Agency, 2023.
5. Saltzer J. H., Schroeder M. D. The Protection of Information in Computer Systems. Proceedings of the IEEE, 63(9), 1975.
6. Security Development Lifecycle. Microsoft, 2024. URL: <https://www.microsoft.com/en-us/securityengineering/sdl>
7. Software Assurance Maturity Model (SAMM). OWASP Foundation. URL: <https://owasp.samm.org/>
8. Secure Software Development Framework (SSDF) Version 1.1: Recommendations for Mitigating the Risk of Software Vulnerabilities. NIST SP 800-218, 2022.
9. Security Champions Playbook. OWASP Foundation. URL: <https://owasp.org/www-project-security-culture/>

ІНТЕЛЕКТУАЛЬНА СИСТЕМА ДИНАМІЧНОГО ЦІНОУТВОРЕННЯ З ВИКОРИСТАННЯМ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Тимчишин В.С.¹⁾, Плавуцький С.М.²⁾
Західноукраїнський національний університет
¹⁾доктор філософії; ²⁾магістрант

I. Постановка проблеми

Швидка цифровізація ринку e-commerce та зростання обсягів доступних даних роблять статичне ціноутворення неефективним. Платформи на кшталт Amazon, Walmart, Uber активно використовують динамічне коригування цін, що суттєво впливає на конкуренцію. За результатами дослідження Chen, Mislove та Wilson, понад 30% проаналізованих продавців Amazon застосовують алгоритмічне ціноутворення, що свідчить про масштабність цього підходу у сучасній комерції [1]. Водночас формування цін залежить не лише від попиту та поведінки користувачів, а й від стану запасів, ризику їх вичерпання та реакції покупців на зміну цін. Це створює потребу в інтелектуальних системах, які можуть поєднувати прогнозування, сегментацію клієнтів, оцінку ризиків і оптимізацію. Тому постає завдання побудувати комплексну систему динамічного ціноутворення з використанням сучасних методів машинного навчання.

II. Мета роботи

Метою цієї роботи є розроблення моделі інтелектуальної системи динамічного ціноутворення для e-commerce, яка використовує машинне навчання для:

- 1) Прогнозування ймовірності покупки користувачем;
- 2) Сегментації клієнтів за поведінковими ознаками;
- 3) Адаптивного визначення оптимального цінового діапазону для кожного сегменту;
- 4) Врахування наявних запасів, попиту та конкурентних цін на схожі товари при прийнятті цінових рішень.
- 5) Автоматизацію формування щоденних цінових рекомендацій.

Досягнення цієї мети можливе через поєднання аналітичних методів кластеризації, регресійних моделей та інструментів оптимізації.

III. Основна частина Огляд існуючих підходів

Сучасні дослідження післяпропонують використання поєднання кластеризації та моделей прогнозування поведінки покупця (Sarkaretal., 2023), багатофакторного аналізу попиту з урахуванням дефіциту товарів (Ferreira, Simchi-Levi, 2020+), а також алгоритмів динамічного оновлення цін у масштабних маркетплейсах (Chenetal., перевидання даних 2021р.). Ці роботи демонструють ефективність ML-підходів, що комбінують попит, поведінку споживачів та інвентар.

Методи, які планується застосувати на практиці

Перелік запланованих методів для застосування:

- 1) K-means або GaussianMixtureModels – для сегментації користувачів за поведінкою, інтенсивністю покупок, ціною чутливістю.
- 2) Логістична регресія, RandomForest або XGBoost– для прогнозування ймовірності покупки при різних цінових рівнях.
- 3) Моделювання втрат продажів (LostSalesEstimation) – для корекції даних попиту, якщо товари були недоступні через низькі запаси (практика, застосована в сучасних моделях Ferreira&Simchi-Levi).
- 4) Багатопродуктова оптимізація – застосування математичного програмування для визначення оптимальної ціни з урахуванням конкурентів, запасів та цінових обмежень.
- 5) Модуль автоматичного оновлення цін – формування щоденних або внутрішньодобових цінових рекомендацій у вигляді decision-support системи.

Концептуальна модель системи

На основі вищенаведених підходів пропонується наступна архітектура інтелектуальної системи динамічного ціноутворення зображена на рисунку 1 та опис її елементів в таблиці 1.

Елементи архітектури інтелектуальної системи динамічного ціноутворення

Елемент архітектури	Опис
Модуль збору даних	Агрегація інформації про поведінку користувачів (візити, кліки), історію покупок, характеристики товарів, запаси.
Кластеризація клієнтів	Застосування алгоритму K-means для групування клієнтів на сегменти із подібними патернами поведінки.
Прогнозування ймовірності покупки	Побудова моделей логістичної регресії або інших ML-моделей для кожного сегменту, що прогнозують рішення “купити/ не купити” в межах цінового діапазону.
Оцінка втрат через нестачу запасів	Проводити оцінку втрачених продажів і включати це у модель прогнозування попиту, щоб не занижувати справжній попит.
Оптимізаційний блок ціноутворення	Багатопродуктова оптимізація цін з урахуванням подібних “reference price” ефектів та обмежень на запаси.
Система підтримки прийняття рішень	Побудова інтерфейсу для ритейлера, який щодня або з іншою періодичністю генеруватиме рекомендовані ціни на основі моделі.



Очікувані переваги та внесок

Запропонована модель забезпечує три ключові переваги. По-перше, вона дозволяє персоналізувати ціни для різних сегментів покупців, підвищуючи відгук і доходність. По-друге, включення оцінки втраченої купівельної активності дає змогу точніше оцінювати реальний попит, що запобігає неправильному ціноутворенню в умовах низьких запасів. По-третє, система оптимізації дозволяє враховувати обмеження запасів, конкуренцію та цінові пороги, що робить модель придатною до використання в реальних бізнес-процесах.

Висновок

Система динамічного ціноутворення на основі методів машинного навчання та оптимізації є актуальним інструментом для e-commerce. Аналіз сучасних підходів показує, що поєднання сегментації, прогнозування попиту та багатопродуктової оптимізації дозволяє покращити прибутковість та адаптивність. Запропонована архітектура може бути основою для створення програмного прототипу, який буде протестовано в симуляціях та на реальних даних торгових платформ.

Рисунок 1 – Архітектура інтелектуальної системи динамічного ціноутворення

Список використаних джерел

- [1] L. Chen, A. Mislove, and C. Wilson, “Algorithmic Pricing on Amazon Marketplace: Updated Empirical Evidence,” *Northeastern University Technical Report*, 2021.
- [2] M. Sarkar, E. H. Ayon, M. T. Mia, R. K. Ray, M. S. Chowdhury, B. P. Ghosh, M. T. Islam, and M. Tayaba, “Optimizing E-Commerce Profits: A Comprehensive Machine Learning Framework for Dynamic Pricing and Predicting Online Purchases,” *JCSST*, vol. 5, no. 4, pp. 186–193, 2023.
- [3] K. Ferreira and D. Simchi-Levi, “Demand Forecasting and Price Optimization in Online Retail: Updated Methods and Applications,” *MIT Working Papers*, 2020.

СИСТЕМА ОПТИМІЗАЦІЇ ХМАРНИХ ВИТРАТ НА ОСНОВІ АНАЛІЗУ МЕТРИК СПОЖИВАННЯ

Марценюк Є.О.¹⁾, Бойко А.А.²⁾

Західноукраїнський національний університет

¹⁾ к.т.н., доцент; ²⁾ магістр

I. Постановка проблеми

Швидке зростання використання хмарних сервісів (AWS, Azure, Google Cloud) у компаніях призводить до збільшення непрямих витрат на інфраструктуру. Типовими проблемами є надлишкові ресурси, некоректні конфігурації, дублювання сервісів, постійна робота непотрібних інстансів та відсутність регулярного моніторингу. За даними Flexera 2024, підприємства в середньому переплачують понад 30% свого хмарного бюджету через неефективне використання ресурсів. Тому актуальною є розробка програмної системи, що автоматично аналізує хмарні метрики, визначає аномалії споживання та пропонує оптимізаційні рекомендації.

II. Мета роботи

Метою роботи є створення програмної системи оптимізації хмарних витрат, яка на основі метрик споживання обчислювальних ресурсів (CPU, RAM, I/O, network) формує рекомендації щодо зменшення витрат, автоматично виявляє неефективні сервіси та будує прогноз майбутнього споживання.

III. Основна частина

У межах роботи було розроблено прототип системи на основі платформи .NET 8.0 з використанням С#. Для збору метрик застосовано механізм інтеграції з API провайдерів (AWS CloudWatch, Azure Monitor). Метрики зберігаються у базі даних PostgreSQL у нормалізованій структурі (таблиці metrics, services, recommendations).

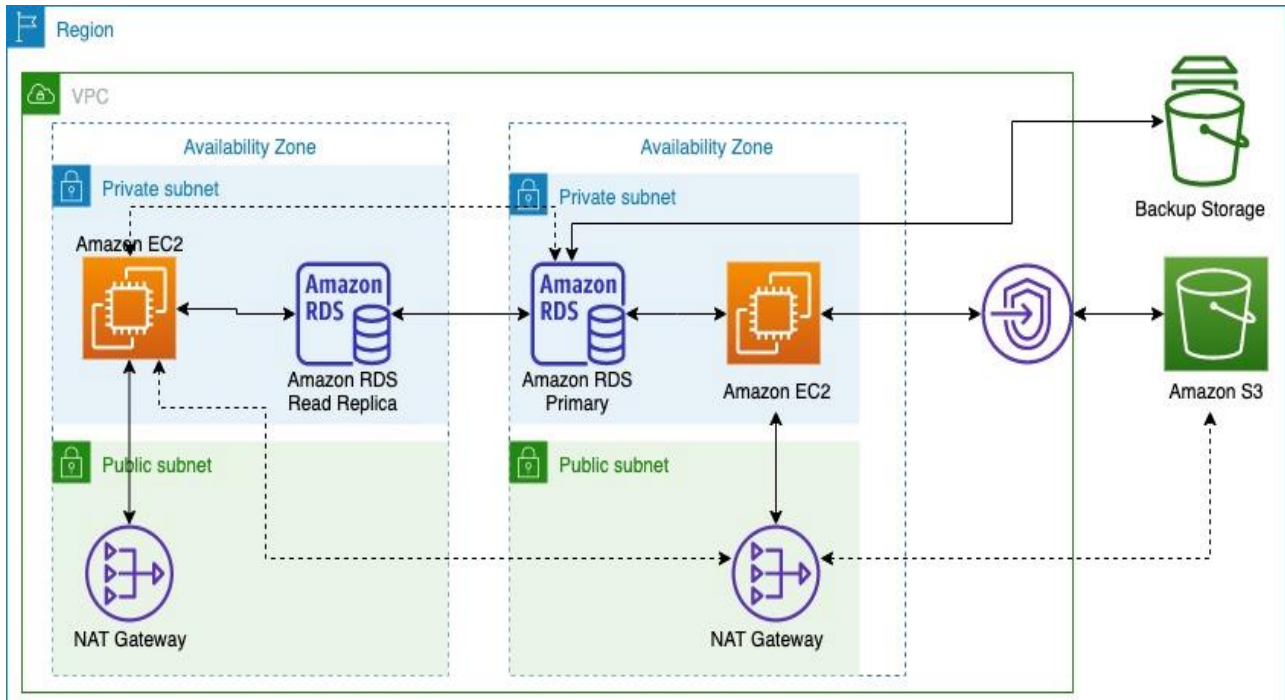


Рисунок 1 – Узагальнена архітектура системи оптимізації хмарних витрат

Архітектура містить такі основні модулі:

- MetricCollector — імпорт метрик споживання ресурсів через API провайдерів;
- Analyzer — обчислення коефіцієнта використання, виявлення аномалій;
- RecommendationEngine — генерація порад (видалити ресурс, зменшити інстанс, змінити тариф);

- Predictor — прогнозування майбутнього навантаження методом ковзного середнього;
 - ReportGenerator — формування PDF/HTML-звітів для користувача.
- Основою аналізу є формула коефіцієнта завантаження ресурсу:

$$K_{util} = \frac{U_{avg}}{U_{alloc}} \times 100\% \quad (1)$$

де U_{avg} – середнє фактичне споживання,
 U_{alloc} – виділені ресурси.

Якщо $K_{util} < 40\%$, система позначає сервіс як over-provisioned.

Таблиця 1

Приклад оцінки ефективності використання ресурсів

Сервіс	CPU avg	CPU alloc	Kutil	Рекомендація
VM-1	12%	4 vCPU	30%	Зменшити до 2 vCPU
VM-2	68%	2 vCPU	68%	Без змін
DB-1	20%	8 GB RAM	25%	Зменшити RAM до 4 GB
API-S	5%	1 vCPU	5%	Видалити або зупинити

Функціональні можливості системи:

- автоматичний щоденний збір метрик;
- побудова дашбордів з тенденціями споживання;
- визначення малоактивних інстансів;
- прогнозування витрат за місяць;
- формування готових рекомендацій у структурованому вигляді.

Під час тестування система показала скорочення витрат до 18% у тестовому середовищі з трьома сервісами та 12 віртуальними машинами. Точність прогнозу навантаження склала 67–74%.

Додатково було проведено аналіз стабільності навантаження, що дозволило визначити сервіси з нерівномірним профілем роботи. Для таких випадків система автоматично пропонує використання механізмів autoscaling, що забезпечує баланс між продуктивністю та вартістю. Також у прототип інтегровано модуль виявлення неактивних ресурсів, який відстежує інстанси, що не використовувалися протягом визначеного періоду часу, та позначає їх як кандидати на вимкнення.

У рамках дослідної експлуатації також було протестовано інтеграцію з Terraform-конфігураціями, що дало змогу автоматично формувати пропозиції щодо зміни параметрів інфраструктури та надсилати їх у вигляді pull-request у репозиторій. Такий підхід дозволяє не лише виявляти неефективні ресурси, а й спрощує процес внесення змін у хмарне середовище, підвищуючи загальну ефективність DevOps-процесів.

Висновок

У результаті виконаної роботи створено прототип системи оптимізації хмарних витрат, здатної автоматично аналізувати метрики, визначати неефективні сервіси та формувати рекомендації. Архітектура проєкту є модульною та масштабованою, що дозволяє розширити підтримку платформ (DigitalOcean, Oracle Cloud) або інтегрувати підтримку autoscaling-алгоритмів.

Список використаних джерел

1. WhatisAWSCloudWatch? [Електронний ресурс] – Режим доступу: <https://docs.aws.amazon.com/cloudwatch>
2. AzureMonitorDocumentation. [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/azure/azure-monitor>
3. Flexera. 2024 State of the Cloud Report. [Електронний ресурс] – Режим доступу: <https://info.flexera.com>
4. Бончук Л. І., Главацький І. В. Інженерія програмного забезпечення: навч. посіб. – Львів: Львівська політехніка, 2020. – 356 с.
5. Чекмарьов С. Г. Тестування програмного забезпечення: навчальний посібник. – Харків: ХНУРЕ, 2018. – 214 с.
6. Spivak I.Ya., Krepuch S.Ya., Horishniy V.I. “Organization of CLOUD architecture for systems ensuring the functional suitability of static systems”. Scientific journal “Modern Information Systems”, Kharkiv, Volume 3, No.2, 2019. – pp.35-39
6. Гамма Е., Хелм Р., Джонсон Р., Вліссідес Дж. Прийоми об’єктно-орієнтованого проєктування. Патерни проєктування. – Київ: Діалектика, 2021. – 416 с.
7. NET 8.0 Overview. [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/dotnet/core/whats-new/dotnet-8>

АРХІТЕКТУРА ВЕБ-СИСТЕМИ УПРАВЛІННЯ СТУДЕНТСЬКИМИ НАУКОВИМИ ПРОЄКТАМИ

Матвійчук Ю.В.¹⁾, Тимчишин В.С.²⁾
Західноукраїнський національний університет
^{1)бакалавр;^{2)доктор філософії}}

I. Вступ

Ефективна організація студентської наукової діяльності є важливою складовою освітнього процесу закладів вищої освіти. Зростання кількості студентських досліджень, необхідність систематизації результатів, оптимізація взаємодії між студентами, науковими керівниками та адміністративним персоналом вимагають застосування сучасних інформаційних технологій. Веб-системи управління студентськими науковими проєктами дозволяють автоматизувати ключові процеси — від вибору теми та подання матеріалів до рецензування та оцінювання — забезпечуючи прозорість, оперативність та контроль якості на всіх етапах роботи. Розробка архітектури такої системи є актуальним завданням, що потребує урахування вимог масштабованості, безпеки та інтеграції з університетською інфраструктурою.

II. Постановка проблеми

У більшості закладів освіти управління студентськими науковими проєктами здійснюється із використанням розрізаних інструментів: електронної пошти, документів у файлових сховищах, месенджерів або навіть паперових форм. Такий підхід призводить до втрати даних, дублювання інформації, складності контролю дедлайнів та неузгодженості між учасниками процесу. Наукові керівники стикаються з труднощами в організації рецензування, а студенти — у відстеженні статусу своїх робіт. Відсутність єдиної платформи ускладнює зберігання архіву проєктів, аналітику та підготовку звітної документації. Тому виникає потреба у створенні централізованої веб-системи, яка підтримує структуроване управління студентськими науковими проєктами та дозволяє автоматизувати основні етапи взаємодії між усіма учасниками освітнього процесу.

III. Мета роботи

Метою роботи є проектування архітектури веб-системи управління студентськими науковими проєктами, яка забезпечує ефективну взаємодію між студентами, науковими керівниками та адміністрацією, підтримує повний життєвий цикл проєкту, гарантує цілісність і безпеку даних, а також створює умови для подальшого розвитку інформаційної системи.

IV. Архітектура веб-системи

На рисунку 1 наведено, розроблену в межах цієї праці, архітектуру управління студентськими науковими проєктами. Як видно з рисунку, запропонована архітектура реалізована на основі трирівневої клієнт-серверної моделі (Three-Tier Architecture), яка складається з Presentation Layer, Application Layer та Data Layer. Такий підхід забезпечує чітке розмежування відповідальностей, полегшує супровід, дає можливість паралельної розробки модулів та збільшує стійкість системи до навантажень[1].

Presentation Layer представлений веб-клієнтом, який працює у браузері та побудований за принципами SPA (Single Page Application). Це забезпечує швидкий відгук інтерфейсу, мінімальну кількість повних перезавантажень сторінки та зручну роботу користувача. Взаємодія з сервером реалізована через HTTPS-запити у форматі JSON. Основними користувачами Presentation Layer є студенти, наукові керівники, та адміністратори. [2].

Application Layer виконує роль центрального логічного ядра і містить REST API, який приймає всі HTTP-запити від клієнта. REST API забезпечує маршрутизацію викликів, перевірку авторизації, обробку даних та формування відповідей. Усі запити надходять саме через цей компонент, що дозволяє легко контролювати потоки даних та безпеку. У межах Application Layer окремою складовою виступає Auth & Security Module, що відповідає за автентифікацію користувачів, видачу токенів доступу та контроль рівнів дозволів. Наявність централізованого модуля безпеки гарантує збереження персональних даних та регламентує можливості користувачів залежно від ролі[3]. Projects Module реалізує головну функціональність — керування життєвим циклом студентських наукових проєктів. Він виконує операції зі створення, редагування, зміни статусів, прикріплення наукових

керівників та відстеження прогресу. Це центральний модуль системи, який взаємодіє з усіма іншими. Users & Roles Module забезпечує підтримку багаторольової моделі доступу. Він зберігає структуру ролей, профілі користувачів та їх прив'язку до груп, кафедр чи освітніх програм. Завдяки цьому модулю система адаптується до організаційної структури університету. Для підтримки академічної якості важливим є Reviews Module. Він забезпечує можливість рецензування робіт, додавання коментарів, зауважень та формування оцінок. Система фіксує всі етапи рецензування, що дозволяє відтворювати історію змін і підвищує прозорість процесу.

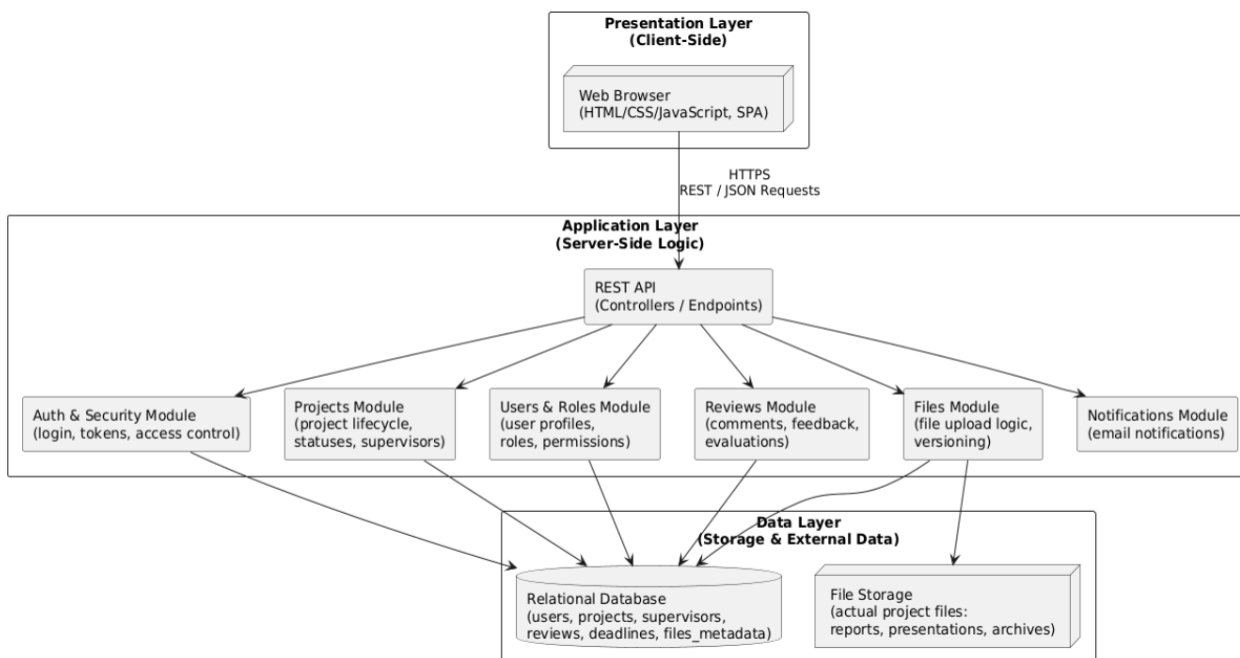


Рисунок 1 – Модель архітектури веб-системи управління студентськими науковими проектами

Files Module відповідає за зберігання документів (звітів, презентацій, додаткових матеріалів). Він включає логіку завантаження, перевірки доступу, зберігання метаданих, версіонування та передачі файлів у файлове сховище. Паралельно Notifications Module надсилає системні сповіщення про зміни статусів, нові рецензії або наближення дедлайнів, забезпечуючи оперативність комунікації.

Data Layer включає реляційну базу даних, де зберігаються структуровані таблиці користувачів, проєктів, рецензій, дедлайнів та метаданих файлів, а також файлове сховище для реальних документів. Такий розподіл забезпечує оптимальне використання ресурсів і високу ефективність обробки даних. Архітектура підтримує можливість інтеграції з університетськими ІС, що робить систему гнучкою, масштабованою та придатною для довгострокової експлуатації.

Висновок

Запропонована архітектура веб-системи управління студентськими науковими проектами вирішує основні проблеми організації студентської дослідницької діяльності, забезпечуючи централізований доступ, структурованість даних, контроль виконання етапів і підтримку рецензування. Трирівнева клієнт–серверна модель дозволяє розмежувати відповідальність між компонентами системи, що підвищує її гнучкість, масштабованість та безпечність. Впровадження такої системи сприятиме підвищенню якості наукових робіт студентів, оптимізації роботи наукових керівників і спрощенню адміністративних процедур в закладі вищої освіти.

Список використаних джерел

1. V. Tymchyshyn, A. Melnyk, B. Tymchyshyn, V. Faifura, I. -S. Mazur and L. Honchar, "Data Management Service Architecture of the Software for Modeling Harmful Emissions in Soil," 2024 14th International Conference on Advanced Computer Information Technologies (ACIT), Ceske Budejovice, Czech Republic, 2024, pp. 678-682.
2. N. Sharma and R. Chowdhary, "Design and Implementation of Three-Tier Architecture in Web Application Development", International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, №12, pp. 505–511, 2020.
3. Taibi D., Lenarduzzi V. Architectural Patterns for Microservices: A Systematic Mapping Study. ACM Digital Library, 2021.

ПРОГРАМНА РЕАЛІЗАЦІЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ РОЗПОДІЛУ РЕСУРСІВ З УРАХУВАННЯМ ПРОФЕСІЙНОГО ВИГОРАННЯ

Осадчук С.Б.¹⁾, Шпінталь М.Я.²⁾
Західноукраїнський національний університет
¹⁾ магістрант; ²⁾ к.т.н., доцент

I. Постановка проблеми

У попередніх дослідженнях авторами було розроблено математичну модель оптимального розподілу ресурсів з урахуванням професійного вигорання [1,2]. Дана модель потребує практичної реалізації та експериментального підтвердження ефективності через створення програмного комплексу з можливістю порівняльного аналізу існуючих методів.

II. Мета роботи

Метою роботи є розробка програмного комплексу для реалізації запропонованої математичної моделі оптимізації розподілу ресурсів, а також проведення попередніх експериментальних досліджень на синтетичних даних з метою аналізу поведінки моделі та оцінки потенційних переваг підходу з урахуванням професійного вигорання порівняно з традиційними методами.

III. Архітектура та реалізація програмного комплексу

Система реалізована на базі Python 3.10+ з використанням бібліотеки PuLP для моделювання задач цілочисельного лінійного програмування та solver'a CBC для їх розв'язання. Архітектура побудована за принципом конвеєрної обробки, де вхідні дані послідовно проходять етапи завантаження, препроцесингу параметрів моделі (розрахунок індексів вигорання B_i та матриці компетенцій M_{ij}), оптимізаційного ядра на базі ILP, аналізу результатів та формування звітів у форматах CSV і JSON для інтеграції з системами управління проектами (див.рис. 1).

Запропонована архітектура орієнтована на подальше розширення та дозволяє проводити серії обчислювальних експериментів з різними параметрами моделі без зміни її базової структури.

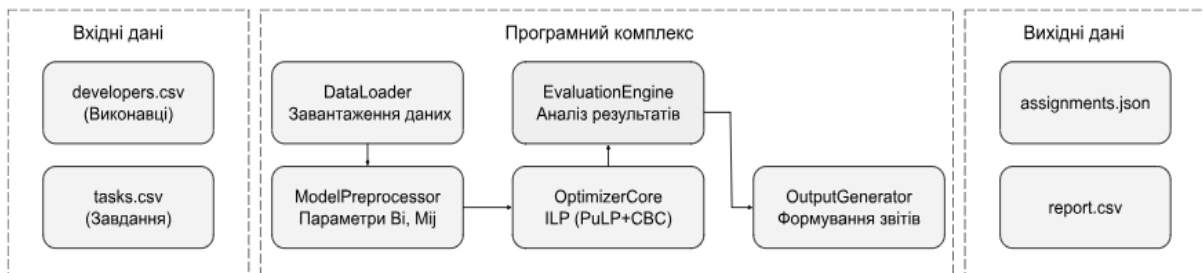


Рисунок 1 - Архітектура системи оптимізації розподілу ресурсів

Програмна реалізація базується на модифікованій цільовій функції з динамічним урахуванням професійного вигорання, де виконавці з підвищеним рівнем втоми отримують менше навантаження через ваговий коефіцієнт ω :

$$\min Z = \sum_{i=1}^m \sum_{j=1}^n [(1 + \omega \cdot B_i) \cdot T_j - M_{ij}] \cdot x_{ij} \quad (1)$$

де B_i – індекс професійного вигорання виконавця i ; T_j – трудомісткість завдання j ; M_{ij} – бінарна матриця відповідності компетенцій; x_{ij} – змінна призначення. Система обмежень включає часові ресурси виконавців, компетентнісні вимоги та унікальність призначення кожного завдання.

IV. Експериментальні дослідження

Експериментальні дослідження проведено на синтетичних даних, згенерованих відповідно до параметрів математичної моделі, з метою аналізу тенденцій та поведінки алгоритму при зміні

розмірності задачі та вагових коефіцієнтів. Отримані числові оцінки мають попередній характер і слугують для ілюстрації потенційних переваг запропонованого підходу.

Для оцінки ефективності проведено три серії експериментів на синтетичних даних.

Експеримент 1: Масштабованість алгоритму. Досліджено продуктивність на конфігураціях від 5×10 до 30×70 (виконавців \times завдань). Результати демонструють зростання часу виконання зі збільшенням розмірності задачі, що залишається прийнятним для офлайн-планування та аналітичних сценаріїв. Результати експериментів свідчать про здатність запропонованого підходу знаходити допустимі оптимізаційні рішення для задач розмірністю від 5×10 до 30×70 (виконавців \times завдань) у прийнятний для практичного використання час.

Таблиця 1

Порівняльна ефективність методів розподілу ресурсів (конфігурація 10×20)

Метод	Час (с)	CV	Макс. навантаження (%)	B_i сер.
Random Assignment	< 0.1	0.40	90	0.50
First Fit	0.05	0.33	85	0.47
ILP (без вигорання)	2.0	0.30	82	0.50
ILP + Burnout ($\omega=0.3$)	3.5	0.26-0.27	78	0.44

де CV – коефіцієнт варіації навантаження; B_i сер. – середній індекс вигорання.

Експеримент 2: Порівняння з baseline методами. Порівняльний аналіз на конфігурації 10×20 демонструє тенденцію до більш рівномірного розподілу навантаження при використанні запропонованої моделі. Зокрема, коефіцієнт варіації навантаження зменшується приблизно на 10–13% порівняно з класичною ILP-моделлю, а середній індекс професійного вигорання — на 10–15%. Максимальне навантаження знижується на 4–7 відсоткових пунктів, що потенційно зменшує ризик перевантаження окремих виконавців.

Експеримент 3: Дослідження впливу вагового коефіцієнта ω показало, що при $\omega = 0$ модель вироджується в класичну ILP без урахування вигорання. У діапазоні $\omega \approx 0.2$ – 0.4 спостерігається найбільш збалансоване співвідношення між рівномірністю розподілу навантаження та обчислювальною складністю. Подальше збільшення ω призводить до покращення показників вигорання, однак супроводжується зростанням часу обчислень, що може обмежувати практичну застосовність підходу.

Висновки

У роботі представлено програмний комплекс для реалізації математичної моделі оптимізації розподілу ресурсів у командах розробки з урахуванням професійного вигорання. Проведені попередні експериментальні дослідження на синтетичних даних свідчать про потенційну ефективність запропонованого підходу з точки зору масштабованості, більш рівномірного розподілу навантаження та зменшення ризику професійного вигорання. Отримані результати мають попередній характер і підтверджують доцільність подальших детальніших досліджень, зокрема на реальних проектних даних, що планується виконати в рамках магістерської роботи.

Список використаних джерел

- Осадчук С.Б., Шпінталь М.Я. Математичні моделі оптимального розподілу ресурсів розробки продуктів. // Матеріали зимової школи-семінару молодих учених і студентів «Комп'ютерні інформаційні технології – СІТ 2024», Тернопіль: ЗУНУ, 2024. – С. 1–3.
- Осадчук С.Б., Шпінталь М.Я. Математична модель розподілу ресурсів у розробці програмного забезпечення з урахуванням професійного вигорання. // Матеріали школи-семінару молодих учених і студентів «Комп'ютерні інформаційні технології – СІТ 2025», Тернопіль: ЗУНУ, 2025. – С. 1–3.
- Yang S., Fu C. Critical chain scheduling in resource-constrained project management. // IEEE Transactions on Engineering Management. — 2022. — Vol. 69, No. 3, pp. 858-871
- Співак І.Я., Крепич С.Я., Федоров О.А. Програмна система оцінювання ефективності праці в залежності від потреб// Матеріали школи-семінару молодих вчених і студентів «Комп'ютерні інформаційні технології» СІТ'2019, 29 листопада 2019р., Тернопіль, стр. 34
- Mitchell S., O'Sullivan M., Dunning I. PuLP: A Linear Programming Toolkit for Python. — 2011. — optimization-online.org
- Крепич С.Я. Моделювання та забезпечення функціональної придатності статичних систем методами аналізу інтервальних даних/ С.Я. Крепич//дис.канд.тен.наук, НУ «Львівська політехніка», Львів, 2016. – 166с.
- Pinedo M. Scheduling: Theory, Algorithms, and Systems. — Springer, 2016.

МАТЕМАТИЧНА МОДЕЛЬ ОПТИМАЛЬНОГО РОЗМІЩЕННЯ ТОЧОК ЗВ'ЯЗКУ У ПРИМІЩЕННЯХ

Марценюк Є.О.¹⁾, Корман М.К.²⁾
Західноукраїнський національний університет
¹⁾ к.т.н., доцент; ²⁾ магістрант

I. Постановка проблеми

З ростом числа користувачів і пристроїв забезпечення надійного Wi-Fi-покриття в приміщеннях стає критичним завданням. Сучасні додатки (HD-відео, VoIP, IoT) висувають підвищені вимоги до пропускної здатності та стабільності мережі, що робить оптимальне розміщення ТП надзвичайно складним технічним завданням. З огляду на це виникає потреба в математичній моделі та автоматизованому алгоритмі, що враховує реалії середовища (геометрію приміщень, матеріали стін, шум) для планування Wi-Fi-мережі з гарантованим охопленням і заданим рівнем якості зв'язку.

II. Мета роботи

Метою є розробка та обґрунтування математичної моделі оптимального розміщення точок доступу Wi-Fi у внутрішніх приміщеннях. Основні завдання моделі: гарантувати заданий рівень покриття всіх важливих зон приміщення та мінімізувати кількість ТП (витрати). Ключовим результатом має стати кількісний показник, що оцінює якість покриття всього простору (наприклад, «коефіцієнт покриття» або сумарна площа з достатнім рівнем RSSI). Такий показник дозволить порівнювати альтернативні варіанти розгортання мережі та приймати рішення на основі об'єктивних даних. Зокрема, ціль моделі - підрахувати оптимальні координати для встановлення точок доступу з урахуванням даних про план приміщення і заданих параметрів мережі.

Отж, ціль моделі – підрахувати оптимальні координати для встановлення точок доступу з урахуванням даних про план приміщення і заданих параметрів мережі.

III. Аналіз недоліків існуючих підходів та інструментарію

Існуючі методи планування Wi-Fi мереж можна умовно розділити на кілька груп: ручні обстеження, інженерні CAD-інструменти, вузькоспеціалізовані симулятори та евристичні алгоритми.

Професійні CAD-інструменти на кшталт Ekahau, AirMagnet тощо використовують детальні плани будівель і моделюють покриття - вони показують теплові карти сигналу і можуть врахувати типи будівельних матеріалів. Проте ці системи здебільшого комерційні, дорогі та не завжди інтегруються в автоматизовані процеси.

Симулятори мереж (NS-3, OMNeT++ та ін.) - відкриті інструменти для детальної перевірки протоколів рівнів PHY/MAC. Результатом їхньої роботи є суто статистичні дані та графіки, а не готовий план розміщення ТП.

Недоліки існуючих рішень у порівнянні з нашою ідеєю наведені в таблиці 1:

Таблиця 1

Порівняльна характеристика методів планування Wi-Fi мереж

Критерій	Запропонована система	Ручне обстеження	Інженерні CAD інструменти (Ekahau/AirMagnet)	Симулятори (NS-3/OMNeT++)	Евристичні алгоритми
Методологія	Кількісна	Якісна	Комбінована	Детальна	Кількісна
Автоматизація	Повна	Відсутня	Часткова	Повна	Варіюються
Модель середовища	Розширена	Немає	Детальна	Глибока	Обмежена
Результат	Оптимальні координати ТП+	Суб'єктивні висновки	Карти покриття та звіти	Статистичні дані для симуляції	Рекомендовані конфігурації ТП

IV. Програмна реалізація

Для реалізації мого проекту, на мою думку найкращим варіантом є мова програмування C# на платформі .NET. NET-середовище забезпечує потужні можливості для роботи з мережевим обладнанням, а також має бібліотеки для чисельної оптимізації (Math.NET, Accord.NET) і створення графічного інтерфейсу (WPF, WinForms). Також дана мова програмування відповідає всім

парадигмам програмування та широко використовується в різноманітних корпоративних системах та програмах і має високу ступінь сумісності з веб технологіями через браузерні інтерфейси що створбе хороші умови для її використання.

Основний обчислювальний модуль реалізує радіотехнічну модель розповсюдження сигналу. При обчисленні покриття враховуються відстань до ТП, втрати у перешкодах та потенційна інтерференція з іншими АР. Алгоритм оптимізації може використовувати, генетичний підхід або локальний пошук із випадковими генерованими варіантами розташування. Результат - набір оптимальних координат ТП - виводиться разом з числовими метриками покриття.

Для візуалізації та звітності в середовищі C# зручно застосувати бібліотеку System.Drawing або WPF: які дозволяють накладати рамки або контури покриття на план приміщення. Таким чином користувач отримає ідентифіковані «сліпі зони» покриття. Будь-які етапи аналізу можна логувати та передавати у звіт.

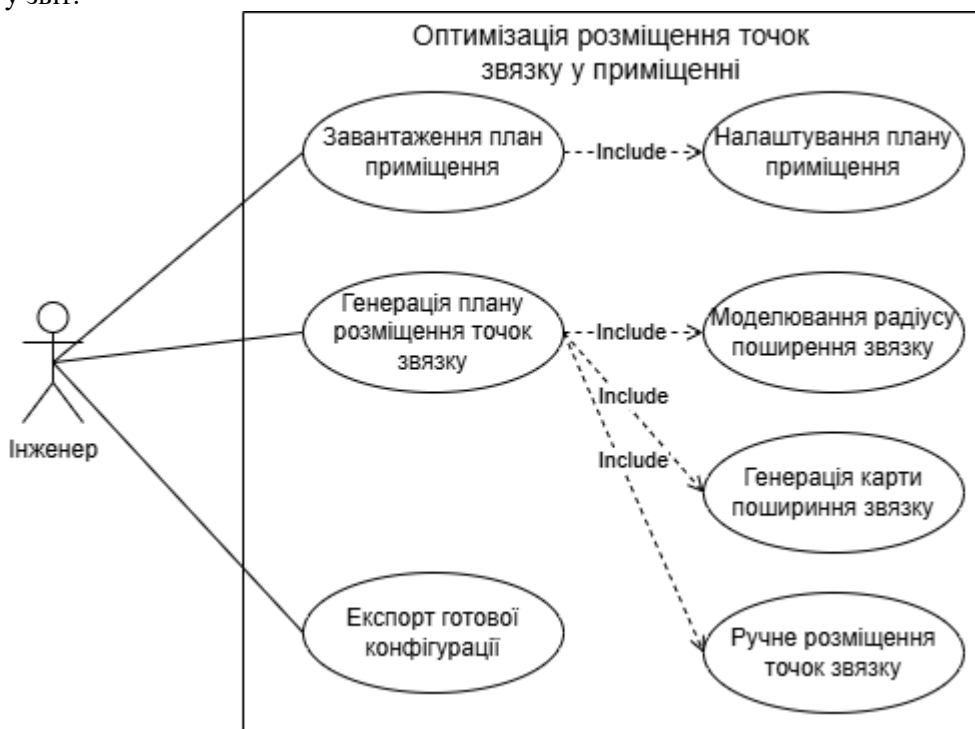


Рисунок 1 - Use Case діаграма

Висновок

Отже проведений аналіз підтверджує, що існують певні неточності в традиційному проектуванні Wi-Fi мереж: ручні перевірки та CAD-інструменти дають лише візуальні оцінки, а наукові моделі часто не мають зручного інтерфейсу та метрик. Запропонована система заповнює цю прогалину, поєднуючи точну математичну модель розповсюдження сигналу з автоматизованим алгоритмом оптимізації. Таким чином впровадження моделі оптимального планування Wi-Fi-мереж сприятиме підвищенню надійності зв'язку, зниженню затрат на обладнання та забезпеченню належного рівня сервісу для кінцевих користувачів.

Ключовою перевагою підходу є можливість моделювати складну геометрію приміщень та радіотехнічні особливості, яких не враховують прості симулятори. Як показують дослідження, правильне оптимальне розміщення ТП суттєво мінімізує втрати сигналу та покращує якість зв'язку. Таким чином впровадження моделі оптимального планування Wi-Fi-мереж сприятиме підвищенню надійності зв'язку, зниженню затрат на обладнання та забезпеченню належного рівня сервісу для кінцевих користувачів що збільшить загальну ефективність роботи в будь якому приміщенні через підвищену якість зв'язку та економію обладнання.

Список використаних джерел:

1. Міжнародний союз електрозв'язку (ITU). Рекомендація ITU-R P.1238-9 "Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 300 MHz to 100 GHz" [Електронний ресурс]. - Женева, 2017.
2. Cisco Systems, Inc. Understand Site Survey Guidelines for WLAN Deployment [Електронний ресурс]. - Cisco Systems, 2023
3. Гадиражу В., У Х.-Ч., Буш К. та ін. Novel Sensor/Access-Point Coverage-Area Maximization for Arbitrary Indoor Polygonal Geometries [Електронний ресурс] // IEEE Wireless Communications Letters. - 2021. - Т. 10, № 12. - С. 2767-2777

ПРОГРАМНА СИСТЕМА ДЛЯ ОРГАНІЗАЦІЇ КОМАНДНОЇ РОБОТИ

Крепич С.Я.¹⁾, Гида І.Р.²⁾, Співак І.Я.³⁾
Західноукраїнський національний університет
^{1)к.т.н., доцент;} ^{2) бакалавр,} ^{3) к.т.н., доцент}

I. Постановка проблеми

В умовах динамічного розвитку ІТ-індустрії та переходу багатьох компаній на гібридний або віддалений режим роботи, ефективність командної взаємодії стає критичним фактором успіху проєктів. Основою для успішної реалізації є налагоджені процеси планування, розподілу ролей та комунікації. На сьогодні ринок пропонує широкий спектр інструментів: Trello або Jira для трекінгу задач, Zoom або GoogleMeet для відеоконференцій, Slack або Telegram для оперативного спілкування. Однак, практика показує, що використання кількох розрізаних платформ створює низку суттєвих проблем. Серед них: необхідність постійного перемикання контексту (contextswitching), дублювання інформації в різних системах, втрата важливих деталей комунікації та складність централізованого моніторингу прогресу. Фрагментований підхід ускладнює робочий процес, особливо у випадках, коли команді потрібно одночасно проводити наради та працювати з дошкою завдань. Тому актуальним завданням є розробка єдиного середовища, яке б інтегрувало управління завданнями, комунікацію в реальному часі та інтелектуальну підтримку користувачів.

II. Мета роботи

Метою роботи є підвищення ефективності командної роботи шляхом розробки веб-застосунку «TaskTracker», що поєднує функціонал Kanban-дошок, відеоконференцій та інтелектуального асистента в єдиному інтерфейсі. Застосунок покликаний мінімізувати час на організаційні витрати та забезпечити прозорість виконання проєктів.

III. Основна частина

Під час аналізу предметної області було досліджено популярні аналоги, такі як Trello, Jira та Asana. Встановлено, що прості інструменти (Trello) часто не мають вбудованих засобів комунікації, а комплексні системи (Jira) є надто складними та дорогими для малих команд або стартапів. Запропонована система «TaskTracker» займає нішу універсального робочого простору, який дозволяє проводити відеодзвінки безпосередньо в контексті активної дошки завдань.

Для забезпечення масштабованості, тестованості та надійності системи було обрано клієнт-серверну архітектуру (див.рис.1) з використанням підходу CleanArchitecture для серверної частини.

1.Клієнтська частина (Frontend): Реалізована на базі фреймворку BlazorWebAssembly. Це дозволяє виконувати клієнтську логіку безпосередньо у браузері користувача, використовуючи мову C# замість JavaScript, що забезпечує високу продуктивність та спільне використання моделей даних із сервером.

2.Серверна частина (Backend): Побудована на платформі ASP.NET CoreWebAPI. Архітектура бекенду розділена на шари згідно з принципами «Чистої архітектури»:

- *Domain*: містить основні сутності (User, Board, Card) та бізнес-правила.
- *Application*: відповідає за сценарії використання (UseCases), CQRS (Command Query Responsibility Segregation) та валідацію.
- *Infrastructure*: реалізує роботу із зовнішніми сервісами.
- *Persistence*: забезпечує взаємодію з базою даних.

3.База даних: В якості сховища даних обрано реляційну СУБД MS SQL Server. Структура бази даних спроектована для забезпечення цілісності зв'язків між сутностями (користувачі, дошки, колонки, коментарі).

Система реалізує рольову модель доступу (Owner, Admin, Member), що дозволяє гнучко налаштовувати права учасників у межах конкретного робочого простору. Ключовою особливістю розробки є інтеграція модуля відеозв'язку, що дозволяє команді обговорювати завдання, не покидаючи інтерфейс веб-застосунку. Це вирішує проблему втрати контексту. Додатково перевагою є впровадження ШІ-помічника. Замість вивчення об'ємної документації, користувачі можуть звертатися до чат-бота для отримання підказок щодо навігації по сайту або оптимізації роботи з завданнями.

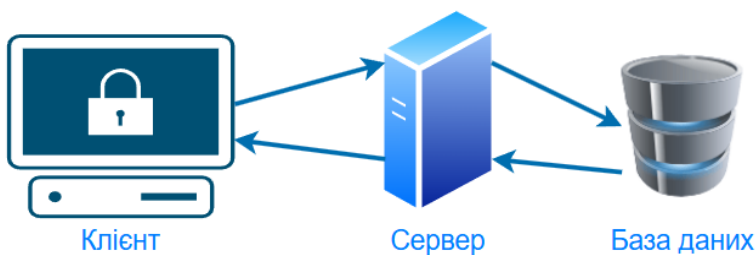


Рисунок 1 –Клієнт-серверна архітектура

На етапі проєктування бази даних було розроблено ER-діаграму (див.рис.2), яка визначає зв'язки «один-до-багатьох» між дошкою та колонками, колонками та картками, а також картками та коментарями. Використання GUID в якості ідентифікаторів дозволяє уникнути колізій при масштабуванні бази даних або злитті даних з різних джерел.

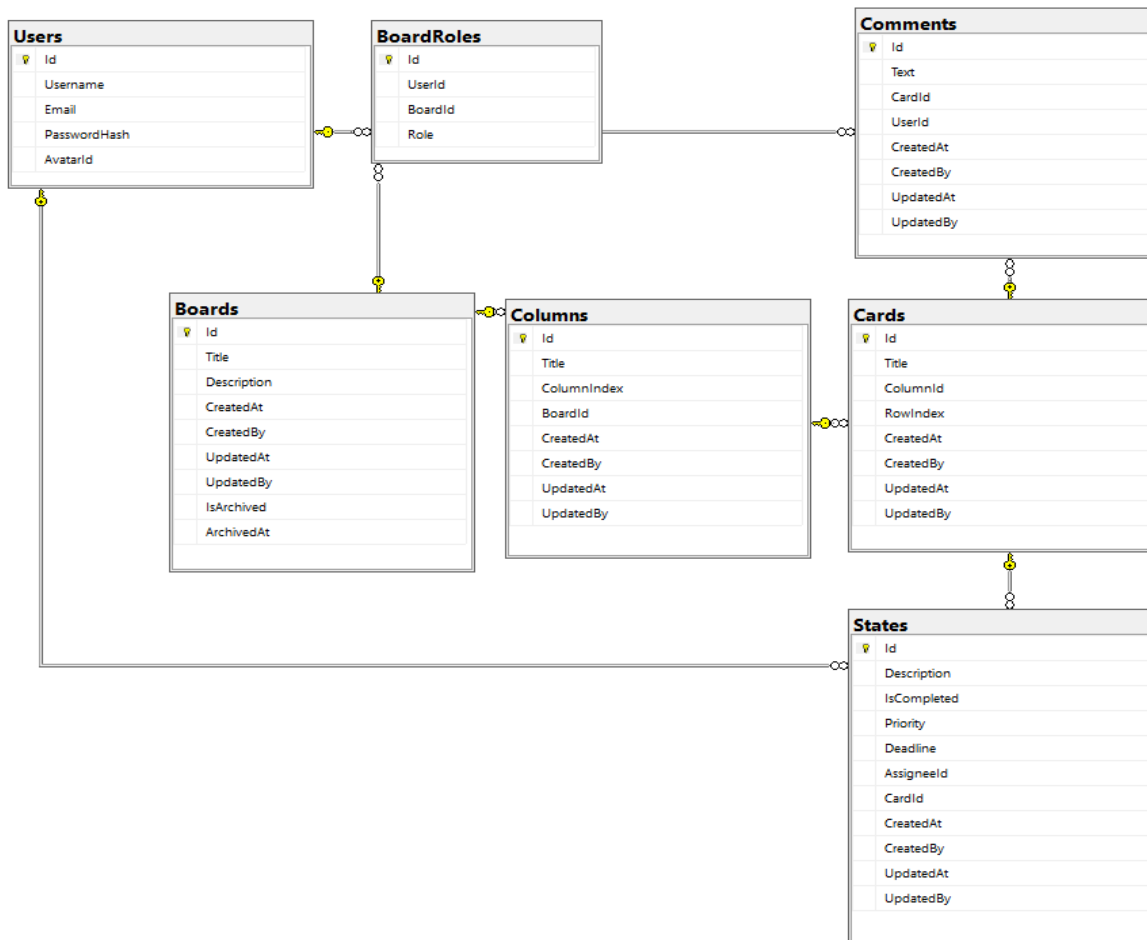


Рисунок 2 – ERD-діаграма

Авторизація користувачів реалізована за допомогою стандарту JWT (JSON WebTokens). Сервер генерує токен після успішної автентифікації, який клієнт використовує для доступу до захищених ресурсів API. Це забезпечує безпеку сесій без необхідності зберігання стану на сервері (stateless).

Висновки

У роботі спроектовано та реалізовано програмну систему «TaskTracker» для організації командної роботи. Використання сучасного стеку технологій .NET (Blazor, ASP.NET Core) та архітектурного патерну CleanArchitecture дозволило створити гнучке, масштабоване рішення.

Список використаних джерел

1. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. PrenticeHall, 2017. 432 p.
2. ASP.NET Core Documentation [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/aspnet/core/>
3. Крепич С.Я. Метод синтезу смугового фільтру для заданих обмежень на його модуль коефіцієнта передачі. Сучасні комп'ютерні інформаційні технології: Матеріали IV Всеукраїнської школи-семінару молодих вчених і студентів АСІТ'2014. – Тернопіль: Економічна думка, 2014. – С.26-29
4. Крепич С.Я., Дивак М.П. Порівняння часової складності реалізації процедур випадкового пошуку в задачі синтезу фільтра та допусків на параметри його елементів. «Інформаційні технології та комп'ютерна інженерія», 2015, Том. 33, №2, стр.47-57
5. Price M. J. C# 10 and .NET 6 – Modern Cross-Platform Development. PacktPublishing, 2021. 824 p.

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ ТЕХНІЧНОГО БОРГУ ПРИ МІГРАЦІЇ LEGACY JAVA-ДОДАТКІВ

Войтюк І.Ф.¹⁾, Косаренко П.К.²⁾

Західноукраїнський національний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

I. Постановка проблеми

Legacy Java-системи накопичують технічний борг – сукупність недосконалостей у кодї та архітектурї, що виникають через використання застарїлих API, відсутність рефакторингу та застосування deprecated конструкцій. Поняття технічного боргу було введено Уордом Каннінґемом у 1992 році як метафора для опису компромїсів у розробцї програмного забезпечення. Технічний борг знижує підтримуваність коду, збільшує ризик виникнення помилок та підвищує вартість супроводу системи.

При мїграцїї на сучасну версїю Java 25 LTS необхідно оцїнити обсяг такого боргу для планування ресурсів. Oracle випускає нову LTS версїю кожні два роки, що створює постійну потребу в мїграцїї корпоративних систем. Кожна нова версїя Java мїстить JEP (JDK Enhancement Proposals), що deprecate або видаляють застарїлі API. Наприклад, Security Manager (JEP 411) позначено як deprecated for removal у Java 17.

Існуючі інструменти статичного аналізу (SonarQube, PMD, SpotBugs, Checkstyle) орієнтовані на поточну підтримку коду та виявлення загальних проблем якості. Вони не враховують специфіки legacy-систем та не пропонують інтегрованих математичних моделей для прогнозування трудомїсткості мїграцїї мїж конкретними версїями Java.

II. Мета роботи

Розробка математичної моделі та програмного забезпечення для автоматизованого виявлення, класифікацїї та кількісної оцїнки технічного боргу в legacy Java-додатках при мїграцїї на Java 25. Система має забезпечувати аналіз вихідного коду, ідентифікацїю deprecated API, розрахунок метрик якості та генерацїю звітів із рекомендацїями щодо послїдовності рефакторингу.

III. Основна частина

Математична модель базується на індексі технічного боргу (TechnicalDebt Index, TDI), який інтегрує кілька метрик якості коду в єдиний нормалізований показник:

$$TDI = \frac{(4 \cdot D + 3 \cdot C + 2 \cdot CC + CD)}{LOC} \times 100 \quad (1)$$

де D – кількість використань deprecated API,

C – критичні проблеми коду,

CC – цикломатична складність за методикою МакКейба,

CD – відсоток дублювання коду,

LOC – загальна кількість рядків коду.

Вагові коефіцієнти (4, 3, 2, 1) визначені експериментально на основі впливу кожного фактора на складність мїграцїї.

Система виконує AST-аналіз (Abstract Syntax Tree) Java-коду за допомогою бібліотеки JavaParser. AST-підхід забезпечує точне розпізнавання синтаксичних конструкцій мови програмування на відміну від текстового пошуку регулярними виразами. Аналізатор будує дерево розбору для кожного Java-файлу та виявляє deprecated API за різними категорїями, наприклад:

– Security: java.security.acl.*, SecurityManager (JEP 411), RMI Activation (видалено в Java 17);

– Concurrency: Thread.stop(), Thread.suspend(), Thread.resume(), Object.finalize();

– Date/Time: конструктори Date(int, int, int), SimpleDateFormat, Calendar (замїна — java.time API);

– Collections: Vector, Hashtable, Stack, Enumeration (замїна — ArrayList, HashMap, Deque, Iterator);

– Boxing: конструктори `newInteger()`, `newBoolean()`, `new Long()` (заміна — статичні методи `valueOf()`).

Для кожного виявленого deprecated API система зберігає метадані: назву API, версію Java де його позначено deprecated, версію планового видалення (якщо визначено), рекомендовану заміну та категорію пріоритету. Ця інформація використовується для генерації детальних рекомендацій щодо рефакторингу коду.

Класифікація проблем здійснюється за чотирма рівнями пріоритету:

1. Critical ($TDI \geq 50$) – негайна міграція необхідна, система містить критичні вразливості безпеки;
2. High ($30 \leq TDI < 50$) – міграція рекомендована в найближчому релізі;
3. Medium ($15 \leq TDI < 30$) – планова міграція протягом року;
4. Low ($TDI < 15$) – міграція за можливості, низький ризик для функціонування системи.

Оцінка трудомісткості міграції обчислюється за формулою:

$$Cost = \frac{LOC}{500} \times (1 + 0.05 \cdot D) \times (1 + \frac{CC}{100}) \quad (2)$$

де результат виражається у людино-днях. Коефіцієнт 500 відповідає середній продуктивності рефакторингу коду досвідченим розробником. Множники враховують додаткові витрати часу на заміну deprecated API та складність навігації у кодї з високою цикломатичною складністю.

Архітектура за стосунку включає два основні компоненти. Консольний аналізатор призначений для автоматизованої пакетної обробки проєктів та інтеграції у CI/CD пайплайни. Реалізовано з використанням records (DeprecationInfo, CodeMetrics) для типобезпечного зберігання структурованих даних аналізу. Records забезпечують immutability та автоматичну генерацію методів equals(), hashCode(), toString().

Графічний інтерфейс розроблено на базі JavaFX 21 з використанням бібліотеки JFreeChart для побудови діаграм. GUI містить: dashboard з візуальним індикатором TDI у вигляді gauge-діаграми, табличне відображення результатів (TableView) з можливістю сортування та фільтрації за будь-яким стовпцем, кругові діаграми розподілу проблем за категоріями та пріоритетами, стовпчикові діаграми порівняння модулів проєкту за значенням TDI.

Модуль експорту генерує звіти у форматах CSV (для подальшої обробки в Microsoft Excel або Google Sheets) та PDF (для документування та архівування результатів аналізу). Структура звіту містить: загальний індекс TDI проєкту, деталізований перелік виявлених deprecated API із зазначенням файлу, номера рядка, версії deprecation та рекомендованої заміни, декомпозицію проблем за пріоритетами та категоріями, оцінку трудомісткості міграції у людино-днях, а також рекомендації щодо послідовності рефакторингу.

Висновок

Розроблено математичну модель TDI та програмне забезпечення для виявлення технічного боргу при міграції legacy Java-додатків на версію 25. Система забезпечує автоматизований AST-аналіз вихідного коду, класифікацію виявлених проблем за пріоритетом, оцінку трудомісткості міграції та формування детальних звітів із рекомендаціями щодо рефакторингу. Застосунок може використовуватися менеджерами проєктів для планування міграційних робіт та технічними спеціалістами для оцінки необхідних ресурсів і визначення оптимальної послідовності рефакторингу застарілого коду.

Список використаних джерел

1. Cunningham W. The WyCashPortfolio Management System / W. Cunningham // OOPSLA'92 Experience Report. – 1992.
2. Kruchten P. Technical Debt: From Metaphor to Theory and Practice / P. Kruchten, R. Nord, I. Ozkaya // IEEE Software. – 2012. – Vol. 29, No. 6. – P.18-21.
3. McCabe T. A Complexity Measure / T. McCabe // IEEE Transactionson Software Engineering. – 1976. – Vol. SE-2, No. 4. – P.308-320.
4. Alves N. Identification and Management of Technical Debt: A Systematic Mapping Study / N. Alves [etal.] // Information and Software Technology. – 2016. – Vol. 70. – P.100-121.
5. Oracle. JDK 17–25 Release Notes [Електронний ресурс]. – Режим доступу: <https://docs.oracle.com/en/java/>
6. JavaParser: Java Code Analysis Toolkit [Електронний ресурс]. – Режим доступу: <https://javaparser.org/>
7. Fowler M. Refactoring: Improvingthe Design ofExisting Code / M. Fowler. – 2nd ed. – Boston: Addison-Wesley, 2018. – 448p.

ІНТЕЛЕКТУАЛЬНА СИСТЕМА SPARK

Крепич С.Я.¹⁾, Маркевич Д.В.²⁾, Співак І.Я.³⁾, Крепич Р.В.⁴⁾

¹⁻³⁾ Західноукраїнський національний університет;

⁴⁾ ВСП Кам'янець-Подільський фаховий коледж НРЗВО "Кам'янець-Подільський державний інститут"

¹⁾к.т.н., доцент; ²⁾магістр; ³⁾к.т.н., доцент; ⁴⁾ викладач

І. Постановка проблеми

На сьогоднішній день, інклюзивна освіта являється одним з найважливіших напрямів розвитку освітніх систем, так як кількість учнів та студентів з особливими освітніми потребами невідомо зростає з кожним роком. За даними Державної служби статистики, станом на 2020 рік в Україні було зареєстровано близько 163 тисяч дітей з інвалідністю. Згідно даних навчалися лише 78 тисяч дітей, з них майже 37 тисяч досі відвідують сегреговані спеціальні школи, які зазвичай розташовані далеко від місця проживання дітей. В 2023 році показник кількості дітей з інклюзією залишився приблизно на тому ж рівні, але зважаючи на загальне зменшення чисельності дитячого населення, ці дані свідчать про зростання поширеності інвалідності серед дітей.

На жаль, цей відсоток дітей щороку лише невідомо зростає і, враховуючи реалії сьогодення, які лише підживлюють негативні впливи на дитячу психіку, цей відсоток ще не швидко не те, що піде на спад, але хоча б стабілізується. Відповідно перед державою і, зокрема, перед системою освіти стоїть велике завдання - навчитись соціалізувати дітей з інклюзією в буденне життя. Адже, навіть спираючись на статистичні дані, лише станом на початок 2017/18 навчального року в закладах вищої освіти України навчалось понад 13 тисяч студентів з інвалідністю. Більшість із них (приблизно 9735 осіб) здобували освіту в університетах, академіях та інститутах, тоді як лише 3481 студент навчався у коледжах, технікумах та училищах. Також дані опитування опікунів та батьків про загальні труднощі в різних областях, особливо прифронтових, показали, що однією з найбільших проблем і надалі залишається доступність навчальних закладів та і освіти загалом.

II. Мета роботи

Метою роботи є розробка та впровадження системи SPARK (Survey Platform for Assessing Responsive Knowledge), призначеної для моніторингу, оцінювання та покращення взаємодії між викладачами, аспірантами та учнями з особливими освітніми потребами. Система має забезпечувати автоматизований збір, зберігання та аналіз даних щодо мотиваційних факторів, рівня підтримки та ефективності освітньої взаємодії в інклюзивному середовищі.

Реалізація системи SPARK спрямована на підвищення якості інклюзивної освіти шляхом об'єктивного аналізу даних, підтримки викладачів у прийнятті рішень та формування сприятливого навчального середовища для студентів і дітей з особливими освітніми потребами.

III. Основна частина

У рамках розробки системи було здійснено детальний аналіз існуючих цифрових рішень у сфері оцінювання професійної діяльності, мотивації персоналу та психологічного фактору в освітньому середовищі. Зокрема, були розглянуті такі популярні онлайн-платформи та інструменти опитування, як Google Forms, SurveyMonkey та Qualtrics. Кожен з них має певні переваги: наприклад, Google Forms забезпечує швидке створення базових опитувань, SurveyMonkey пропонує розширені шаблони анкет, а Qualtrics надає можливість гнучкого налаштування логіки опитувань та аналітики. Водночас було виявлено низку суттєвих обмежень щодо їх використання у контексті інклюзивної освіти.

Найпоширенішими недоліками зазначених платформ є відсутність можливості комплексно оцінювати сприйняття процесу роботи з дітьми з особливими освітніми потребами та рівень залученості аспірантів, які допомагають учням під час виконання навчальних завдань. Також більшість універсальних сервісів не підтримує побудову мотиваційних кривих, які є ключовим елементом методики оцінки впливу мотиваційних факторів на ефективність навчального процесу. Крім того, зовнішні системи не забезпечують розрахунку індивідуальних та групових показників, що дозволяють визначати фактори, які найбільше підвищують або знижують якість виконання професійних обов'язків працівників інклюзивних груп.

З огляду на зазначені обмеження буде розроблена власна система – SPARK, яка забезпечить не лише стандартний збір даних, але й глибокий аналітичний супровід процесів інклюзивної освіти. Архітектура системи буде складатися з кількох ключових модулів, кожен із яких виконує окрему функцію:

- Модуль збору даних відповідатиме за проведення структурованих опитувань серед аспірантів, викладачів та фахівців, що працюють з інклюзивними групами. Він охоплюватиме анкетування щодо сприйняття роботи з дітьми з вадами розвитку, оцінювання залученості студентів-помічників та фіксацію значущих поведінкових або мотиваційних показників.

- Модуль обробки даних здійснюватиме аналіз відповідей, нормалізацію показників, визначення ключових мотиваційних факторів та розрахунок індивідуальних і групових індикаторів ефективності. Також він відповідатиме за побудову мотиваційних кривих, виявлення тенденцій, порівняння результатів між працівниками та формування рекомендацій щодо покращення навчального процесу.

- База даних — централізоване сховище на базі PostgreSQL, яке міститиме результати опитувань, історію змін показників, профілі респондентів, результати розрахунків та рекомендації, згенеровані аналітичним модулем.

- Користувацький інтерфейс буде реалізований як сучасний веб-додаток, що забезпечує інтуїтивний перегляд аналітичних панелей, мотиваційних кривих, групових порівнянь, а також доступ до рекомендацій та динамічних звітів. Він даватиме можливість викладачам, аспірантам та адміністрації оперативно отримувати інформацію про стан мотивації та ефективність взаємодії в інклюзивних групах.

На етапі проєктування архітектури було створено діаграму класів (див.рис.1), яка відображає логіку взаємодії між основними сутностями системи: респондентом, опитуванням, відповіддю, мотиваційним фактором, рекомендацією, профілем студента-помічника тощо.

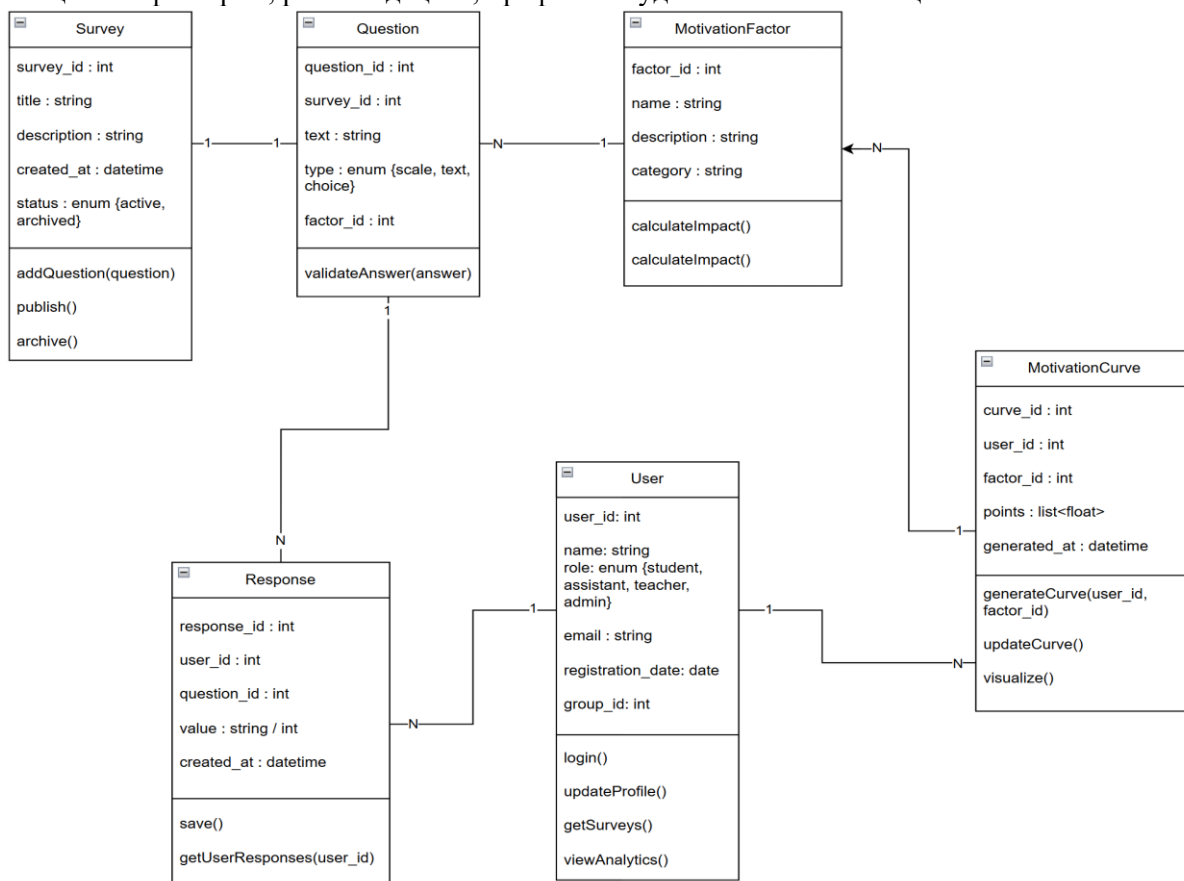


Рисунок 1 – Діаграма класів

Для реалізації серверної частини системи SPARK обрано технологічний стек NestJS, Prisma та PostgreSQL, що забезпечує високу продуктивність, модульність та можливість масштабування. REST-орієнтовані API реалізовані з урахуванням вимог безпеки та структурованої роботи з даними опитувань, відповідей користувачів, мотиваційних факторів та побудованих мотиваційних кривих.

Клієнтська частина створена з використанням React.js та Next.js, що дозволяє розробити швидкий, інтерактивний та доступний інтерфейс для всіх учасників освітнього процесу – студентів, викладачів та студентів-помічників. Основна панель керування забезпечує доступ до опитувань, аналітичних даних та індивідуальних рекомендацій.

Окрім цього, система включає декілька функціональних модулів:

- Модуль опитувань (див.рис.2) – забезпечує створення структурованих анкет, визначення типів запитань та фіксацію відповідей користувачів. Усі результати зберігаються у вигляді відповідей (Response), що надалі використовуються аналітичним блоком.

- Модуль мотиваційних факторів – містить перелік факторів, які впливають на сприйняття роботи з дітьми з особливими освітніми потребами та на рівень професійної залученості. Кожне питання опитування пов'язане з відповідним мотиваційним фактором, що дозволяє виконувати точковий аналіз.

- Аналітичний модуль – відповідає за побудову індивідуальних та групових мотиваційних кривих. На основі відповідей користувачів система автоматично формує візуалізацію динаміки мотивації та визначає фактори, що мають найбільший вплив на ефективність роботи.

- Панель користувача – дозволяє студентам, викладачам і помічникам проходити опитування, переглядати результати власних мотиваційних кривих, порівнювати їх із попередніми періодами та відстежувати рівень залученості.

Таким чином, система SPARK поєднує функціональність збору відповідей, аналітичної обробки даних, візуалізації мотиваційних показників та гнучкої взаємодії з користувачами, забезпечуючи комплексний підхід до оцінки ефективності роботи в інклюзивних освітніх середовищах.

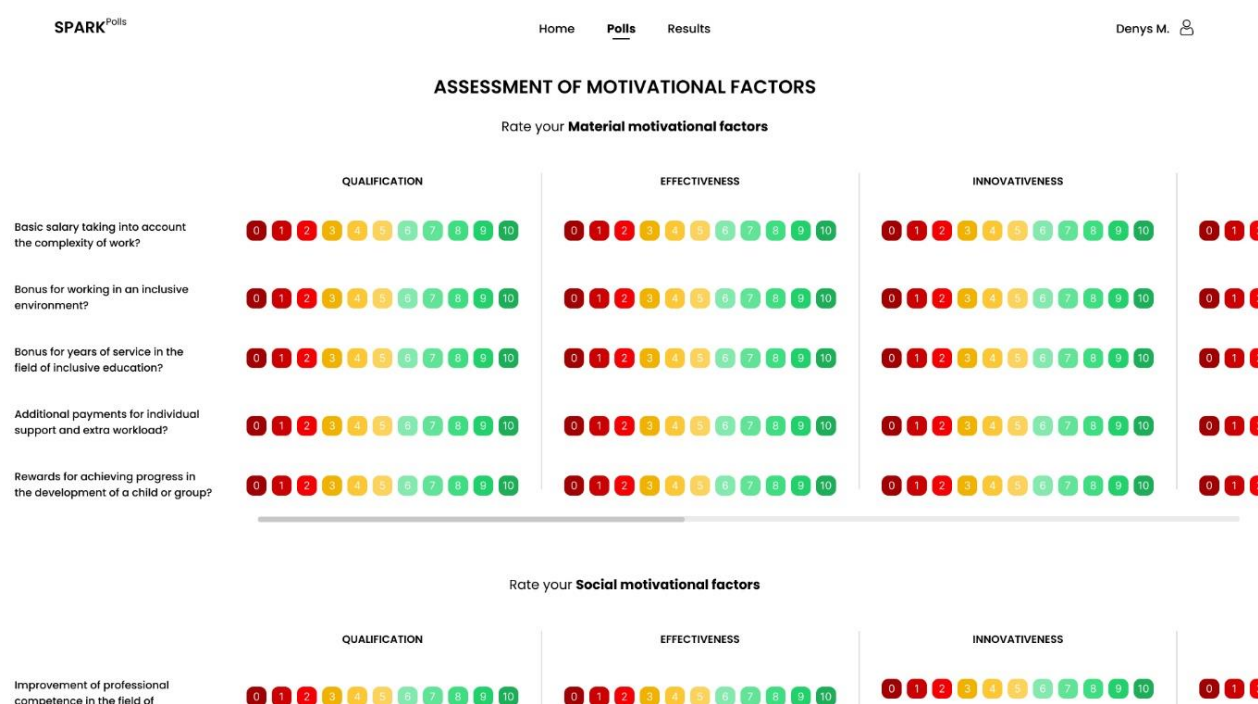


Рисунок 2 – Сторінка опитування

Висновок

Розроблена система SPARK – Survey Platform for Assessing Responsive Knowledge забезпечує ефективний інструмент для автоматизованого збору, аналізу та візуалізації даних щодо мотиваційних факторів працівників, які працюють з інклюзивними дітьми.

Використання системи SPARK сприяє підвищенню ефективності управління персоналом, підтримує науково обгрунтоване прийняття рішень та створює умови для покращення якості роботи фахівців у сфері інклюзивної освіти.

Список використаних джерел

1. Spivak I., Krepych S., Spivak S., Fedorov O. Approach to estimate the level of influence of motivation on the effectiveness of employees9 depending on their needs, in 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019 – Proceedings 8847890, pp.46-49
2. Spivak I., Bayurskii A., Krepych S., Spivak S., Criterion for evaluation the level of experts competence during the evaluation of a software system based on the modified interval method of expert evaluation, in 2021 11th International Conference on Advanced Computer Information Technologies, ACIT 2021 – Proceedings, 2021, pp.582-586
3. Крепич С.Я. Програмний комплекс оцінювання функціональної придатності пристроїв при заданих допустимих значеннях вихідних характеристик та допусків на параметри їх елементів. Сучасні комп'ютерні інформаційні технології: Матеріали V Всеукраїнської школи-семінару молодих вчених і студентів АСІТ'2015. – Тернопіль: ТНЕУ, 2015. – С. 23-25.
4. Співак І.Я., Крепич С.Я. Оцінювання рівня впливу мотивації праці на ефективність діяльності. *Інформаційні технології та комп'ютерна інженерія*. Вип.3, 2020. – С.22-29.

АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПТИМІЗАЦІЇ ПАРАМЕТРІВ СХОДІВ У ЖИТЛОВИХ ПРИМІЩЕННЯХ

Войтюк І.Ф.¹⁾, Кузьмяк Б.В.²⁾

Західноукраїнський національний університет

¹⁾к.т.н., доцент; ²⁾ магістрант

I. Постановка проблеми

Проектування сходів у житлових приміщеннях вимагає точних розрахунків параметрів для забезпечення безпеки, комфорту та відповідності будівельним нормам. Онлайн-калькулятори сходів стали популярним інструментом для архітекторів, будівельників та приватних замовників. Проте різні програмні рішення мають суттєві відмінності у функціональності, точності розрахунків та зручності використання.

II. Мета роботи

Метою дослідження є критичний аналіз функціональних можливостей та обмежень провідних програмних засобів для оптимізації параметрів сходів для визначення їх придатності до проектування сходів у житлових приміщеннях.

III. Аналіз програмного забезпечення

Для порівняльного аналізу було обрано три онлайн-платформи, що представляють різні підходи до розрахунку параметрів сходів.

Calculator.net/stair-calculator (див.рис.1) є комплексним безкоштовним інструментом, що пропонує базову та розширену версії калькулятора. Базова версія розраховує основні параметри (висоту підйому, довжину прогону, довжину косоура) на основі введених даних про загальну висоту та прогон. Розширена версія додає можливості налаштування товщини проступу, обмежень по висоті прольоту та типу монтажу косоура. Програма автоматично перевіряє відповідність розрахунків будівельним нормам США, зокрема вимогам щодо мінімальної глибини проступу 10 дюймів та максимальної висоти підступінка 7,75 дюймів. Інтерфейс містить наочну діаграму з позначенням усіх елементів сходів та таблицю конвертації дробових значень у десяткові. Недоліком є відсутність можливості експорту результатів та обмежена підтримка інших систем будівельних норм.

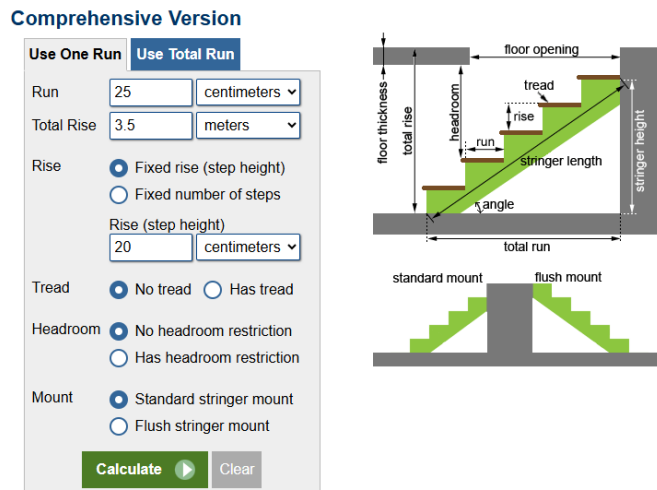


Рисунок 1 – Робоча форма програми Calculator.net/stair-calculator

Stair-calculator.net позиціонується як професійний інструмент для точного проектування сходів у житловому, комерційному та терасному будівництві. Платформа базується на стандартах IRC, IBC та ADA. Калькулятор пропонує можливість збереження результатів у форматі PDF та зображення, що значно полегшує роботу з документацією. Інтерфейс включає візуалізацію конструкції з позначенням розмірів та кутів. Додатковою перевагою є наявність освітніх матеріалів про золоте правило Блонделя, вимоги безпеки та рекомендації щодо освітлення сходів. Обмеженням є орієнтація виключно на американські будівельні стандарти без підтримки європейських норм.

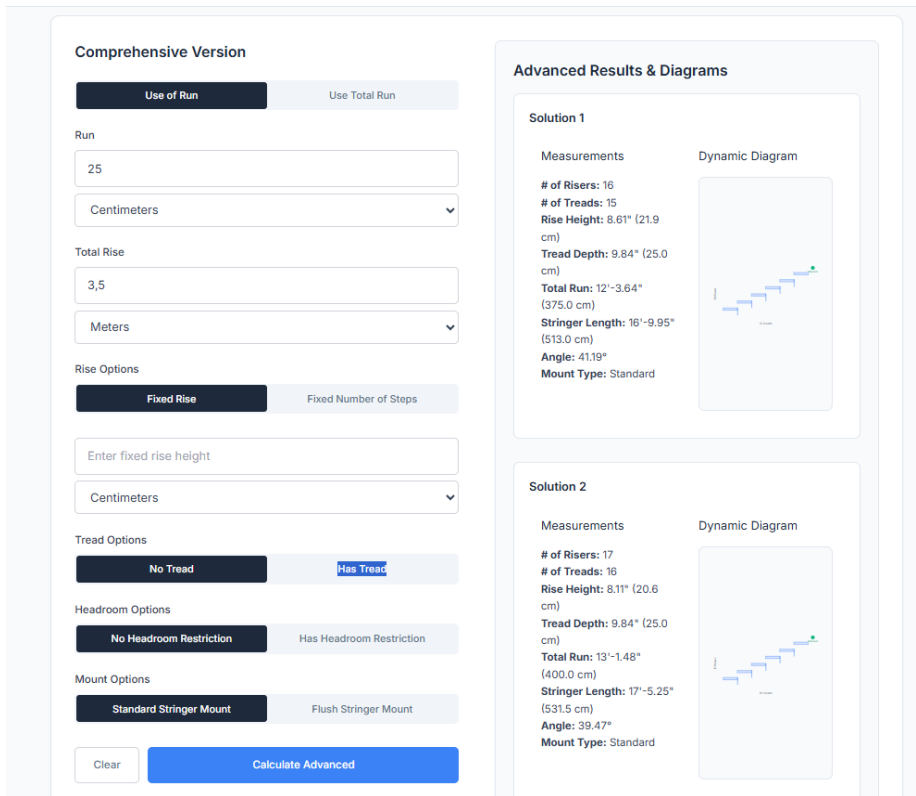


Рисунок 2 – Робоча форма програми Stair-calculator.net

Staircalculator.net пропонує двохрівневий підхід із базовим режимом для початківців та комплексним режимом для професіоналів. Програма використовує формулу Блонделя (оптимальне відношення між висотою підйому h (мм) та шириною проступу b (мм)) для розрахунку оптимальних параметрів, що забезпечує комфортну ходу для більшості людей.

$$2h + b = 600 - 650(\text{мм}) \quad (1)$$

Калькулятор підтримує різні одиниці вимірювання (фути, дюйми, метри, сантиметри) та автоматично виконує конвертацію. Інтерфейс адаптовано для мобільних

пристроїв, що дозволяє виконувати розрахунки безпосередньо на будівельному майданчику. Слабкою стороною є відсутність детальної документації про використовувані алгоритми та обмежена функціональність експорту даних.

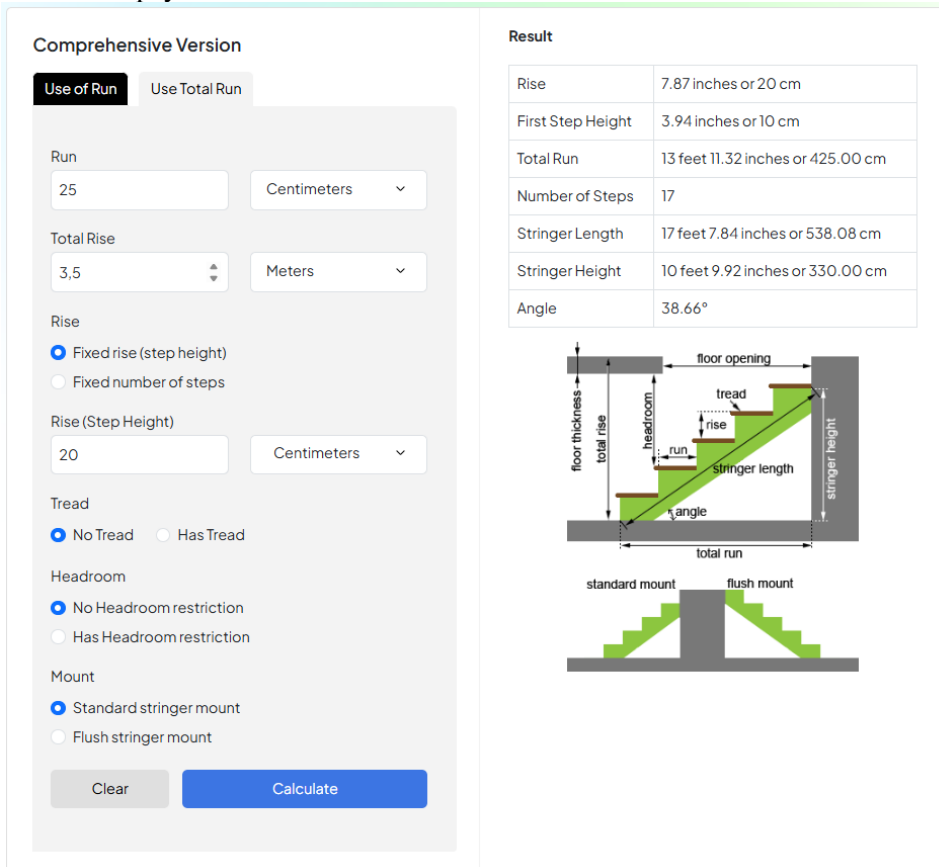


Рисунок 3 – Робоча форма програми Staircalculator.net

Порівняльна характеристика онлайн-калькуляторів сходів

Назва платформи	Складність інтерфейсу	Підтримка стандартів	Експорт результатів	Мобільна версія	Освітні матеріали
Calculator.net	Середня	IRC (CIPA)	Відсутній	Обмежена	Базові
Stair-calculator.net	Висока	IRC, IBC, ADA	PDF, зображення	Повна	Розширені
Staircalculator.net	Низька	Формула Блонделя	Обмежений	Повна	Середні

IV. Обґрунтування результатів

Аналіз показав, що жоден з розглянутих калькуляторів не забезпечує повної функціональності для проєктування сходів у житлових приміщеннях відповідно до українських будівельних норм ДБН В.2.2-15:2019. Усі програми орієнтовані на американські стандарти або загальноприйняті європейські формули без урахування специфіки національного законодавства.

Жодна з досліджуваних систем не забезпечує повного набору функцій для комплексного проєктування сходових конструкцій різної складності.

Calculator.net характеризується критичною відсутністю функції експорту результатів, що унеможливує використання системи для створення робочої документації. Спрощений алгоритм розрахунку не враховує формулу Блонделя, що може призвести до проєктування неоптимальних конструкцій з погіршеною ергономікою. Система не підтримує розрахунок сходів з поворотними майданчиками та забіжними сходишками.

Stair-Calculator.net демонструє найкращу функціональність серед досліджуваних систем, проте має суттєві обмеження у роботі зі складними геометричними конфігураціями. Система не підтримує розрахунок гвинтових сходів, забіжних сходинок та конструкцій з проміжними майданчиками. Відсутність можливості збереження проєктів ускладнює ітеративний процес проєктування та унеможливує ведення бази типових рішень. Критичним обмеженням є фіксована база розрахунків виключно на імперських одиницях вимірювання з наступною конвертацією у метричну систему, що призводить до похибок округлення до 0,3 см при розрахунках проступів.

StairCalculator.net забезпечує оптимальний баланс між функціональністю та доступністю завдяки архітектурі з вибором режиму складності. Проте критичним недоліком є відсутність функції експорту результатів у форматі PDF, що обмежує можливості інтеграції у робочі процеси професійного проєктування. Відсутність детальної технічної документації про використовувані алгоритми перевірки безпеки викликає сумніви щодо надійності розрахунків для критичних проєктів.

Висновок

Проведений критичний аналіз програмного забезпечення для оптимізації параметрів сходів у житлових приміщеннях виявив суттєві відмінності у функціональності, точності розрахунків та зручності використання досліджуваних систем. Calculator.net забезпечує базовий функціонал для попереднього проєктування, Stair-Calculator.net є оптимальним вибором для професійної діяльності з необхідністю підготовки робочої документації, а StairCalculator.net пропонує універсальне рішення для користувачів різного рівня підготовки.

Результати дослідження показали, що оптимальний вибір програмного забезпечення повинен базуватися на комплексній оцінці специфіки проєкту, вимог до документації, рівня підготовки користувача. Перспективним напрямком є розробка спеціалізованого програмного забезпечення для оптимізації параметрів сходів, для універсального використання широким колом користувачів від початківців до досвідчених фахівців, що враховуватиме вимоги українських будівельних норм та надаватиме розширені можливості візуалізації проєктних рішень. Існуючі онлайн-калькулятори можуть використовуватися, як допоміжні інструменти після адаптації розрахунків до національних стандартів.

Список використаних джерел

1. «Будівельне проєктування» Нойферт Е - Дніпро: Книжкова Літера, 2017. - 614 с.
2. ДБН В.2.2-15:2019. Будинки і споруди. Житлові будинки. Основні положення. - Київ: Мінрегіон України, 2019. - 52 с.
3. Calculator.net StairCalculator [Електронний ресурс]. – Доступ: <https://www.calculator.net/stair-calculator.html>
4. Stair-Calculator Professional Tool [Електронний ресурс]. – Доступ: <https://www.stair-calculator.com/>
5. Staircalculator.net OnlinePlatform [Електронний ресурс]. – Доступ: <https://staircalculator.net/>
6. «The Staircase: Studies of Hazards, Falls, and Safer Design» Дж. Темплер. – Кембридж 1992. - 264 р.
7. International Building Code (IBC) Requirements for Stairs [Електронний ресурс]. - Доступ: <https://www.iccsafe.org>

ВЕБ-ПЛАТФОРМА ДЛЯ СОЦІАЛЬНОЇ ІНТЕГРАЦІЇ ВНУТРІШНЬО ПЕРЕМІЩЕНИХ ОСІБ В УКРАЇНІ

Кліщ С.С.¹⁾, Крепич С.Я.²⁾, Співак І.Я.³⁾

Західноукраїнський національний університет

¹⁾ бакалавр; ²⁻³⁾ к.т.н., доцент

Постановка проблеми

Повномасштабна війна, розв'язана Росією проти України, призвела до безпрецедентного внутрішнього переміщення населення. Кількість внутрішньо переміщених осіб (ВПО) стрімко зростає з приблизно 1,5 млн (zareєстрованих до 2022 року) до понад 4,6 млн станом на кінець 2024 року. Майже п'ята частина цих переселенців – діти, які зазнають значних соціально-психологічних труднощів, пов'язаних зі зміною середовища проживання та звичного кола спілкування. Під час вимушеного переселення люди втрачають не лише житло, а й соціальні зв'язки, друзів, звичне оточення, що особливо ускладнює адаптацію дітей і підлітків у нових громадах. Водночас наразі бракує цифрових інструментів, які б сприяли соціальній інтеграції ВПО. Існуючі програми підтримки залишаються фрагментованими і не охоплюють повною мірою потребу в довгостроковій інтеграції переселенців у приймаючих громадах.

Мета

Метою проекту є створення веб-платформи, яка допоможе спростити процес соціальної інтеграції внутрішньо переміщених осіб у нових громадах. Особливий акцент робиться на підтримці переселенців на найскладнішому початковому етапі зміни звичного способу життя – при переїзді до нового міста або регіону, в незнайоме соціальне середовище.

Основний виклад

Запропоноване рішення – веб-платформа, яка слугуватиме інтеграційною соціальною мережею для переселенців. Діаграма варіантів використання (див.рис.1) відображає ключові функції системи та сценарії взаємодії користувачів. Основні компоненти платформи включають:

- **Реєстрація користувачів.** Нові користувачі створюють обліковий запис, вказуючи статус (ВПО, місцевий мешканець, волонтер тощо) та базові дані. Після реєстрації користувач отримує доступ до всіх функцій платформи.
- **Профіль користувача.** Кожен користувач має персональний профіль з основною інформацією (місто походження, поточне місто, інтереси, потреби або пропозиції допомоги). Профіль дозволяє іншим швидко зрозуміти потреби та можливості конкретної особи, сприяючи налагодженню контактів.
- **Створення спільнот.** Користувачі можуть ініціювати створення тематичних чи локальних спільнот (груп) для спілкування та взаємодопомоги. Приклади спільнот: «Переселенці у місті N», «Батьки школярів-ВПО», «ІТ-фахівці серед ВПО» тощо. Спільноти модеруються їх засновниками або призначеними модераторами і слугують майданчиком для групових обговорень, проведення зустрічей, заходів, обміну порадами.
- **Розділ «Допомога».** Окремий модуль платформи призначений для публікації та перегляду оголошень щодо допомоги. ВПО можуть знайти актуальну інформацію про доступні ресурси (житло, гуманітарну та психологічну допомогу, освітні можливості, вакансії тощо) у новому регіоні.
- **Скарги та зворотний зв'язок.** Для підтримання безпечного і комфортного середовища передбачено механізм подання скарг. Користувач може повідомити про неправомірну поведінку, образливий контент чи порушення правил платформи.
- **Модерація.** Адміністратори та модератори платформи мають окремі привілеї для перевірки контенту і користувацької активності. Вони розглядають скарги, видаляють неналежні матеріали, можуть блокувати профілі порушників, а також відповідають на запитання користувачів. Система модерації забезпечує дотримання етичних норм і правил спільноти, що особливо важливо при роботі з вразливими групами користувачів (діти, травмовані війною).
- **Чати та повідомлення.** Вбудована система комунікації дозволяє користувачам обмінюватися повідомленнями. Реалізовано як приватні чати між двома користувачами, так і групові чати в межах створених спільнот. Ця функція сприяє оперативному спілкуванню: переселенці

можуть напряду звернутися за порадою до місцевих мешканців, отримати емоційну підтримку від людей з подібним досвідом, організувати зустрічі тощо.



Рисунок 1 – Діаграма варіантів використання

Висновки

Розробка спеціалізованої веб-платформи для соціальної інтеграції внутрішньо переміщених осіб є своєчасною і доцільною з огляду на масштаб актуальної міграційної кризи та виявлені проблеми адаптації переселенців. Очікується, що використання платформи сприятиме налагодженню соціальних зв'язків і взаємопідтримки: переселенці зможуть швидше знайти друзів і однодумців, отримати необхідну допомогу або консультацію, долучитися до життя громади.

Список використаних джерел

1. Юрченко Н. *Скільки в Україні переселенців і де найбільше ВПО: статистика за областями* [Електронний ресурс] – Режим доступу: <https://www.rbc.ua/rus/news/skilki-ukrayini-pereselentsiv-i-de-naybilshe-1729603384.html>.
2. *Соціалізація та інтеграція дітей ВПО у громадах: рекомендації МОН* [Електронний ресурс] – Режим доступу: <https://osvita.ua/school/92618/>.
3. S. Krepych, I.Spivak, *Improvement of SVD algorithm to increase the efficiency of recommendation systems*. Advanced Information Systems, 5(4), 2021, pp.55-59
4. S.Krepych, I.Spivak, *Algorithm of Automatic Generation of Hotel Descriptions Using Templates Based on Markov Chains*, 2018 International Scientific-Practical Conference Problems of Infocommunications Science and Technology Kharkiv, pp.257-260
5. A. Bayurskii, S.Krepych, *Intelligent System Analyzing Quality of Land Plots*, International Conference “Advanced Computer Information Technologies” ACIT-2018, Ceske Budejovice, Czech Republic, June 1-3, 2018, pp. 166-169

ПРОЄКТУВАННЯ СТРУКТУРИ БАЗИ ДАНИХ СИСТЕМИ ОНЛАЙН-ЗАПИСУ ДО ЛІКАРЯ

Могильська І.М.¹⁾, Тимчишин В. С.²⁾
Західноукраїнський національний університет
^{1) бакалавр; ^{2) доктор філософії, старший викладач}}

I. Вступ

Цифровізація медичних послуг є одним із пріоритетних напрямків розвитку сучасної системи охорони здоров'я. Впровадження електронних сервісів у сфері e-Health сприяє підвищенню доступності медичної допомоги, оптимізації робочих процесів та зменшенню адміністративного навантаження на медичний персонал. Одним із ключових компонентів таких сервісів є системи онлайн-запису пацієнтів на прийом до лікаря, які забезпечують автоматизовану взаємодію між пацієнтами, лікарями та адміністрацією медичних закладів.

Фундаментом подібних інформаційних систем є реляційна база даних, яка відповідає за збереження, обробку та захист персональних і медичних даних. Якісно спроектована структура бази даних повинна гарантувати цілісність інформації, масштабованість та ефективну обробку запитів у режимі реального часу.

II. Постановка проблеми

Сучасні заклади охорони здоров'я все частіше впроваджують онлайн-сервіси для запису пацієнтів на прийом до лікаря з метою підвищення доступності медичних послуг, зменшення черг та оптимізації роботи персоналу. Проте ефективне функціонування таких систем значною мірою залежить від коректно спроектованої структури бази даних, яка має забезпечувати надійне зберігання персональних даних пацієнтів, інформації про лікарів, графіки прийому, записи на консультації та історію взаємодій. Недостатньо продумана модель даних може призводити до втрати цілісності інформації, дублювання записів, проблем із масштабуванням та зниження продуктивності системи. Типові рішення без чіткої нормалізації та продуманої системи зв'язків між сутностями ускладнюють подальший розвиток функціоналу, зокрема інтеграцію з іншими медичними інформаційними системами або аналітичними сервісами. У зв'язку з цим постає потреба в обґрунтованому проєктуванні структури реляційної бази даних системи онлайн-запису до лікаря, яка б забезпечувала надійність, масштабованість і зручність подальшої програмної реалізації.

III. Мета роботи

Метою даної роботи є проєктування структури реляційної бази даних для системи онлайн-запису до лікаря, яка забезпечує логічну узгодженість даних, підтримку ключових бізнес-процесів та готовність до інтеграції у клієнт-серверну архітектуру веб-застосунку, а також мінімізує ризики конфліктів розкладу та порушення цілісності інформації.

IV. Основна частина

Проєктування структури бази даних системи онлайн-запису до лікаря, у ході цієї роботи, здійснювалося із застосуванням підходів концептуального та логічного моделювання даних характерних для реляційних інформаційних систем. На початковому етапі було проведено аналіз предметної області з метою визначення ключових об'єктів, їх атрибутів та взаємозв'язків [1,2].

На рисунку 1 представлено ER-діаграму, яка відображає логічну структуру бази даних та взаємодію основних сутностей системи. У межах моделі виділено такі базові сутності: Users, Patients, Doctors, Specializations, Schedules та Appointments. Кожна з них представлена окремою таблицею з визначеним первинним ключем, що забезпечує унікальність записів і можливість їх однозначної ідентифікації.

Сутність Users є центральною у схемі та використовується для реалізації рольової моделі доступу. Усі типи користувачів, включно з пацієнтами, лікарями та адміністратором, зберігаються в єдиній таблиці користувачів, а їхні повноваження визначаються атрибутом role. Такий підхід зменшує складність схеми та спрощує її подальше розширення.

Сутності Patients та Doctors пов'язані з таблицею Users за допомогою зовнішніх ключів, що дозволяє розмежувати облікові дані користувачів і спеціалізовану інформацію, пов'язану з

пацієнтами та медичним персоналом. Лікарі додатково асоційовані з таблицею Specializations, яка визначає їхній професійний напрям.

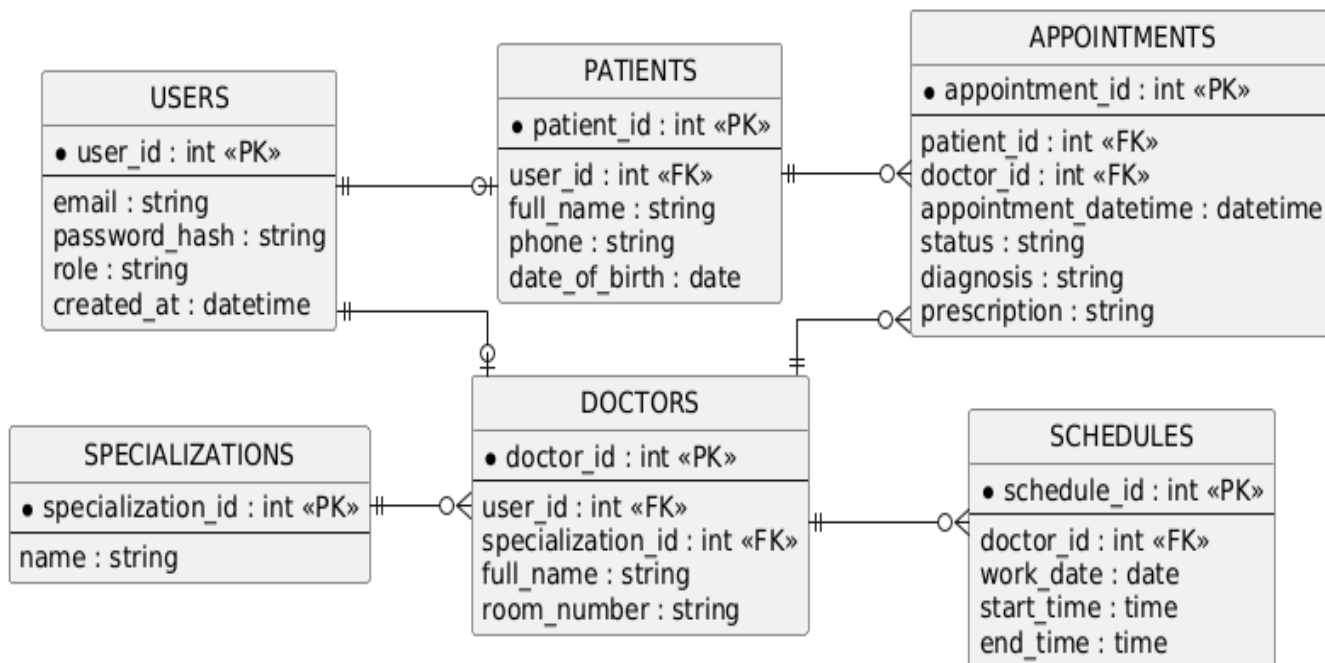


Рисунок 1 – ER-діаграма структури бази даних системи онлайн-запису до лікаря

Як видно з ER-діаграми, між основними сутностями системи реалізовано різні типи зв'язків. Зв'язки типу один-до-одного використовуються між таблицями Users та Patients, а також Users та Doctors, що дозволяє розділити облікові дані користувача та його доменно-орієнтовану інформацію.

Зв'язки типу один-до-багатьох застосовано між сутностями Specializations і Doctors, Doctors і Schedules, а також між Patients і Appointments, що відповідає логіці медичної предметної області та дозволяє одному лікарю або пацієнту мати кілька пов'язаних записів.

Процес запису пацієнтів на прийом моделюється за допомогою сутності Appointments, яка пов'язує пацієнта, лікаря та конкретний часовий інтервал. Зв'язки між сутностями реалізовано за допомогою зовнішніх ключів, що гарантує референтну цілісність даних і запобігає виникненню неузгоджених або конфліктних записів.

Під час проєктування також враховано принципи мінімізації надлишковості інформації та логічного розділення даних, що позитивно впливає на підтримуваність схеми та продуктивність виконання SQL-запитів.

Запропонована структура бази даних орієнтована на використання у клієнт-серверній архітектурі та може бути інтегрована з серверною частиною веб-застосунку через програмні інтерфейси прикладного програмування, що створює основу для подальшого розвитку та масштабування системи.

Висновок

У роботі здійснено проєктування структури реляційної бази даних системи онлайн-запису до лікаря на основі концептуального та логічного моделювання, що дозволило виділити ключові сутності предметної області, визначити їх атрибути та взаємозв'язки відповідно до бізнес-процесів медичного закладу.

Запропонована модель забезпечує логічну узгодженість і референтну цілісність даних, підтримує рольову модель доступу, мінімізує надлишковість інформації та створює умови для ефективної обробки запитів.

Список використаних джерел

1. Connolly T., Begg C. Database Systems: A Practical Approach to Design, Implementation, and Management. — 7th ed. — Harlow : Pearson Education, 2020. — 1408 p.
2. Coronel C., Morris S. Database Systems: Design, Implementation, & Management. — 13th ed. — Boston : Cengage Learning, 2021. — 832 p.

ВЕБ-ПЛАТФОРМА ДЛЯ ОРГАНІЗАЦІЇ ТА АГРЕГАЦІЇ МІСЬКИХ ПОДІЙ

Крепич С.Я.¹⁾, Молдавчук А.В.²⁾, Співак І.Я.³⁾

Західноукраїнський національний університет

^{1)к.т.н., доцент;} ^{2) бакалавр;} ^{3)к.т.н., доцент}

I. Постановка проблеми

Сучасне міське середовище характеризується високою динамікою культурного та соціального життя. У місті Тернопіль щодня відбуваються десятки заходів різного масштабу: від великих концертів до локальних воркшопів та благодійних зборів. Проте, ключовою проблемою залишається інформаційна фрагментація. Інформація про події розпорошена між соціальними мережами, новинними порталами та афішами, що ускладнює її пошук для кінцевого користувача. Існуючі платформи (Concert.ua, Eventbrite) орієнтовані або на продаж квитків для комерційних шоу, або на глобальний ринок, ігноруючи потреби локальних організаторів некомерційних подій. Це зумовлює необхідність розробки спеціалізованого програмного забезпечення — веб-платформи "EvenTer", яка б централізувала інформаційні потоки міста.

II. Мета роботи

Метою роботи є проектування та розробка веб-орієнтованої інформаційної системи для автоматизації процесів публікації, модерації та пошуку міських подій. Система покликана забезпечити організаторам зручний інструмент для просування заходів, а мешканцям міста – єдину точку доступу до актуальної афіші з можливостями фільтрації та персоналізації.

III. Основна частина

При дослідженні аналогічних програмних рішень було виявлено, що ключовим недоліком існуючих систем є відсутність підтримки повного життєвого циклу локальної події для малих організаторів. Для усунення зазначених недоліків та забезпечення ефективної взаємодії організаторів з аудиторією, було спроектовано власне програмне забезпечення.

Розроблена система "EvenTer" базується на клієнт-серверній архітектурі.

Серверна частина (Backend) реалізована на платформі .NET (ASP.NET CoreWeb API), що забезпечує високу продуктивність. Для взаємодії з даними використано ORM EntityFrameworkCore.

В якості системи управління базами даних обрано Microsoft SQL Server, що гарантує надійне збереження даних та підтримку транзакцій.

Клієнтська частина (Frontend) взаємодіє з сервером через REST API, використовуючи захищений протокол HTTPS та JWT-токени для авторизації.

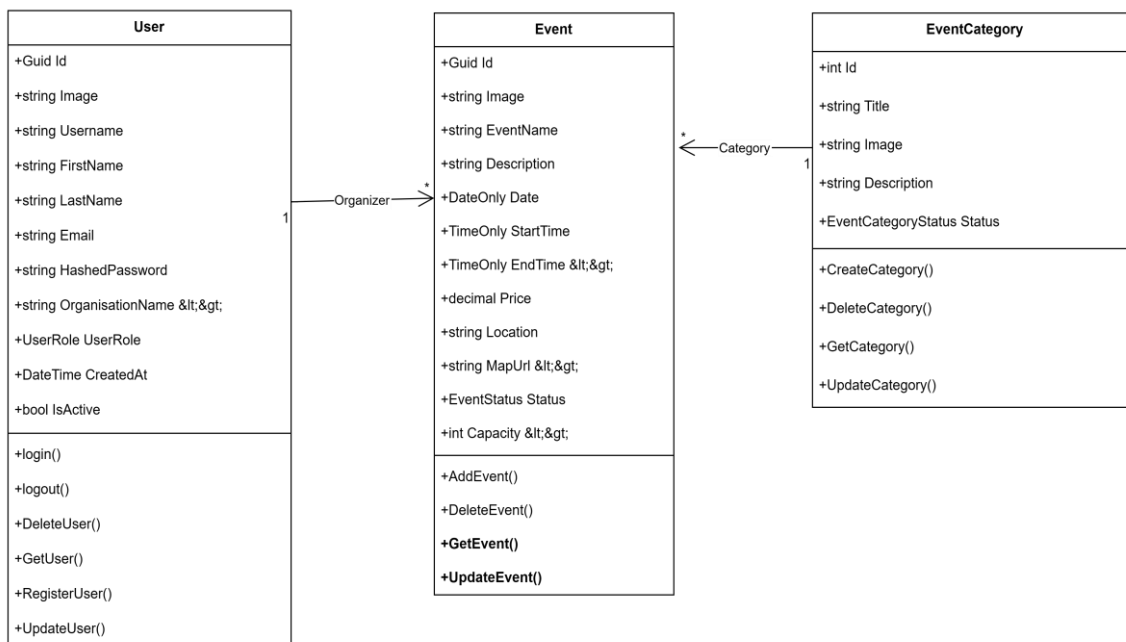


Рисунок 1 – Діаграма класів

- Архітектура бази даних побудована навколо трьох ключових сутностей:
1. User – користувач системи з ролями (Гість, Організатор, Адміністратор).
 2. Event – сутність події, що містить описові дані, дату, час та локацію.
 3. EventCategory – класифікатор для зручної фільтрації контенту.

Для проектування системи було використано об'єктно-орієнтований підхід.

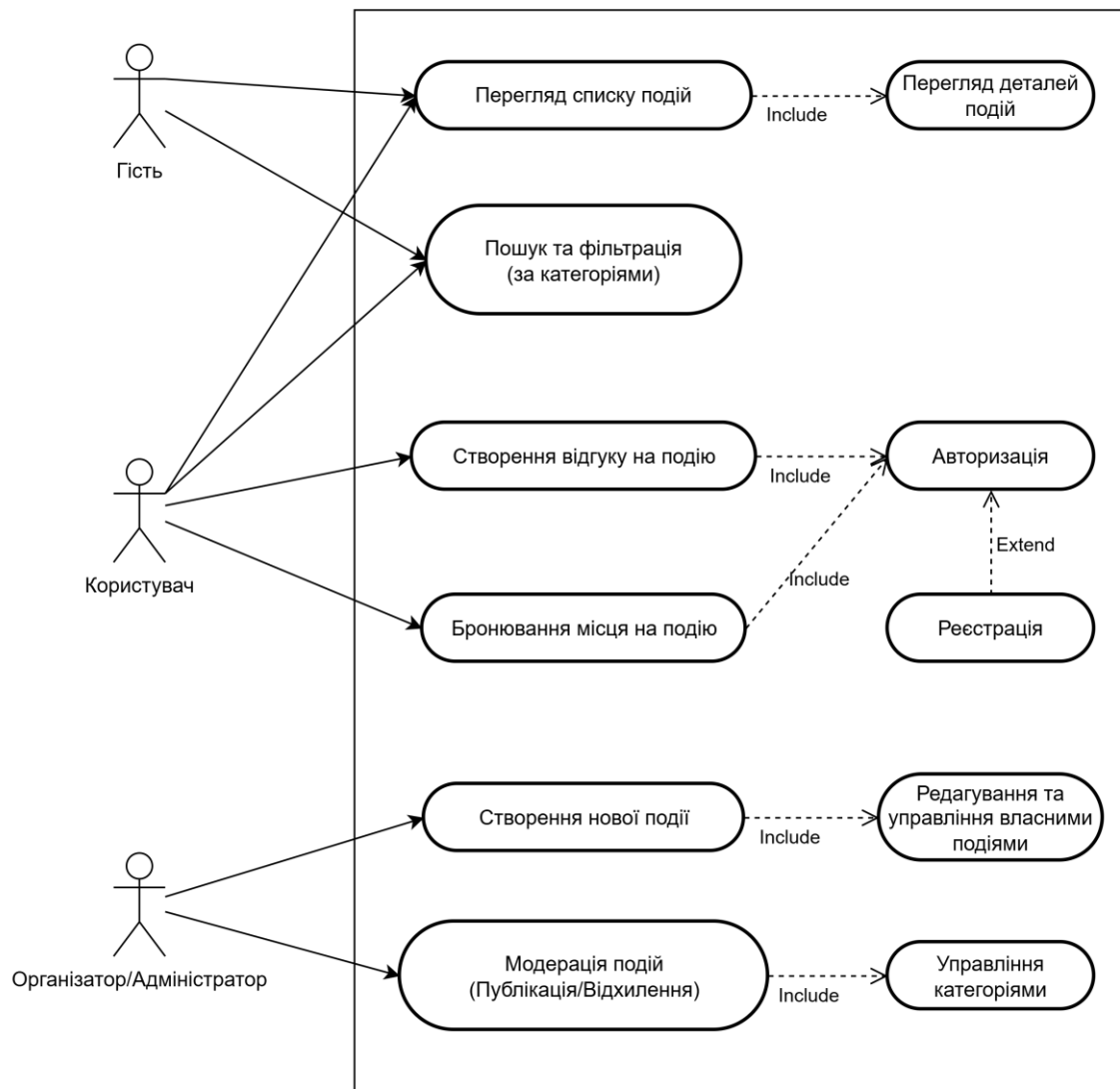


Рисунок 2 – Діаграма варіантів використання системи "EvenTer"

Особливістю реалізації є впровадження механізму премодерації. Алгоритм роботи передбачає валідацію даних на етапі введення, збереження події зі статусом «Очікує» та процес затвердження адміністратором, що дозволяє уникнути публікації небажаного контенту.

Висновок

У роботі обґрунтовано необхідність створення та розроблено веб-платформу "EvenTer" для організації міських подій. Система вирішує проблему інформаційної децентралізації у м. Тернопіль, надаючи зручний інструмент для взаємодії між організаторами та відвідувачами. Практична реалізація підтвердила ефективність обраного технологічного стека (.NET) для забезпечення швидкодії та масштабованості рішення.

Список використаних джерел

1. Троелсен Е., Джепикс Ф. Мова програмування C# 7.0 і платформа .NET Core 2.0. — Київ : Вільямс, 2018. — 1328 с.
2. Фаулер М. Архітектура корпоративних програмних додатків. — Київ : Вільямс, 2017. — 544 с.
3. S.Krepnych, P.Stakhiv, I.Spivaak, Analysis of the tolerance area parameters REC based on technological area scattering, 12-th International Conference "The Experience Of Designing And Application Of CAD Systems in Microelectronics" Polyana Svalyava (Zakarpattia) UKRAINE 19-23 February 2013. – pp.179-180
4. S.Krepnych, I.Spivak, Forecasting system of utilities service costs based on neural network, Advanced Information Systems, Vol.4, No.4, 2020, pp. 102-108

ВЕБ-СИСТЕМА ДЛЯ ОРГАНІЗАЦІЇ ДІЯЛЬНОСТІ ТАНЦЮВАЛЬНОГО КОЛЕКТИВУ

Мелько І.М.¹⁾, Тимчишин В.С.²⁾

*Західноукраїнський національний університет
1)бакалавр; 2) доктор філософії, старший викладач*

І. Постановка проблеми

Робота танцювального колективу це не лише творчий процес, а й постійна організаційна діяльність. Вона включає планування репетицій, підготовку до виступів, координацію учасників, своєчасне інформування про зміни та домовленості. Від того, наскільки злагоджено організовані ці процеси, безпосередньо залежить результат роботи колективу.

На практиці більшість танцювальних колективів використовують для цього різні інструменти одночасно: месенджери, нотатки, електронні таблиці або навіть паперові записи. Такий підхід є незручним, оскільки інформація зберігається в різних місцях, швидко втрачає актуальність і часто призводить до непорозумінь між керівником та учасниками.

У зв'язку з цим виникає потреба у створенні єдиної веб-системи, яка б об'єднала основні організаційні процеси танцювального колективу та спростила щоденну роботу з інформацією.

II. Мета роботи

Метою роботи є розробка веб-системи для організації діяльності танцювального колективу, яка дозволяє зручно керувати інформацією про учасників, розклад репетицій і заходів, а також забезпечує ефективну комунікацію між керівником і членами колективу.

III. Основна частина

Запропонована веб-система побудована на основі клієнт-серверної архітектури з використанням трирівневого підходу. Така архітектура є зрозумілою, надійною та добре підходить для систем організаційного призначення.

Рівень представлення відповідає за взаємодію користувачів із системою через веб-інтерфейс. Саме на цьому рівні керівник та учасники колективу отримують доступ до розкладу репетицій, переглядають актуальну інформацію та отримують організаційні повідомлення. Інтерфейс орієнтований на простоту використання та доступність з різних пристроїв[1]. Клієнтська частина відповідає за відображення інформації, обробку дій користувачів і формування запитів до системи, тоді як серверна частина забезпечує виконання бізнес-логіки, перевірку прав доступу та роботу з даними. Такий підхід зменшує складність програмної реалізації та підвищує підтримуваність веб-системи.

Рівень прикладної логіки забезпечує виконання основних функцій системи. До них належить управління складом колективу, формування та оновлення розкладу, обробка інформаційних повідомлень і контроль доступу користувачів. Саме на цьому рівні реалізується логіка організації діяльності танцювального колективу[2]. Обмін даними між клієнтом і сервером здійснюється через програмний інтерфейс прикладного програмування з використанням структурованих форматів даних. Це забезпечує незалежність клієнтської реалізації від внутрішньої логіки сервера та дозволяє модифікувати або розширювати функціональність системи без порушення її цілісності. Завдяки цьому веб-система може еволюціонувати відповідно до нових вимог без необхідності суттєвих змін у користувацькому інтерфейсі.

Рівень даних призначений для зберігання всієї необхідної інформації: відомостей про учасників, розкладів репетицій, заходів та інших організаційних даних. Такий підхід дозволяє забезпечити цілісність інформації та зручний доступ до неї. Централізоване зосередження бізнес-логіки та механізмів контролю доступу на серверному рівні сприяє підвищенню безпеки та масштабованості системи. Клієнт-серверна архітектура створює умови для ефективної обробки зростаючої кількості запитів, підтримки багатокористувацького режиму та подальшого розширення функціональних можливостей веб-системи без зниження стабільності її роботи.

Для забезпечення ефективної організації діяльності танцювального колективу під час проектування веб-системи було проведено аналіз основних ролей і сценаріїв взаємодії користувачів із системою. Визначено ключові ролі, зокрема керівника колективу та учасників, кожна з яких має власний набір повноважень і доступних функцій. Такий підхід дозволив чітко окреслити межі

відповідальності користувачів і закласти основу для реалізації механізмів контролю доступу на рівні прикладної логіки.

Особливу увагу приділено формалізації процесів планування репетицій та інформування учасників колективу. Ці процеси є критичними з точки зору своєчасності та узгодженості дій, оскільки будь-які зміни в розкладі або організаційних заходах мають оперативно доводитися до всіх зацікавлених сторін. Автоматизація таких процесів зменшує залежність від ручного управління та знижує ризик виникнення організаційних помилок.

З метою підвищення наочності та структурованості проєктних рішень було прийнято рішення використовувати модельно-орієнтований підхід до опису бізнес-процесів. Це дозволяє не лише описати функціонування системи на концептуальному рівні, а й забезпечити зрозумілу комунікацію між розробниками, замовниками та потенційними користувачами системи. Формалізований опис процесів також спрощує подальше вдосконалення та масштабування веб-системи.

Для графічного подання та аналізу бізнес-процесів обрано нотацію BPMN, яка є загальноприйнятим стандартом у сфері інженерії програмного забезпечення та управління процесами. Використання BPMN дозволяє чітко відобразити послідовність дій, точки прийняття рішень та взаємодію між учасниками процесу, що є важливим для коректної реалізації прикладної логіки системи та перевірки її відповідності реальним організаційним потребам.

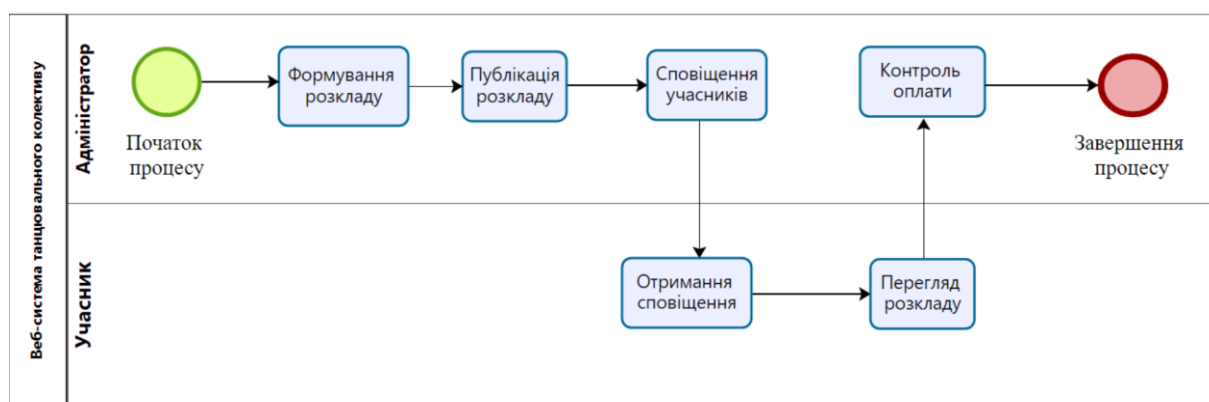


Рисунок 1 – BPMN-діаграма бізнес-процесу організації діяльності танцювального колективу

Функціонально система охоплює управління учасниками колективу, планування репетицій і заходів, а також інформаційну взаємодію між керівником і учасниками. Завдяки використанню єдиного інформаційного середовища зменшується кількість організаційних помилок та підвищується ефективність повсякденної роботи колективу. Система забезпечує централізоване зберігання розкладу, автоматизоване сповіщення учасників про організаційні зміни та підтримує оперативний контроль виконання основних процесів. Практичне значення роботи полягає у можливості використання розробленої веб-системи в реальній діяльності танцювальних та інших творчих колективів з метою підвищення ефективності їх організаційної роботи.

Крім того, запропонована веб-система може бути адаптована до специфіки різних типів колективів шляхом налаштування ролей користувачів, розкладів і сценаріїв взаємодії, що робить її універсальним інструментом організаційного управління. Модульна архітектура та використання веб-технологій створюють передумови для подальшого розширення функціоналу, зокрема інтеграції з календарними сервісами, мобільними застосунками або системами сповіщення, що підвищує практичну цінність і гнучкість розробленого рішення.

Висновки

У роботі розглянуто проблему організації діяльності танцювального колективу та запропоновано веб-систему, яка дозволяє впорядкувати основні організаційні процеси. Запропоноване рішення спрощує управління інформацією, покращує комунікацію між керівником і учасниками колективу та створює зручні умови для планування репетицій і заходів. Розроблена веб-система може бути використана як основа для подальшого розвитку та адаптації до потреб різних творчих колективів.

Список використаних джерел

1. Dumas M., La Rosa M., Mendling J., Reijers H. A. Fundamentals of Business Process Management. – 2nd ed. – Berlin : Springer, 2018. –527 p.
2. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. – Sebastopol : O'Reilly Media, 2020. – 419 p.

ВЕБ-ЗАСТОСУНОК ДЛЯ ПІДТРИМКИ ПРОЦЕСІВ ОРЕНДИ ЖИТЛА

Балюх В.І.¹⁾, Крепич С.Я.²⁾, Співак І.Я.³⁾

Західноукраїнський національний університет

^{1) бакалавр; 2-3) к.т.н., доцент}

I. Постановка проблеми

Ринок житлової нерухомості є стратегічно важливою складовою національної економіки, що забезпечує одну з базових фізіологічних та соціальних потреб людини – потребу в житлі. В умовах сучасної урбанізації, зростання мобільності трудових ресурсів та внутрішньої міграції населення, оренда житла стає все більш популярною та економічно виправданою альтернативою його купівлі.

Проте, детальний аналіз предметної області свідчить, що на сьогодні цей процес характеризується високим рівнем невизначеності, інформаційною асиметрією та комунікаційними бар'єрами. Традиційна схема взаємодії, що часто відбувається через дошки оголошень або соціальні мережі, має низку критичних недоліків для обох сторін угоди. Для орендодавців основними проблемами є складність керування вхідними дзвінками, ризику натрапити на недобросовісних платників та часові витрати на "пусті" покази житла. Для орендарів ключовими "больовими точками" є ризик шахрайства, втрата часу на перегляд варіантів, що не відповідають опису, та складність комунікації, коли доводиться повторювати один і той самий запит багатьом власникам.

Аналіз існуючих аналогів (Airbnb, Booking, OLX) показав, що більшість популярних сервісів мають суттєві обмеження. Сервіси бронювання орієнтовані переважно на короткострокову оренду, а класичні дошки оголошень не забезпечують повноцінної двосторонньої взаємодії. Критичним недоліком є те, що орендарі не мають змоги подати структуровану заявку зі своїми побажаннями, а орендодавці - зручно керувати цими заявками та обирати кандидатів.

II. Мета роботи

Метою роботи є створення спеціалізованого веб-застосунку, який дозволить реалізувати прозорий процес оренди, де обидві сторони зможуть взаємодіяти напряму. Система має спростити процес від пошуку квартир до укладання угоди, забезпечуючи управління заявками, внутрішню комунікацію.

III. Основна частина

Для досягнення поставленої мети було розроблено архітектуру системи та діаграму варіантів використання (Use Case Diagram), яка чітко розмежовує права та можливості користувачів.

У системі реалізовано рольову модель доступу, що включає чотири типи користувачів:

1. Гість - має базовий доступ до пошуку та фільтрації оголошень, а також перегляду об'єктів на карті для оцінки інфраструктури.
2. Орендар - авторизований користувач, який отримує можливість подавати заявки на оренду, вести перемовини через внутрішній чат та залишати відгуки.
3. Орендодавець - користувач, який керує своїми об'єктами нерухомості, розглядає вхідні заявки від кандидатів та приймає рішення щодо укладання угоди.
4. Адміністратор - забезпечує модерацію контенту та вирішення конфліктних ситуацій, що є критично важливим для безпеки платформи.

Ключовий бізнес-процес: Обробка заявок. Основною інновацією розробки є формалізація процесу укладання угоди, який складається з таких етапів:

1. Ініціація: Орендар замість телефонного дзвінка формує структуровану заявку, вказуючи бажані дати, кількість осіб та наявність тварин.
2. Розгляд: Система створює запис зі статусом *Pending*. Орендодавець отримує сповіщення та переглядає профіль кандидата, його історію та відгуки.
3. Прийняття рішення: Власник може схвалити (*Approved*) або відхилити (*Rejected*) заявку. Лише у випадку схвалення сторонам відкриваються контактні дані для завершення угоди.

Такий підхід вирішує проблему інформаційної асиметрії, оскільки обидві сторони отримують доступ до історії та репутації один одного ще до початку переговорів.

Нефункціональні вимоги та надійність. Окрім функціоналу, при проектуванні значну увагу приділено нефункціональним вимогам, які забезпечують якість програмного продукту.

1. Зручність використання (*Usability*): Інтерфейс спроектовано інтуїтивно зрозумілим для широкого кола користувачів з адаптивним дизайном для мобільних пристроїв.

- Надійність даних: Забезпечується валідація вхідних даних на етапі створення оголошення та подачі заявки, що гарантує точність інформації в базі даних.
- Безпека: Реалізовано розмежування доступу та захист персональних даних користувачів. Модерація оголошень адміністратором дозволяє відсіювати шахрайський контент.

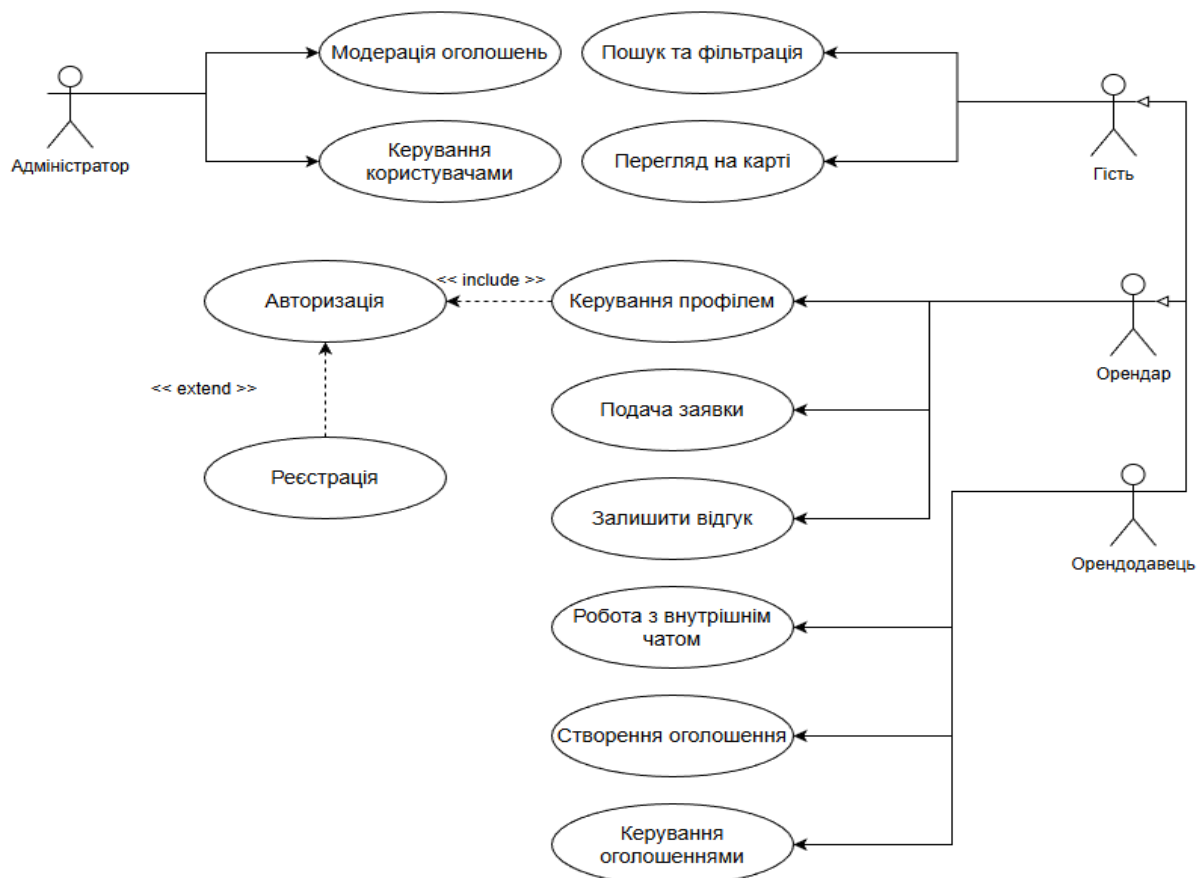


Рисунок 1 – Діаграма варіантів використання системи

Висновки

У ході виконання роботи було проведено комплексний аналіз ринку оренди житлової нерухомості та виявлено ключові недоліки існуючих рішень, зокрема відсутність інструментів для двосторонньої верифікації та керування заявками. Розроблений веб-застосунок пропонує ефективне вирішення цих проблем завдяки впровадженню:

- Систематизації даних: Створення єдиного каталогу з уніфікованими параметрами оголошень значно спрощує пошук релевантних варіантів.
- Автоматизації комунікації: Перехід від телефонних дзвінків до структурованих цифрових заявок економить час обох сторін та знижує психологічне навантаження.
- Системи репутації: Можливість залишати відгуки та формувати рейтинг учасників підвищує довіру до сервісу та сприяє побудові цивілізованих відносин на ринку.

Практична цінність роботи полягає у створенні повністю функціонального інструменту, який може бути використаний як альтернатива існуючим дошкам оголошень, забезпечуючи вищий рівень безпеки та комфорту для користувачів. Подальший розвиток системи передбачає інтеграцію з платіжними шлюзами для безпечних розрахунків та впровадження алгоритмів штучного інтелекту для персоналізованих рекомендацій житла.

Список використаних джерел

- Трофименко О. Г., Козін О. Б., Задерейко О. В. Веб-технології та веб-дизайн : навч. посіб. Одеса: Фенікс, 2019. 284 с.
- Крепич С.Я., Співак І.Я. Якість програмного забезпечення та тестування: базовий курс. Тернопіль: ФОП Паляниця В.А., 2020. – 478с.
- S.Krepych, I.Spivak, Improvement of SVD algorithm to increase the efficiency of recommendation systems, Advanced Information Systems, 5(4), 2021. pp.55-59
- Krepych S., Spivak I. and Spivak S., Parametric identification of static systems by methods of interval data analysis using parallel calculations, International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2021, pp.189-193

ІНТЕРАКТИВНИЙ СИМУЛЯТОР ЕЛЕКТРОННИХ СХЕМ ДЛЯ НАВЧАЛЬНИХ ЗАВДАНЬ

Фанта В.В.

*Західноукраїнський національний університет
магістрант*

I. Постановка проблеми

Опанування принципів роботи компонентів електронних схем вимагає поєднання теоретичних знань із практичними навичками. Традиційні лабораторні роботи потребують спеціалізованого обладнання, що обмежує доступ студентів до практики. Крім того, моделювальні середовища на кшталт Proteus чи Multisim мають високий поріг входу та орієнтовані переважно на професійне проектування[1].

Тому актуальною є розробка інтерактивних систем, які дозволяють у спрощеній формі моделювати роботу електронних компонентів із мінімальними технічними вимогами та вдосконаленою наочністю. Особливий інтерес становлять програмні засоби, що дозволяють поетапно виконувати навчальні завдання, аналізувати роботу схем та візуально відстежувати проходження сигналу.

II. Мета роботи

Метою роботи є створення інтерактивного симулятора електронних схем у середовищі Unity[2], який дає можливість візуально моделювати принцип роботи основних компонентів мікроелектроніки та виконувати навчальні завдання з електротехніки.

III. Особливості програмної реалізації

На відміну від існуючих програмних засобів моделювання електроніки, таких як Proteus або Multisim, запропонований симулятор поєднує автоматичне формування з'єднань між компонентами на основі просторової сітки, спрощену модель електричних і теплових процесів та ігровий підхід до подання навчальних завдань.

У розробленому симуляторі схема представлена у вигляді двовимірної сітки, де кожна клітинка може містити компонент: резистор, джерело живлення, діод, світлодіод, вимикач, провідник, котушка, транзистор, реле. З'єднання між компонентами формуються автоматично відповідно до їх просторового розташування, що дозволяє уникнути проблеми маршрутизації та перетину доріжок[2].

Система реалізує спрощену модель електричного кола постійного струму[3]. Проходження сигналу обчислюється на основі обходу графа суміжності. Загальний опір кола визначається як сума послідовно з'єднаних резисторів, після чого струм розраховується за формулою $I = U / \sum R$.

Оскільки параметри елементів обираються користувачем, варто враховувати можливі помилки, тому окремо вираховується потужність на кожному елементі $P = I^2 \cdot R$ і температура за спрощеною формулою $\Delta T = \alpha \cdot P \cdot \Delta t$.

Діоди та світлодіоди мають визначену полярність: проходження струму можливе лише у напрямку «анод → катод». У разі правильного складання схеми LED переходить у стан «увімкнено», що слугує індикатором успішності виконання навчального завдання. Кожне наступне завдання вимагає використання нового компоненту. Перелік компонентів симулятора та їхні параметри представлені в таблиці 1.

Таблиця 1

Основні компоненти симулятора та їх функціональні особливості

Компонент	Призначення	Налаштовувані параметри (область визначення)	Робочі обмеження
Джерело живлення (Battery / PSU)	Забезпечує напругу в колі	Напруга U : 1–24 В Внутрішній опір $R_{вн}$: 0–2 Ом Ємність акумулятора: 500–5000 мА·год	Макс. струм навантаження: 1–3 А

Резистор	Обмежує струм, формує подільники напруги	Опір R : 1–10 000 Ом	Робоча потужність: 0.125–1 Вт (задається користувачем)
Діод (звичайний)	Одностороннє пропускання струму	Тип: 1N4148 / 1N4007	Макс. струм: 100–1000 мА Зворотня напруга: 50–1000 В
Вимикач (Switch)	Розмикає/змикає коло	Немає параметрів	Макс. струм: 0.5–3 А
Провідник	З'єднання компонентів	Довжина проводу (кількість сегментів)	Опір на одиницю довжини: 0.01–0.05 Ом
Конденсатор	Накопичення заряду, фільтрація	Ємність C : 1 нФ – 4700 мкФ Макс. напруга: 5–50 В	Струм заряду/розряду обмежений серіальним опором (ESR)
Транзистор (NPN/PNP)	Керування струмом, підсилення	Тип: NPN/PNPβ (коэф. підсилення): 20–300 U_{be} : 0.6–0.7 В	Макс. струм колектора: 0.1–5 А
Польовий транзистор (MOSFET)	Керування великими струмами	U : 2–10 В R_{dsON} : 10–200 мОм	Макс. струм: 5–30 А
Реле	Галванічна розв'язка та перемикач	Напруга котушки: 5–12 В	Макс. струм контактів: 1–10 А

Запропонований набір компонентів дозволяє формувати широкий спектр навчальних завдань – від простих послідовних кіл до моделювання елементарних фільтрів, транзисторних ключів та інших базових структур мікроелектроніки.

Архітектура симулятора побудована за модульним принципом і складається з інтерфейсу користувача, менеджера рівнів, моделі електричної схеми, модуля симуляції та блоку візуалізації станів компонентів (див.рис.1). Така структура дозволяє розділити логіку взаємодії користувача, обчислювальні процеси та відображення результатів, що спрощує розширення системи й додавання нових компонентів або навчальних сценаріїв без зміни базової архітектури.



Рисунок 1 – Архітектура інтерактивного симулятора електронних схем

Висновок

У роботі представлено підхід до створення інтерактивного симулятора електронних схем із використанням Unity. Реалізована система забезпечує наочну демонстрацію принципів роботи основних електронних компонентів, дозволяє виконувати навчальні завдання та може бути використана як засіб підтримки навчального процесу з дисциплін «Електротехніка», «Мікроелектроніка» чи «Основи схемотехніки».

Список використаних джерел

- Boylestad R., Nashelsky L. *Electronic Devices and Circuit Theory*. 12th ed. Pearson, 2020.
- Unity Technologies. *Unity Manual: Scripting API and UI Toolkit Documentation*. Unity, 2023.
- Nilsson J.W., Riedel S. *Electric Circuits*. 11th ed. Pearson, 2019.

ПРОЄКТУВАННЯ БАЗИ ДАНИХ СОЦІАЛЬНОЇ ПЛАТФОРМИ З АІ-МОДЕРАЦІЄЮ ДЛЯ ВИЯВЛЕННЯ ТА ФІЛЬТРАЦІЇ ТОКСИЧНИХ ПОВІДОМЛЕНЬ У СОЦІАЛЬНИХ МЕРЕЖАХ

Тимчишин В.С.¹⁾, Сіماشко І.В.²⁾

Західноукраїнський національний університет

¹⁾ доктор філософії, ст. викладач; ²⁾ магістрант

I. Постановка проблеми

Соціальні мережі є важливою складовою сучасної цифрової комунікації, проте зростання кількості користувачів і анонімність онлайн-середовища сприяють поширенню токсичної поведінки у вигляді агресивних, принизливих висловлювань та мови ворожнечі, що негативно впливає на якість взаємодії та психологічний стан людей[1].

Традиційні методи модерації, зокрема ручна перевірка та фільтрація за ключовими словами, є ресурсоемними й недостатньо ефективними через нездатність враховувати контекст, сарказм і завуальовані форми агресії. У зв'язку з цим актуальним є застосування інструментів штучного інтелекту, які дають змогу автоматизовано, масштабовано та з урахуванням змісту й емоційної тональності виявляти та фільтрувати токсичні повідомлення, забезпечуючи баланс між захистом користувачів і свободою онлайн-спілкування.

II. Мета роботи

Метою даної роботи є проектування та реалізація реляційної бази даних соціальної платформи, орієнтованої на зберігання, обробку та модерацію користувацького контенту з використанням АІ-моделі, інтегрованої через API, для автоматичного виявлення та фільтрації токсичних повідомлень. У межах дослідження передбачено розробку структури бази даних для підтримки основних соціальних взаємодій, реалізацію механізму попереднього аналізу повідомлень перед публікацією, а також забезпечення обробки запитів у реальному часі та гнучкого налаштування правил модерації. Запропонований підхід спрямований на підвищення якості онлайн-комунікації без необхідності розробки й підтримки власних моделей машинного навчання, що робить рішення придатним для впровадження у невеликих проєктах та стартапах.

III. Основна частина

Оснovoю зберігання та обробки даних у розробленій соціальній платформі становить реляційна база даних PostgreSQL, структура якої побудована з урахуванням типових взаємодій користувачів у соціальних мережах. Схема бази даних (рисунок 1) включає п'ять основних таблиць: users, tweets, comments, likes, які пов'язані між собою через зовнішні ключі та забезпечують цілісність даних[2, 3].

Таблиця users використовується для зберігання облікових записів користувачів і містить унікальний ідентифікатор, ім'я користувача, електронну адресу, зашифрований пароль, а також часові мітки створення та оновлення запису. Вона є базовою сутністю, на яку посилаються інші таблиці системи. Таблиця tweets призначена для збереження текстових повідомлень користувачів та містить поле body, ідентифікатор автора (user_id) і поле origin_id, яке дозволяє реалізувати механізм репостів або відповідей шляхом посилання на оригінальний запис.

Для забезпечення автоматизованої модерації контенту структура таблиці tweets передбачає можливість розширення додатковими атрибутами, пов'язаними з результатами аналізу повідомлень АІ-моделлю. Зокрема, на рівні бази даних може бути реалізовано збереження метаданих щодо тональності тексту, рівня токсичності або статусу перевірки повідомлення, що дозволяє відокремити етап аналізу від етапу публікації контенту. Такий підхід забезпечує гнучкість системи, спрощує аудит рішень модерації та створює передумови для подальшої аналітики й оптимізації правил фільтрації.

Використання реляційної моделі даних у поєднанні з АІ-модерацією також сприяє реалізації обробки повідомлень у режимі реального часу. Перед збереженням нового запису в таблиці tweets текст повідомлення передається до зовнішнього АІ-сервісу через API, після чого на основі отриманого результату приймається рішення щодо його публікації або блокування. Збереження відповідної інформації в базі даних дозволяє не лише контролювати поточний стан контенту, а й формувати історію перевірок, що є важливим для подальшого аналізу поведінки користувачів та вдосконалення механізмів модерації.

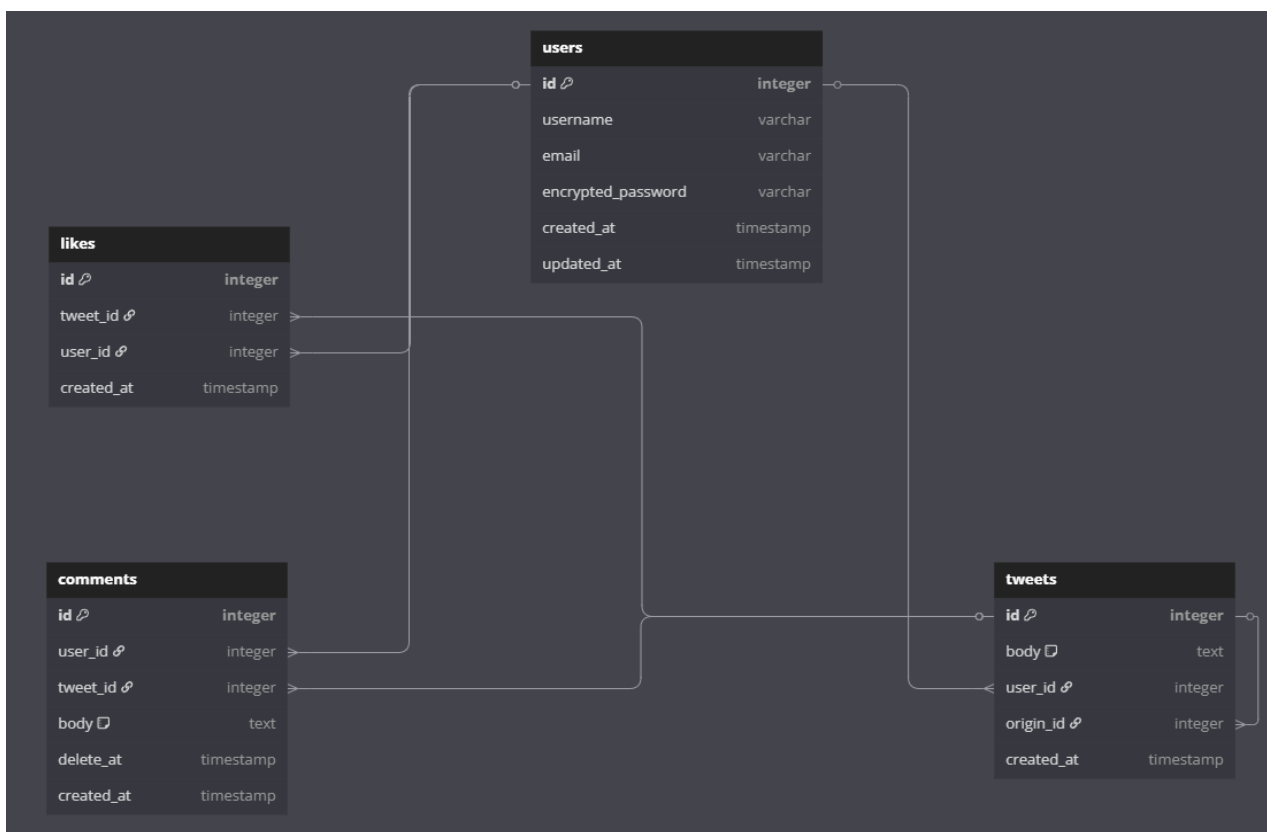


Рисунок 1 – Схема бази даних

Коментарі до повідомлень зберігаються в таблиці `comments`, яка пов'язана як з таблицею `users`, так і з таблицею `tweets`. Наявність поля `delete_at` дозволяє реалізувати логічне видалення коментарів без фізичного видалення записів із бази даних, що є важливим для модерації та аудиту. Реакції користувачів у вигляді вподобань реалізовані через таблицю `likes`, яка фіксує зв'язок між користувачем і повідомленням, забезпечуючи можливість підрахунку популярності контенту.

Запропонована структура бази даних є масштабованою та дозволяє ефективно реалізувати механізми модерації контенту, зокрема аналіз токсичності повідомлень перед їх публікацією. Чітке розділення сутностей і використання зовнішніх ключів спрощує розширення функціоналу системи, наприклад додавання нових типів взаємодії або збереження результатів автоматичної перевірки повідомлень.

Висновок

У ході роботи було спроектовано та описано реляційну базу даних соціальної платформи, орієнтованої на зберігання, обробку та модерацію користувачького контенту з використанням AI-моделі, інтегрованої через API. Запропонована структура бази даних на основі PostgreSQL охоплює ключові сутності соціальної взаємодії користувачів та забезпечує цілісність і узгодженість даних за рахунок використання зовнішніх ключів і чіткого розділення логічних компонентів системи.

Запропоноване рішення є масштабованим і придатним для розширення функціоналу, зокрема шляхом додавання нових типів взаємодії, збереження метаданих AI-аналізу та впровадження аналітичних інструментів для оцінювання токсичності контенту. Поєднання реляційної моделі даних із зовнішніми AI-сервісами дає змогу реалізувати ефективну модерацію в режимі реального часу без необхідності розробки та підтримки власних моделей машинного навчання.

Список використаних джерел

1. "Дослідження впливу соціальних мереж на наше життя та мислення" [Електронний ресурс] – Режим доступу: https://drukarnia.com.ua/articles/doslidzhennya-vplivu-socialnikh-merezh-na-nashe-zhittya-ta-mislennya-013_6
2. OpenAI API Documentation [Електронний ресурс] – Режим доступу: <https://platform.openai.com/docs/>
3. PostgreSQL Documentation [Електронний ресурс] – Режим доступу: <https://www.postgresql.org/docs/>

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ ВЕБ-ДОДАТКУ ДЛЯ УПРАВЛІННЯ ОСОБИСТИМИ ФІНАНСАМИ

Палійчук Х.М.¹⁾, Тимчишин В.С.²⁾

Західноукраїнський національний університет

¹⁾бакалавр; ²⁾ доктор філософії, ст. викладач

I. Постановка проблеми

У сучасних умовах цифровізації фінансових процесів питання ефективного управління особистими фінансами набуває особливої актуальності. Щоденні витрати, регулярні доходи, заощадження та фінансові цілі потребують систематичного обліку й аналізу. Більшість користувачів або ведуть фінансові записи вручну, або використовують сторонні мобільні застосунки, які часто мають обмежений функціонал, складний інтерфейс або не враховують індивідуальні потреби користувачів.

Окрім цього, значна частина існуючих сервісів є платними або не забезпечує достатній рівень прозорості та контролю над фінансовими даними. Відсутність універсального веб-рішення, доступного з будь-якого пристрою без прив'язки до конкретної платформи, ускладнює процес фінансового планування та аналізу. Це зумовлює необхідність створення веб-додатку, який би забезпечував зручний, надійний і безпечний інструмент для управління особистими фінансами.

II. Мета роботи

Метою даної роботи є проєктування та реалізація веб-системи управління особистими фінансами, яка забезпечує зручний облік доходів і витрат, аналіз фінансових даних та планування бюджету з урахуванням індивідуальних потреб користувачів, а також надійне збереження інформації й доступ до функціоналу через веб-браузер незалежно від платформи використання. Додатково система орієнтована на застосування підходів до проєктування програмного забезпечення з метою забезпечення масштабованості, безпеки та можливості подальшого розширення функціоналу.

III. Основна частина

Для наочного представлення функціональних можливостей розроблюваної системи та взаємодії користувачів із веб-додатком у роботі використано діаграму варіантів використання (див.рис.1). Діаграма варіантів використання дозволяє визначити межі системи, ролі користувачів та їхні права доступу, а також слугує основою для формування функціональних вимог і подальшого проєктування архітектури веб-додатку. На діаграмі прецедентів відображено основні сценарії взаємодії користувача з системою, зокрема реєстрацію, авторизацію, додавання фінансових записів, перегляд статистики та планування бюджету.

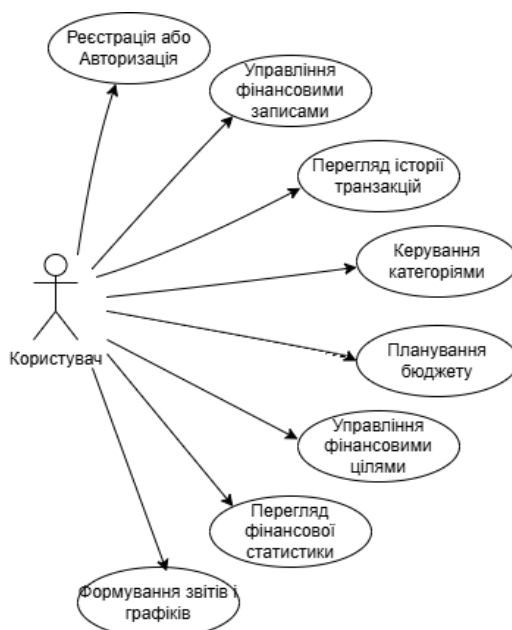


Рисунок 1 – Діаграма варіантів використання веб-додатку для управління особистими фінансами

Дана діаграма дозволяє візуально продемонструвати функціональні можливості системи та взаємозв'язок між користувачем і веб-додатком.

Розробка веб-додатку передбачає використання сучасних веб-технологій та клієнт-серверної архітектури. Клієнтська частина реалізується у вигляді односторінкового веб-додатку що забезпечує динамічну взаємодію з користувачем без перезавантаження сторінок та підвищує зручність користування системою. Навігація між основними розділами додатку здійснюється за допомогою маршрутизації на стороні клієнта[1].

Серверна частина додатку забезпечує обробку запитів клієнта, збереження та доступ до фінансових даних користувачів і реалізується у вигляді програмного інтерфейсу прикладного програмування. Для зберігання даних використовується реляційна база даних, що дозволяє забезпечити цілісність інформації, структурованість даних та ефективну роботу з фінансовими записами[2].

На діаграмі класів (дис.рис.2) зображено структуру веб-системи управління особистими фінансами та взаємозв'язки між її основними сутностями. Центральним елементом є клас User, який зберігає облікові дані користувача та пов'язаний із фінансовими об'єктами системи. Користувач може мати кілька Transaction, що відображають доходи й витрати та асоціюються з відповідними Category, які визначають тип фінансової операції. Для контролю витрат передбачено клас Budget, що встановлює ліміти на певний період, а також клас FinancialGoal, який використовується для постановки та відстеження фінансових цілей користувача. Клас Report забезпечує формування узагальненої фінансової інформації за вибраний період.

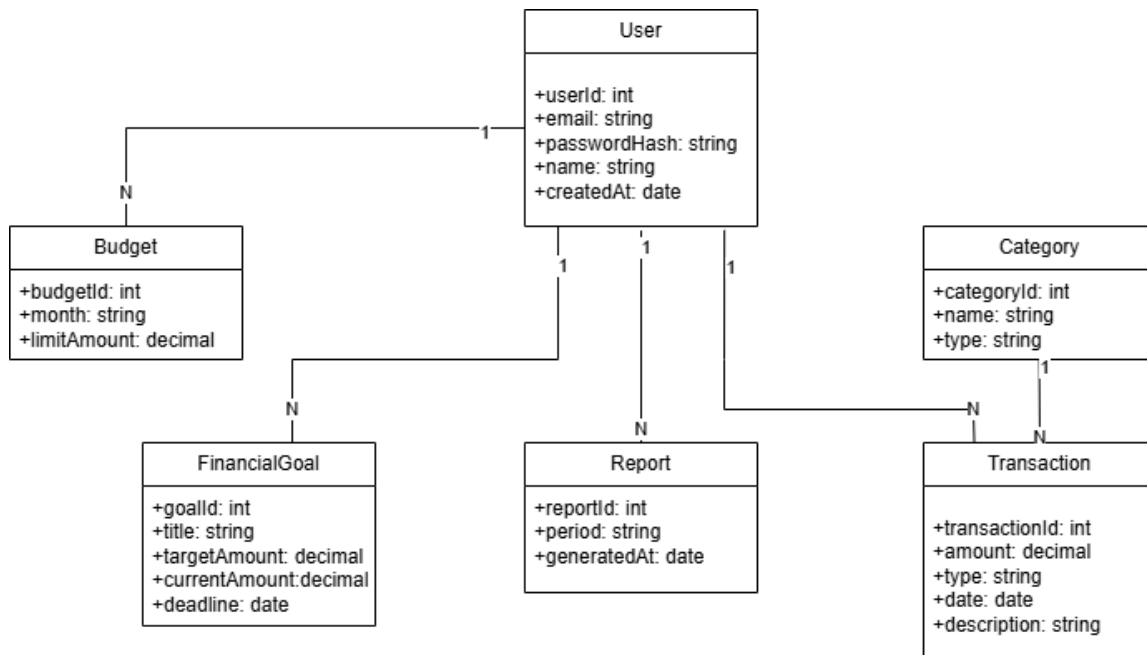


Рисунок 2 – Діаграма класів системи управління особистими фінансами

Зв'язки типу «один-до-багатьох» між користувачем і фінансовими сутностями відображають логіку предметної області та забезпечують цілісність і структурованість даних у системі.

Висновок

У межах даної роботи було розроблено концепцію веб-додатку для управління особистими фінансами, що дозволяє користувачам ефективно контролювати власні доходи та витрати, аналізувати фінансові дані та планувати бюджет. Запропоноване рішення забезпечує зручний доступ до функціоналу, структуроване збереження даних і наочне представлення фінансової інформації.

У подальшому система може бути розширена шляхом впровадження додаткових аналітичних інструментів, підтримки мультивалютності, інтеграції з банківськими сервісами та створення мобільної версії застосунку.

Список використаних джерел

1. Sommerville I. Software Engineering. – 10th ed. – Boston : Pearson Education, 2021. – 816 p.
2. Richards M., Ford N. Fundamentals of Software Architecture: An Engineering Approach. – Sebastopol : O'Reilly Media, 2020. – 419 p.

АРХІТЕКТУРА ПРОГРАМНОЇ СИСТЕМИ ПОЄДНАННЯ АРТ-ТЕРАПІЇ ТА ФІЗИЧНОЇ РЕАБІЛІТАЦІЇ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

Цапів Я.А.

Західноукраїнський національний університет
аспірант

I. Постановка проблеми

Арт-терапія є новим методом психотерапії, вона є корисним втручанням із зростаючою доказовою базою, яка підтримує її використання для різних груп пацієнтів[1]. Потреба у фізичній реабілітації є актуальною у цілому світі, а особливо в Україні через наслідки воєнних дій. За даними МОЗ України [2], в Україні за 2023 на 16% збільшилась кількість інсультів, при виникненні яких з'являються порушення рухливої діяльності кінцівок. Традиційні методи реабілітації, такі як механічні вправи чи мануальна терапія, часто є неефективними через брак мотивації пацієнтів та труднощі в точному моніторингу прогресу [3]. Останнім часом в арт-терапії розвиваються та впроваджуються нові технології, такі як віртуальна та доповнена реальність [4-5].

II. Мета роботи

Метою даної праці є розробка архітектури та основних функцій програмної системи підтримки процесів проведення арт-терапії з використанням засобів та методів доповненої реальності.

III. Архітектура програмної системи

Розглянуто комплексну мультисервісну архітектуру, що включає чотири ключові компоненти: програмний модуль для MagicLeap 2, серверну інфраструктуру з базою даних, хмарні вебсервіси Webex, а також інструменти, які використовує терапевт — зокрема пристрій для формування голограми та робочий застосунок на персональному комп'ютері (див.рис.1).

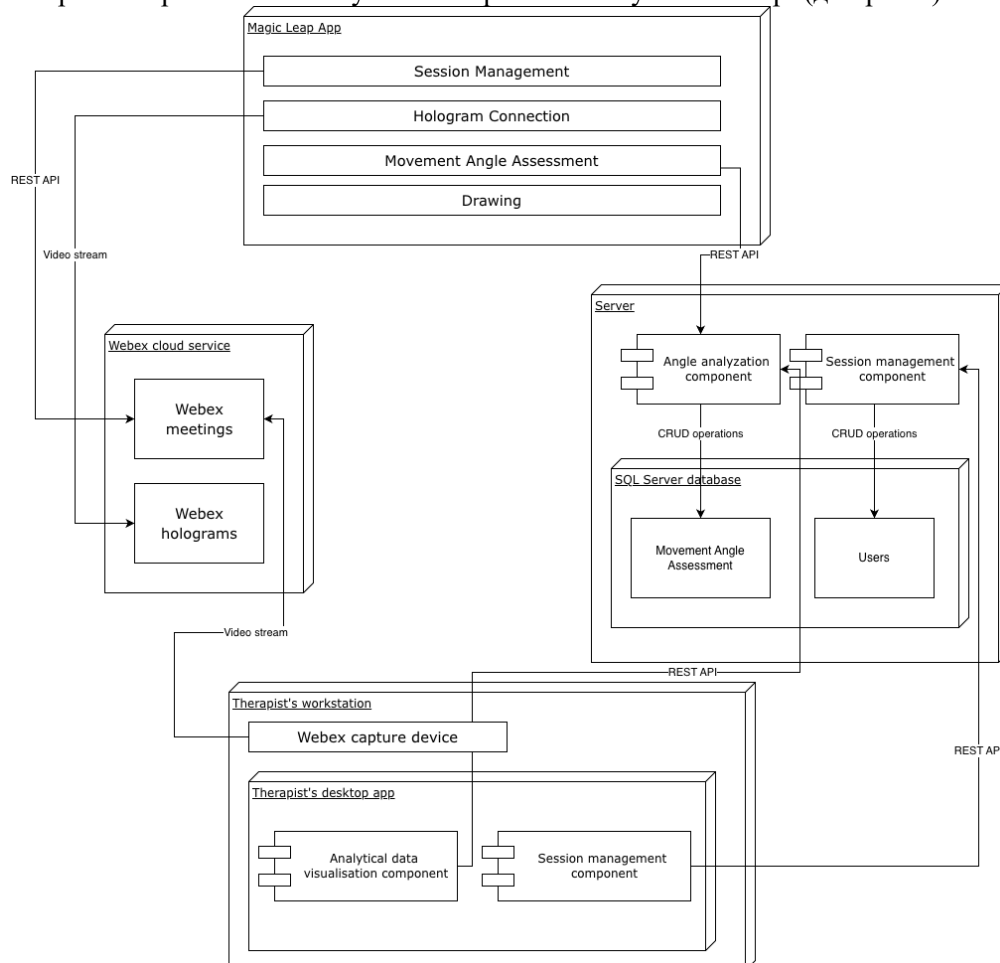


Рисунок 1 – Діаграма компонентів програмної системи

Клієнтський застосунок для MagicLeap2 відображає сформований на сервері сценарій та надає користувачеві інструменти для виконання поставлених завдань. У процесі роботи застосунок здійснює вимірювання та накопичення даних щодо положення кистей та пальців пацієнта, після чого ці дані передаються на спеціалізований сервер для подальшої обробки та аналізу.

Функціонально застосунок складається з таких модулів:

- SessionManagement- відповідає за авторизацію користувача та організацію комунікації між пацієнтом і терапевтом;
- HologramConnection- забезпечує відображення Webex-голограми терапевта та керування її параметрами;
- MovementAngleAssessment- реалізує автоматичне вимірювання кутів згину суглобів та визначення положення верхніх кінцівок пацієнта;
- Drawing- надає інструменти для арт-терапії, включаючи вибір кольорів, середовища, інструментів малювання тощо.

Для реалізації застосунку використано наступний технологічний стек:

C#, UnityHub, UnityEditor, MagicLeapHub, MagicLeap SDK, Unity3D, WebRTC, FishNet + Dissonance, UltimateXR (або Unity XR), MagicLib, SqlDBClient.

Модуль WebexCloudService забезпечує взаємодію між пацієнтом і терапевтом шляхом організації сеансу комунікації з використанням фотореалістичної голограми терапевта. Для реалізації даного функціоналу застосовуються два компоненти платформи Webex: WebexMeetings та WebexHolograms. Компонент WebexMeetings надає REST API для створення онлайн-сеансів та підтримання каналу зв'язку між пацієнтом і терапевтом. Компонент WebexHolograms забезпечує передачу та відтворення фотореалістичної голограми терапевта у середовищі доповненої реальності.

Робоче місце терапевта надає можливість формувати середовище реабілітаційного сеансу, задавати позиції та набір інструментів, а також визначати додаткові умови проходження терапії. Отримані під час сесії дані відображаються у вигляді клієнтської картки у спеціальному розділі інтерфейсу. Робоче місце терапевта складається з двох основних компонентів: WebexCaptureDevice та DesktopApplication.

Компонент WebexCaptureDevice здійснює захоплення зображення терапевта для формування фотореалістичної голограми.

Компонент DesktopApplication забезпечує керування обліковими записами пацієнтів, організацію терапевтичних сесій та відображення їх результатів, зокрема аналітичних показників і динаміки відновлення.

Для реалізації робочого місця використано наступний технологічний стек:

Unity3D, WebRTC, FishNet + Dissonance, UltimateXR (або Unity XR), MagicLib, SqlDBClient, Webex.

Висновок

У цій роботі представлено архітектуру та ключові модулі програмної системи, призначеної для проведення арт-терапії з використанням технологій доповненої реальності. Запропонована система вводить інноваційні підходи у процес психологічної та фізичної реабілітації, покращуючи механізми моніторингу та оцінювання ефективності лікування. Архітектурні рішення забезпечують гнучкість щодо подальшого розширення й модифікації, що дозволяє застосовувати систему як базовий реабілітаційний інструмент або як додатковий засіб терапії. Впровадження такого підходу може суттєво підсилити існуючі методики арт-терапії та фізичної реабілітації верхніх кінцівок, розширюючи їх функціональні можливості та підвищуючи ефективність.

Список використаних джерел

1. Chelsea Cheng, Mohamed Elhassan Elamin, Helen Mayand Miriam Kennedy "Drawing on emotions: the evolving role of art therapy" Published on line by Cambridge University Press: April 19, 2021, doi.org/10.1017/ipm.2021.20
2. МОЗ підготувало пропозиції до плану заходів в межах Ukrainian Facility від ЄС: <https://moz.gov.ua/uk/moz-pidgotuvalo-propozicii-do-planu-zahodiv-v-mezhah-ukrainian-facility-vid-es->.
3. Sang S Pak, Dora Janela, Nina Freitas, Fabíola Costa, Robert Moulder, Maria Molinos, Anabela C Areias, Virgílio Bento, Steven P Cohen, Vijay Yanamadala, Richard B Souza, Fernando Dias Correia "Comparing Digital to Conventional Physical Therapy for Chronic Shoulder Pain: Randomized Controlled Trial", PMID: 37490337, PMCID: PMC10474513, DOI: 10.2196/49236.
4. Shamri Zeevi L "Making Art Therapy Virtual: Integrating Virtual Reality Into Art Therapy With Adolescents.". doi: 10.3389/fpsyg.2021.584943
5. Christos Hadjipanayi, Domna Banakou, Despina "Michael-Grigoriou. Art as therapy in virtual reality: A scoping review". DOI:10.3389/frvir.2023.1065863

Наукове видання

Комп'ютерні інформаційні технології

Матеріали
зимової школи-семінару молодих вчених і
студентів СІТ'2025

Відповідальний за випуск:

Пукас А.В., д.т.н., професор,
завідувач кафедри комп'ютерних наук
Західноукраїнського національного університету

Підписано до друку 28.11.2025р.
Формат 60x84/16. Папір офсетний.
Друк офсетний. Зам. № 9-365
Тираж 25 прим.

Віддруковано ФО-П Шпак В. Б.
Свідоцтво про державну реєстрацію В02 № 924434 від 11.12.2006 р.
Свідоцтво платника податку: Серія Е № 897220
м. Тернопіль, вул. Просвіти, 6.
тел. 8 097 299 38 99
E-mail: tooums@ukr.net