

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

на тему: «Використання ігрових підходів в неігровому просторі»

ЦИВІНА ІГОР ВОЛОДИМИРОВИЧ

(прізвище, ім'я та по батькові студента повністю)

Київ, 2024 р.

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ

автоматизації і інформаційних технологій

(факультет)

інформаційних технологій

(кафедра)

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

„___” _____ 2024 року

ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»

на тему: «Використання ігрових підходів в неігровому просторі»

Виконав: студент 4-го курсу, групи КН-20-1 _____

Спеціальності: 122 «Комп'ютерні науки _____

Освітня програма: «Інформаційні управляючі
системи та технології» _____

(шифр і назва напрямку підготовки, спеціальності)

Цивіна І.В.

(прізвище та ініціали)

Керівник _____ Вацкель В.Ю.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ, 2024 р.

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
БУДІВНИЦТВА І АРХІТЕКТУРИ**

Факультет: автоматизації і інформаційних технологій

Кафедра: інформаційних технологій

Освітній рівень: «бакалавр» за ОП

Спеціальність: 122 «Комп'ютерні науки»

Освітня програма: Інформаційні управляючі системи і технології

ЗАТВЕРДЖУЮ

Завідувач кафедри ІТ

к.т.н., доцент Гончаренко Т.А.

„___” _____ 2024 року

**З А В Д А Н Н Я
ДО ВИКОНАННЯ АТЕСТАЦІЙНОЇ ВИПУСКНОЇ РОБОТИ
НА ЗДОБУТТЯ ОСВІТНЬОГО РІВНЯ «БАКАЛАВР»**

Цивіна Ігор Володимирович

Тема роботи: Використання ігрових підходів у неігровому просторі

затверджена наказом ректора КНУБА № 433/2 від 29.02.2024.

2. Керівник роботи: Вацкель Володимир Юрійович, асистент кафедри
інформаційних технологій

3. Строк подання студентом роботи до захисту: _____

4. Зміст пояснювальної записки за розділами:

P.1. Аналіз предметної області та постановка задачі

P.2. Алгоритмічне та математичне пояснення

P.3. Розробка програмного забезпечення

P.4. Ергономіка інформаційних технологій

5. Інформаційні слайди:

C.1. Мета та актуальність теми

C.2. Теоретичні основи

C.3. Аналіз предметної області

- C.4. Методологія розробки _____
- C.5. Збір вимог _____
- C.6. Проектування архітектури _____
- C.7. Розробка та тестування _____

6. Календарний план виконання атестаційної випускної роботи

Види робіт та їх зміст	Дата виконання
P. 1. Аналіз предметної області та постановка задачі	Січень 2024 р.
P. 2. Алгоритмічне та математичне пояснення	Лютий 2024 р.
P. 3. Розробка програмного забезпечення	Березень 2024 р.
Тестовий приклад програми	Квітень 2024 р.
P. 4. Ергономіка інформаційних технологій	Травень 2024 р.
Остаточне оформлення роботи	Травень 2024 р.
Направлення роботи на рецензування	Червень 2024 р.
Попередній захист роботи на кафедрі	Червень 2024 р.

7. Консультанти розділів атестаційної випускної роботи

Розділ	Прізвище, ініціали та посада консультанта, представника комісії	дата	підпис

8. Дата видачі завдання: 10.10.2023

Завідувач

Гончаренко Т.А.

(підпис) (прізвище та ініціали)

Керівник

Вацкель В.Ю.

(підпис) (прізвище та ініціали)

Студент

Цивіна І.В.

(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Цивіна І.В. Використання ігрових підходів в неігровому середовищі.

Атестаційна випускна робота бакалавра за спеціальністю: 122 «Комп'ютерні науки», спеціалізація: «Інформаційні управляючі системи і технології». – Київський національний університет будівництва та архітектури. – Київ, 2024.

Робота присвячена розробці мобільного додатку EchoAid, що призначений для організації та участі у волонтерських заходах. Особливістю додатку є використання ігрових елементів для підвищення мотивації та залучення користувачів. Розробка додатку включала проектування інтуїтивно зрозумілого та привабливого інтерфейсу, забезпечення швидкої реакції на дії користувачів та мінімізацію помилок.

Ключові слова: мобільний додаток, волонтерство, ігрові підходи, гейміфікація, інтерфейс користувача, EchoAid.

SUMMARY

Tsyvina I.V. Using Game Approaches in a Non-Gaming Environment.

Bachelor's thesis in the specialty: 122 "Computer Science", specialization: "Information managing systems and technologies". - Kyiv National University of Construction and Architecture. - Kyiv, 2024.

This thesis is devoted to the development of the EchoAid mobile application, designed to organize and participate in volunteer activities. A distinctive feature of the application is the use of game elements to increase user motivation and engagement. The development of the application included designing an intuitive and attractive interface, ensuring quick response to user actions, and minimizing errors.

Keywords: mobile application, volunteering, game approaches, gamification, user interface, EchoAid.

ЗМІСТ

ВСТУП	9
1. РОЗДІЛ 1. Аналітичний огляд предметної області	10
1.1. Вступ до розділу	10
1.1.1. Мета та актуальність теми	10
1.1.2. Об'єкт дослідження	11
1.1.3. Предмет дослідження	12
1.2. Теоритичні основи волонтерства та гейміфікації	13
1.2.1. Історія волонтерства	13
1.2.2. Роль гейміфікації у волонтерстві	16
1.3. Аналіз потреби у віртуальному волонтерстві	19
1.3.1. Соціально-економічний контекст	19
1.3.2. Переваги віртуального волонтерства	19
1.3.3. Підтримка військових та цивільних у кризових регіонах	20
1.3.4. Розширення освітніх можливостей та допомога тваринам	20
1.3.5. Вплив гейміфікації на волонтерську активність	21
1.4. Стан віртуального волонтерства в Україні	21
1.4.1. Розвиток та поточний стан	21
1.4.2. Волонтерські ініціативи та платформи	21
1.4.3. Вплив гейміфікації	23
1.5. Обґрунтування необхідності розробки нової платформи	23
1.6. Висновки до першого розділу	26
2. РОЗДІЛ 2. Алгоритмічне та математичне пояснення	29
2.1. Методологія розробки	29
2.1.1. Вибір методології	29
2.1.2. Принципи методології Agile	29
2.1.3. Обґрунтування вибору методології Agile	30
2.1.4. Реалізація Agile в проекті EchoAid	30
2.2. Збір вимог до проекту	32
2.2.1. Методи збору вимог	32
2.2.2. Документація вимог	33
2.2.3. Пріоритизація вимог	34
2.3. Проектування архітектури системи	35
2.3.1. Визначення архітектури	35

	7
2.3.2. Створення діаграм	36
2.4. Бібліотеки та фреймворки для розробки додатку	44
2.4.1. Android SDK	44
2.4.2. Retrofit	45
2.4.3. Room	45
2.4.4. Glide	46
2.4.5. Hilt	46
2.4.6. Navigation Component	47
2.5. Опис функціональних особливостей	48
2.5.1. Реєстрація користувача	48
2.5.2. Вхід в систему	49
2.5.3. Створення завдання	49
2.5.4. Перегляд завдань	50
2.5.5. Пошук завдань	51
2.5.6. Створення спільноти	51
2.5.7. Перегляд спільнот	52
2.5.8. Управління профілем	53
2.5.9. Публікація новин	53
2.5.10. Отримання рекомендацій завдань	54
2.5.11. Перегляд нагород	57
2.5.12. Вступ до спільноти	58
2.5.13. Вихід зі спільноти	58
2.5.14. Участь у завданні	59
3. РОЗДІЛ 3. Розробка програмного забезпечення	61
3.1. Загальні положення	61
3.2. Вибір програмних засобів	61
3.3. Реалізація компонентів програмного забезпечення	63
3.3.1. Реєстрація користувача	63
3.3.2. Вхід в систему	64
3.3.3. Створення завдання	65
3.3.4. Управління профілем	67
3.4. Фінальна архітектура системи	68
3.4.1. Клієнтський додаток	68
3.4.2. Серверна частина	70

	8
3.4.3. База даних	72
3.5. Логіка роботи програмного забезпечення	75
3.5.1. Алгоритм реєстрації користувачів	75
3.5.2. Алгоритм авторизації	75
3.5.3. Алгоритм створення завдання	75
3.5.4. Алгоритм перегляду завдань	75
3.5.5. Алгоритм управління профілем користувача	76
3.5.6. Алгоритм синхронізації даних	76
3.6. Інсталяція та налаштування	77
3.6.1. Інсталяція додатку через APK	77
3.6.2. Налаштування середовища виконання	77
3.6.3. Плани щодо розповсюдження через Google Play	78
3.7. Тестування програмного забезпечення	79
3.7.1. Процес тестування	79
3.7.2. Висновки за результатами тестування	80
4. РОЗДІЛ 4. Ергономіка інформаційних систем	82
4.1. Ергономіка ІТ	82
4.1.1. Вимоги до програмного забезпечення з точки зору користувача та проектування компонентів інтерфейсу	82
4.1.2. Параметри при розробці інтерфейсу користувача	83
4.2. Дизайн додатку	85
4.2.1. Загальні положення	85
4.2.2. Основні елементи дизайну	86
4.2.3. Особливості дизайну EchoAid	89
4.3. Техніко-економічне обґрунтування розробки	98
4.3.1. Резюме проекту	98
4.3.2. Опис проектованого продукту або виду послуг	99
4.3.3. Оцінка ринку збуту	100
4.3.4. Аналіз конкуренції	100
4.3.6. План виробництва	101
4.3.7. Організаційний та юридичний план	101
4.3.8. Фінансовий план та стратегія фінансування	101
4.3.9. Оцінка ризику та страхування	101
ВИСНОВКИ	103

ВСТУП

У сучасному світі, де технології проникають у всі сфери життя, використання ігрових підходів в неігровому середовищі стає все більш актуальним. Гейміфікація, тобто впровадження елементів гри в процеси, що не є ігровими, дозволяє значно підвищити мотивацію, залученість та ефективність користувачів. Однією з таких сфер є волонтерство, яке відіграє важливу роль у розвитку суспільства.

Однак, організація та участь у волонтерських заходах часто стикається з проблемами недостатньої залученості та мотивації волонтерів. Люди можуть не знаходити потрібної інформації, витратити багато часу на пошук відповідних заходів або втрачати інтерес через відсутність стимулів. Вирішенням цих проблем може стати мобільний додаток, який спрощує процес організації та участі у волонтерських заходах, а також підвищує мотивацію користувачів через гейміфікацію.

Метою даної роботи є розробка мобільного додатку EchoAid, що забезпечить ефективну організацію волонтерських заходів та підвищення залученості волонтерів. Додаток використовуватиме сучасні технології та методи дизайну, що забезпечать інтуїтивно зрозумілий інтерфейс, швидку реакцію на дії користувачів та мінімізацію помилок.

Основними завданнями роботи є дослідження потреб та очікувань користувачів, проектування інтерфейсу з урахуванням ергономічних принципів та сучасних тенденцій дизайну, розробка та тестування функціональних можливостей додатку, а також впровадження гейміфікаційних елементів для підвищення мотивації користувачів.

1. РОЗДІЛ 1. Аналітичний огляд предметної області

1.1. Вступ до розділу

1.1.1. Мета та актуальність теми

У сучасному світі, що стикається з безпрецедентними викликами, такими як війна в Україні, віртуальне волонтерство відіграє ключову роль у мобілізації ресурсів та координації допомоги. Цифрові технології дозволяють залучати широку аудиторію та забезпечувати підтримку навіть тоді, коли традиційні методи виявляються недостатньо ефективними або недоступними.

Метою цього дослідження є аналіз та оцінка ефективності гейміфікаційних¹ стратегій у контексті віртуального волонтерства, а також розробка рекомендацій щодо впровадження інноваційної платформи, яка б підвищила залученість і мотивацію волонтерів. Дослідження спрямоване на виявлення оптимальних механізмів інтеграції ігрових елементів у волонтерські проекти для досягнення максимального соціального впливу та підтримки довгострокової активності волонтерів.

Використання гейміфікації у віртуальному волонтерстві демонструє значний потенціал для підвищення ефективності та мотивації волонтерів. Дослідження показують, що гейміфікація може сприяти покращенню залученості та продуктивності волонтерів завдяки створенню відчуття досягнення та визнання їхніх зусиль. Наприклад, дослідження, проведене Kim et al., виявило, що студенти, які брали участь у гейміфікованих курсах, показали на 40% кращі результати порівняно з тими, хто навчався у традиційній формі [3-5].

Гейміфікація також сприяє створенню більш інтерактивного та захоплюючого досвіду для волонтерів, що допомагає утримувати їхню увагу та підтримувати високу продуктивність протягом тривалого часу. Важливо відзначити, що ігрові елементи, такі як бали, медалі, рівні та виклики, створюють стимули для активної

¹Гейміфікація – використання ігрових практик та механізмів у неігровому контексті для залучення кінцевих користувачів до розв'язання проблем

участі та сприяють підвищенню ефективності виконання завдань. За даними дослідження, проведеного Lampropoulos та Sidiropoulos, використання гейміфікації у навчальному середовищі значно покращило показники успішності, мотивації та залученості студентів порівняно з традиційними методами навчання [2].

Крім того, гейміфікація може забезпечити більш прозорий та відстежуваний процес волонтерської діяльності. Це особливо важливо для організацій, які працюють у кризових умовах, де швидка мобілізація ресурсів та координація дій є ключовими факторами успіху. Дослідження показують, що інтеграція гейміфікаційних елементів може сприяти кращій координації та співпраці між волонтерами, що є критично важливим для ефективного реагування на надзвичайні ситуації [4].

Таким чином, можемо зробити висновок, що впровадження гейміфікації у віртуальне волонтерство не лише підвищує ефективність мобілізації ресурсів, але й сприяє кращій координації, мотивації та утриманню волонтерів, що є важливим для забезпечення стійкості та продуктивності волонтерських програм у довгостроковій перспективі [5].

1.1.2. Об'єкт дослідження

Об'єктом дослідження у цій роботі є віртуальне волонтерство, яке є важливим інструментом сучасної соціальної взаємодії. Віртуальне волонтерство слугує платформою для мобілізації індивідуальних і колективних зусиль, дозволяючи учасникам з усього світу вносити свій вклад у вирішення глобальних соціальних, екологічних та гуманітарних проблем через інтернет. Ця форма волонтерства забезпечує ефективне реагування на потреби в умовах, коли традиційна допомога може бути обмежена часом або логістикою [1, 2].

В контексті триваючої війни в Україні та глобальних криз, віртуальне волонтерство виявилось незамінним у забезпеченні швидкої та ефективної підтримки військовим і цивільним особам. Воно допомагає забезпечити логістичну

підтримку, медичну допомогу, розподіл гуманітарних ресурсів та психологічну підтримку, а також відіграє ключову роль у відновленні і розвитку постраждалих спільнот.

Окрім цього, віртуальне волонтерство включає ряд інших ініціатив, які охоплюють ширший спектр діяльності, таких як розробка та реалізація онлайн-курсів та навчальних програм для підвищення обізнаності та кваліфікації у різних сферах, організація віртуальних кампаній на підтримку притулків, а також інформаційних програм про права та добробут тварин, підтримка місцевих проектів з розвитку спільнот, включаючи екологічні ініціативи та програми забезпечення стійкості спільнот. Віртуальне волонтерство також включає розробку та використання інноваційних технологій, таких як гейміфікація, для залучення та утримання волонтерів. Ці ініціативи значно підвищують гнучкість та доступність волонтерської діяльності, дозволяючи людям із різних куточків світу активно долучатися до вирішення глобальних та місцевих проблем [3-5].

1.1.3. Предмет дослідження

Предметом дослідження у цій дипломній роботі є гейміфікація в контексті віртуального волонтерства. Це дослідження фокусується на тому, як впровадження ігрових елементів — таких як бали, медалі, рівні, та виклики — може радикально трансформувати взаємодію волонтерів із діяльністю, роблячи процес не тільки більш захоплюючим, але й значно продуктивнішим. Гейміфікація волонтерства забезпечує створення стимулів для активної участі, що сприяє не лише залученню більшої кількості учасників, але й підтримці їхньої мотивації на тривалий період.

Механізми гейміфікації включають декілька цікавих елементів. По-перше, бали допомагають оцінити внесок кожного волонтера, створюючи здорову конкуренцію та надаючи відчуття визнання їхніх зусиль. Уявіть собі, як приємно отримати додаткові бали за виконане завдання і побачити, що ваш вклад оцінили. По-друге, медалі та

значки – це ті маленькі, але дуже важливі символи досягнень, які нагадують про ваші успіхи і зусилля. Вони служать видимими відзнаками за особливі вклади або за успішне завершення важливих завдань.

Рівні, в свою чергу, додають мотивації для подальшого розвитку та освоєння нових навичок. Кожен новий рівень відкриває перед волонтерами нові можливості та виклики, що робить процес волонтерства ще більш захоплюючим. І нарешті, виклики додають елемент захоплення та взаємодії. Вони змушують волонтерів долати складнощі та знаходити креативні рішення для вирішення завдань. Виклики роблять волонтерську діяльність не просто корисною, а й неймовірно цікавою!

Ефективність гейміфікаційних стратегій:

Аналіз ефективності гейміфікаційних стратегій включає оцінку того, як ці інструменти впливають на волонтерську діяльність. Важливо вивчити, як гейміфікація може підвищити залученість учасників, спонукати їх до регулярної участі, і надати їм додаткові стимули для здійснення соціальних дій. Спеціальна увага приділяється тому, як ці механізми можуть бути адаптовані для залучення молодих волонтерів, які часто шукають динамічні та інтерактивні форми участі, що дозволяють їм відчувати безпосередній вплив своїх дій.

Довогостроковій перспективі:

Особливо цікавим є дослідження впливу гейміфікаційних стратегій на довгострокову активність волонтерів. Вивчення таких підходів дозволяє розуміти, як постійне введення нових ігрових елементів може підтримувати інтерес та мотивацію волонтерів протягом тривалого часу, що є ключовим для успішної і стійкої волонтерської діяльності.

1.2. Теоритичні основи волонтерства та гейміфікації

1.2.1. Історія волонтерства

Давні часи

Волонтерство має глибоке коріння, що сягає давніх часів. У стародавніх суспільствах, таких як Єгипет, Греція та Рим, існували практики допомоги ближнім, які часто були пов'язані з релігійними або спільнотними зобов'язаннями. Наприклад, у Стародавньому Римі громадяни могли добровільно брати участь у будівництві громадських споруд та підтримці громадського порядку.

Середньовіччя

У середньовічний період волонтерство було тісно пов'язане з релігійними установами, такими як церкви та монастирі. Багато людей допомагали у наданні притулку та їжі для бідних і хворих. Християнська церква відігравала важливу роль у сприянні благодійності та організації допомоги тим, хто її потребував. Орден Госпітальєрів, заснований у 11-му столітті, є прикладом організації, яка забезпечувала медичну допомогу паломникам і хрестоносцям.

Новий час

З приходом Нового часу та розвитком громадянського суспільства волонтерство набуло нових форм. У 19-му столітті у Великобританії та інших країнах Європи почали з'являтися благодійні організації, що займалися боротьбою з бідністю та підтримкою знедолених. У цей період було засновано багато відомих організацій, таких як Червоний Хрест (1863 рік), яка стала моделлю для багатьох інших гуманітарних організацій.

20-е століття

20-е століття ознаменувалося значним розвитком волонтерського руху. Після Першої світової війни багато молодих людей брали участь у відновленні зруйнованих війною територій. У цей час також були створені такі організації, як Peace Corps (1961 рік) у США, яка надихала молодих людей на участь у міжнародних волонтерських проектах, спрямованих на допомогу країнам, що розвиваються.

Друга світова війна

Під час Другої світової війни волонтерство набуло нового значення. Мільйони людей у різних країнах допомагали у військових зусиллях, працюючи в лікарнях, на фабриках, у сільському господарстві та в різних благодійних організаціях. Після війни волонтерство відіграло важливу роль у відновленні зруйнованих спільнот і забезпеченні підтримки постраждалих.

Сучасність

У сучасному світі волонтерство продовжує розвиватися, з'являються нові форми та підходи. З розвитком інтернету та цифрових технологій волонтерство стало глобальним явищем. Тепер люди можуть брати участь у волонтерських проектах незалежно від свого місця проживання, використовуючи онлайн-платформи та інструменти для координації зусиль.

Волонтерство також стало важливим елементом корпоративної соціальної відповідальності. Багато компаній заохочують своїх працівників до участі у волонтерських програмах, що сприяє зміцненню корпоративного духу та покращенню іміджу компаній.

1.2.2. Роль гейміфікації у волонтерстві

Гейміфікація — це використання ігрових елементів та механік у неігрових контекстах для підвищення залученості та мотивації учасників. Цей підхід часто використовується в маркетингу, освіті, здоров'ї та управлінні персоналом, але останнім часом він також знайшов своє застосування у волонтерстві. Гейміфікація може включати такі елементи, як бали, рівні, медалі, досягнення, таблиці лідерів, виклики та нагороди.

Переваги гейміфікації у волонтерстві

Гейміфікація у волонтерстві має декілька ключових переваг, які роблять її дуже ефективним інструментом для залучення та мотивування волонтерів. Вона робить процес волонтерства не лише більш привабливим, але й захоплюючим, що допомагає залучити нових волонтерів, особливо молодь. Молоді люди, які виростили в цифрову епоху, добре знайомі з ігровими механіками, і їх використання у волонтерських програмах може мотивувати їх до участі, оскільки вони відчують себе частиною інтерактивного і динамічного процесу.

Використання ігрових елементів, таких як бали, медалі та рівні, надає волонтерам відчуття досягнення та прогресу. Це стимулює їх до подальшої участі, оскільки вони бачать конкретні результати своїх зусиль. Нагороди та визнання за виконані завдання підвищують мотивацію та задоволення від волонтерської діяльності, що, в свою чергу, сприяє більшій залученості та постійному бажанню допомагати.

Гейміфікація також сприяє створенню спільноти волонтерів, які підтримують та мотивують один одного. Таблиці лідерів та спільні виклики стимулюють дружню конкуренцію та співпрацю серед волонтерів. Це зміцнює зв'язки між учасниками та

підвищує ефективність їх роботи, оскільки волонтери відчують підтримку та натхнення від своїх колег.

Окрім цього, гейміфікація допомагає організувати та координувати волонтерські зусилля, роблячи процеси більш структурованими та зрозумілими. Волонтери можуть легко відслідковувати свої досягнення та прогрес, що дозволяє їм ефективніше розподіляти свої зусилля та час. Це не лише підвищує продуктивність, але й забезпечує волонтерів чітким уявленням про їхній внесок та вплив на загальну справу. Таким чином, гейміфікація робить волонтерську діяльність більш інтерактивною, захоплюючою та ефективною, що є важливим для залучення нових учасників та підтримки їхньої активності на довготривалій основі.

Приклади гейміфікації у волонтерстві

Сучасні волонтерські платформи активно використовують елементи гейміфікації для залучення та мотивації учасників. Наприклад, платформа "Be My Eyes" дозволяє волонтерам допомагати незрячим людям через відеодзвінки, за що вони отримують бали та відзнаки за кожну успішну допомогу, що підвищує їх мотивацію та залученість. Деякі компанії також використовують гейміфікацію для заохочення своїх працівників до участі у волонтерських проектах. Вони можуть створювати внутрішні конкурси, де працівники отримують бали та нагороди за виконані завдання, що підвищує корпоративний дух і сприяє створенню позитивного іміджу компанії. Гейміфікація також використовується у проектах, спрямованих на охорону довкілля. Наприклад, програма "Recyclebank" надає учасникам бали за правильну утилізацію відходів, які вони можуть обмінювати на знижки та подарунки, що стимулює людей брати активну участь у екологічних ініціативах.

Психологічні аспекти гейміфікації

Гейміфікація базується на принципах мотиваційної психології. Використання нагород та визнання стимулює вироблення дофаміну, що створює відчуття задоволення та заохочує до подальших дій. Важливу роль відіграє також теорія самовизначення, яка підкреслює значення автономії, компетентності та пов'язаності у підвищенні внутрішньої мотивації.

Виклики та обмеження гейміфікації у волонтерстві

Гейміфікація у волонтерстві стикається з кількома викликами та обмеженнями, які потребують ретельного врахування та вирішення. Перш за все, надзвичайно важливо зберігати автентичність волонтерства і не зосереджуватися надмірно на нагородах та інших зовнішніх стимулах. Якщо акцентувати увагу лише на винагородах, волонтери можуть втратити справжню мотивацію допомагати іншим і брати участь у проектах лише заради отримання винагород, що може зменшити щирість та якість їхньої участі. Волонтерство, в першу чергу, повинно залишатися актом альтруїзму та доброзичливості, де головною метою є допомога тим, хто цього потребує, а не просто накопичення нагород та відзнак.

Крім того, варто враховувати, що люди можуть мати різні мотивації для участі у волонтерських проектах. Деякі волонтери можуть бути мотивовані бажанням особистого розвитку або набуття нових навичок, тоді як інші можуть прагнути соціальної взаємодії або просто відчуття причетності до чогось важливого. Тому надзвичайно важливо розробляти стратегії гейміфікації, які враховують ці різноманітні мотивації та надають волонтерам можливість обирати завдання, що відповідають їхнім інтересам та цінностям. Це дозволить забезпечити більш індивідуалізований та гнучкий підхід, який підвищить залученість і задоволення волонтерів від їхньої діяльності.

Впровадження гейміфікації також вимагає значних технічних ресурсів та експертизи. Розробка та підтримка гейміфікаційних елементів можуть бути складними та витратними, особливо для невеликих організацій, які можуть не мати достатніх фінансових або технічних можливостей. Це включає не лише створення та інтеграцію відповідного програмного забезпечення, але й постійне його оновлення та адаптацію до змінних потреб та очікувань волонтерів. Технічні виклики також можуть включати питання безпеки даних та забезпечення конфіденційності особистої інформації волонтерів, що є надзвичайно важливим у сучасному цифровому середовищі.

Таким чином, хоча гейміфікація може надати значні переваги для волонтерських програм, важливо враховувати та ретельно опрацьовувати виклики та обмеження, пов'язані з її впровадженням. Це забезпечить, що гейміфікаційні елементи будуть ефективно підтримувати волонтерів у їхній діяльності, зберігаючи при цьому основні цінності та цілі волонтерства.

1.3. Аналіз потреби у віртуальному волонтерстві

1.3.1. Соціально-економічний контекст

Сучасний світ стикається з численними соціальними, економічними та екологічними викликами, серед яких глобальні здоров'я кризи, природні катастрофи та політичні конфлікти. Ці виклики вимагають швидкої та ефективної відповіді, яку може забезпечити віртуальне волонтерство. Використання цифрових технологій дозволяє організаціям оперативно мобілізувати ресурси та координувати допомогу, не обмежуючись географічними рамками [1,2].

1.3.2. Переваги віртуального волонтерства

Віртуальне волонтерство має численні переваги. Воно дозволяє залучати більшу кількість учасників, включаючи тих, хто не може брати участь у традиційних волонтерських програмах через фізичні обмеження, часові рамки або інші бар'єри.

Цифрові платформи дозволяють волонтерам з різних країн співпрацювати над проектами, ділитися знаннями та навичками, а також здійснювати координацію діяльності в реальному часі.

Ще однією перевагою є швидка мобілізація ресурсів. В умовах кризи важлива кожна хвилина. Віртуальне волонтерство дозволяє оперативно залучати необхідні ресурси та волонтерів з усього світу. Це особливо актуально під час природних катастроф, військових конфліктів чи епідемій, коли швидка реакція може врятувати життя.

Віртуальне волонтерство також зменшує витрати, пов'язані з логістикою та організацією традиційних волонтерських програм. Волонтери можуть працювати з будь-якої точки світу, що дозволяє значно знизити витрати на транспорт, проживання та харчування.

Крім того, віртуальне волонтерство сприяє інклюзивності, оскільки дозволяє залучати людей з різними здібностями та з різних культур. Це створює більш різноманітне та багатогранне волонтерське співтовариство, яке може запропонувати унікальні перспективи та рішення [1-5].

1.3.3. Підтримка військових та цивільних у кризових регіонах

Особливу увагу віртуальне волонтерство приділяє підтримці постраждалих від воєнних конфліктів та інших криз. Онлайн платформи дозволяють швидко організовувати збір коштів, розподіл гуманітарної допомоги та надання психологічної підтримки, що є критично важливим у перші години та дні після кризових подій. У випадку війни в Україні, віртуальне волонтерство стало важливим інструментом для координації допомоги та підтримки постраждалих.

1.3.4. Розширення освітніх можливостей та допомога тваринам

Віртуальне волонтерство також забезпечує важливі можливості для освітніх ініціатив та проектів по збереженню природи та допомоги тваринам. Освітні

програми, що проводяться онлайн, можуть досягати широкої аудиторії, підвищуючи обізнаність з питань екології, прав тварин, а також забезпечуючи професійну підготовку для людей у регіонах, що розвиваються.

Наприклад, онлайн-курси та семінари дозволяють волонтерам навчатися новим навичкам та знанням, що можуть бути корисними у їхній волонтерській діяльності. Крім того, віртуальні кампанії можуть залучати підтримку для притулків для тварин, забезпечуючи необхідні ресурси та підвищуючи обізнаність про проблеми захисту прав тварин.

1.3.5. Вплив гейміфікації на волонтерську активність

Впровадження елементів гейміфікації у віртуальне волонтерство виявилось ефективним способом підвищення мотивації та залученості волонтерів. Ігрові механіки, такі як бали, рівні, виклики та нагороди, допомагають перетворити рутинну діяльність на захоплюючий досвід, що спонукає волонтерів продовжувати свою участь та прагнути досягти більших результатів. Гейміфікація стимулює волонтерів до регулярної участі та надає їм відчуття досягнення та прогресу, що підвищує їхню загальну задоволеність від участі у волонтерських програмах.

1.4. Стан віртуального волонтерства в Україні

1.4.1. Розвиток та поточний стан

Віртуальне волонтерство в Україні зазнало значного розвитку в останні роки, особливо у відповідь на воєнні дії та їх наслідки. Під час кризи, викликані агресією Росії проти України, онлайн-платформи стали важливим інструментом для координації допомоги та мобілізації ресурсів. Використання цифрових технологій дозволило волонтерам з усього світу долучитися до підтримки українських військових і цивільних осіб, забезпечуючи гуманітарну, медичну та психологічну допомогу.

1.4.2. Волонтерські ініціативи та платформи

1) UNITED24:

Платформа UNITED24, створена урядом України, стала однією з головних ініціатив для збору коштів та підтримки гуманітарних проектів. За допомогою цієї платформи міжнародні донори можуть робити внески для підтримки різних програм, таких як медична допомога, відбудова інфраструктури та допомога внутрішньо переміщеним особам [6].

2) Razom for Ukraine:

Організація Razom for Ukraine використовує цифрові платформи для координації волонтерських зусиль та збору коштів для підтримки різних ініціатив, включаючи медичну допомогу та підтримку освіти. Ця організація активно співпрацює з міжнародними партнерами, щоб забезпечити швидку та ефективну допомогу [7].

3) Dobro.ua:

Dobro.ua є однією з провідних українських благодійних платформ, яка використовує інтернет для збору коштів на різні соціальні та гуманітарні проекти. Віртуальні волонтери можуть підтримувати ці ініціативи, поширюючи інформацію, організовуючи збори коштів та надаючи технічну підтримку [8].

4) HelpUkraine.Center:

Ця платформа забезпечує логістичну підтримку та координацію доставки гуманітарної допомоги з усього світу до України. Віртуальні волонтери допомагають у зборі інформації, координації вантажів та організації транспортування [9].

5) UN Volunteers (UNV) Online Volunteering Service

UNV пропонує онлайн платформу, яка дозволяє волонтерам з усього світу брати участь у різноманітних проектах, що організовуються різними організаціями ООН та їх партнерами [10].

6) VolunteerMatch

Опис: VolunteerMatch — це платформа, яка з'єднує волонтерів з організаціями, що потребують допомоги в США. Платформа пропонує широкий спектр можливостей для волонтерів [11].

1.4.3. Вплив гейміфікації

Віртуальне волонтерство в Україні продовжує розвиватися, надаючи важливу підтримку у вирішенні кризових ситуацій та підвищуючи ефективність гуманітарних ініціатив. Завдяки онлайн-платформам та інноваційним підходам, таким як гейміфікація, волонтерство стає більш доступним та ефективним. Але всеж залишаються люди які хочуть допомогти справою, тому для таких людей потрібно створити платформу на якій вони зможуть шукати запити як на фізичну, так і на віртуальну допомогу.

1.5. Обґрунтування необхідності розробки нової платформи

1.5.1. Недоліки існуючих систем

Хоча існуючі платформи для віртуального волонтерства в Україні та в інших країнах пропонують значні можливості для залучення громадськості, вони часто мають обмеження або незручності.

Приклади:

1) Недоліки UNITED24

- Обмежена взаємодія з волонтерами: Платформа в основному орієнтована на збір коштів, а не на координацію волонтерських

зусиль. Це обмежує можливості для активної участі волонтерів у проектах.

- Відсутність гейміфікаційних елементів: Немає використання ігрових механік, які б могли підвищити мотивацію та залученість волонтерів.

2) Недоліки Razom for Ukraine

- Географічні обмеження: Основна діяльність організації спрямована на Україну, що може обмежувати участь міжнародних волонтерів.
- Відсутність інтерактивних елементів: Платформа не використовує інтерактивні елементи, які могли б залучити більше волонтерів і підтримати їхню активність.

3) Недоліки Dobro.ua

- Обмежена функціональність для волонтерів: Платформа більше орієнтована на збір коштів, а не на активну участь волонтерів у виконанні завдань.
- Недостатня гейміфікація: Відсутні ігрові елементи, які могли б підвищити залученість та мотивацію волонтерів.

4) Недоліки HelpUkraine.Center

- Фокус на логістику: Основний акцент платформи зроблено на логістичній підтримці, що обмежує можливості для залучення волонтерів до інших видів діяльності.
- Відсутність елементів гейміфікації: Платформа не використовує гейміфікаційні елементи, які могли б зробити процес волонтерства більш захоплюючим і мотивуючим.

5) Недоліки UN Volunteers (UNV) Online Volunteering Service

- Обмежена спеціалізація: Платформа сильно зосереджена на проектах, що відповідають специфічним потребам організацій ООН, що може не відповідати інтересам або спеціалізації всіх волонтерів.
- Відсутність гейміфікації: На платформі майже не використовуються елементи гейміфікації, які б могли підвищити мотивацію та залученість волонтерів.

6) Недоліки VolunteerMatch

- Географічні обмеження: Більшість можливостей зосереджена в США, що обмежує участь міжнародних волонтерів.
- Недостатній захист даних: Існують звіти про проблеми з конфіденційністю та захистом персональних даних волонтерів.

1.5.2. Як нова платформа може вирішити ці проблеми:

- Краща інтеграція та спеціалізація: Розробка платформи з можливістю більш точного налаштування під специфічні потреби та інтереси волонтерів, включаючи більш широкий спектр проектів не тільки гуманітарного спрямування.
- Використання гейміфікації: Інтеграція гейміфікаційних елементів для підвищення мотивації та залученості волонтерів, включаючи досягнення, медалі, бали та виклики.
- Міжнародне залучення та доступність: Забезпечення більшої доступності для міжнародних волонтерів, включаючи багатомовну підтримку та адаптацію платформ для міжнародних проектів із врахуванням культурних особливостей учасників.

- Підвищення конфіденційності та безпеки даних: Впровадження сучасних технологій захисту даних для забезпечення конфіденційності інформації волонтерів, використовуючи найкращі практики кібербезпеки.

1.5.3. Потреба у покращенні технологій

Розвиток цифрових технологій і волонтерських сервісів створює умови для створення більш ефективних, гнучких та інтегрованих платформ. Такі платформи забезпечать кращу інтеграцію з іншими системами та сервісами, покращуючи координацію дій. Вони також дозволяють індивідуальні налаштування, адаптуючи функціонал до потреб конкретного користувача або ситуації. Крім того, ці платформи ефективно використовують гейміфікацію для залучення більшої кількості волонтерів та підтримки їх активності на довготривалій основі.

1.5.4. Вплив на волонтерську активність

Нова платформа може значно підвищити ефективність волонтерської діяльності, надаючи більш широкий доступ до волонтерських можливостей, залучаючи не тільки місцевих, але й міжнародних учасників. Вона також сприятиме підвищенню залучення молоді через інноваційні технологічні рішення, які роблять волонтерську діяльність більш привабливою та доступною для молодих людей [1,2].

1.5.5. Передові практики та інновації

Розробка нової платформи має бути спрямована на впровадження передових практик і технологічних інновацій, що з'явилися на ринку в останні роки. Зокрема, важливо використовувати дані для аналізу ефективності волонтерських програм, а також, за можливістю, штучний інтелект для персоналізації досвіду користувачів і оптимізації ресурсів.

1.6. Висновки до першого розділу

Ціллю даного дослідження є розробка та впровадження інноваційної платформи для віртуального волонтерства з використанням гейміфікаційних елементів, щоб підвищити залученість та мотивацію волонтерів. Основні умови завдання включають забезпечення інтерактивності, доступності та ефективності платформи у контексті сучасних соціальних викликів, таких як війна в Україні та інші глобальні кризи.

Для вирішення поставленого завдання необхідні наступні вхідні дані. Перш за все, було проведено аналіз існуючих платформ віртуального волонтерства, досліджуючи сильні та слабкі сторони поточних рішень, таких як UNITED24, Razom for Ukraine, Dobro.ua, HelpUkraine.Center, UNV Online Volunteering Service, VolunteerMatch тощо. Далі слід зібрати та проаналізувати інформацію про потреби користувачів, включаючи дані про потреби та очікування волонтерів, організацій та бенефіціарів. Важливим аспектом також стало вивчення теорій та практик гейміфікації, зокрема ефективних гейміфікаційних стратегій у неігрових контекстах, таких як волонтерство.

Результатом виконання завдання стане розробка нової платформи для віртуального волонтерства. Ця платформа включатиме інтерактивні інструменти, такі як система гейміфікації з балами, медалями, рівнями, викликами та таблицями лідерів. Вона також забезпечить засоби комунікації та координації між волонтерами та організаціями. Крім того, платформа буде мати інформаційну базу даних, що міститиме документи, кадри на екрані монітора, інформацію в базі даних та вихідний сигнал пристрою управління.

Вимоги до виконання завдання включають інтеграцію гейміфікаційних елементів, щоб стимулювати активну участь волонтерів. Платформа повинна відповідати найкращим практикам захисту персональних даних, забезпечуючи безпеку та конфіденційність.

Пропозиції та вимоги до платформи включають адаптацію до потреб користувачів, забезпечення гнучких налаштувань для індивідуальних потреб кожного користувача. Використання передових технологій буде спрямоване на аналіз даних та персоналізацію досвіду користувачів. Платформа також повинна підтримувати довгострокову активність волонтерів, вводячи нові ігрові елементи для підтримки інтересу та мотивації протягом тривалого часу.

2. РОЗДІЛ 2. Алгоритмічне та математичне пояснення

2.1.Методологія розробки

2.1.1. Вибір методології

Для розробки мобільного додатку EchoAid було обрано методологію Agile. Agile — це підхід до управління проектами, який наголошує на гнучкості, співпраці, постійному вдосконаленні та доставці цінності замовнику. Методологія Agile базується на принципах, викладених у "Маніфесті Agile розробки програмного забезпечення"[12].

2.1.2. Принципи методології Agile

- 1) Задоволення клієнта: Найвищий пріоритет — це задоволення клієнта через ранню та постійну доставку цінного програмного забезпечення [12].
- 2) Гнучкість до змін: Ласкаво просимо до змін вимог, навіть на пізніх етапах розробки. Agile процеси використовують зміни для конкурентної переваги клієнта [12].
- 3) Часта доставка: Часто доставляйте робоче програмне забезпечення, з періодичністю від кількох тижнів до кількох місяців, віддаючи перевагу коротшому часовому проміжку [12].
- 4) Щоденна співпраця: Замовники та розробники повинні щоденно співпрацювати протягом усього проекту [12].
- 5) Мотивовані люди: Створюйте проекти навколо мотивованих людей. Надайте їм середовище та підтримку, і довіряйте їм виконання роботи [12].
- 6) Особиста комунікація: Найефективніший та найрезультативніший метод передачі інформації до та в межах команди розробників — це розмова віч-на-віч [12].
- 7) Робоче програмне забезпечення: Робоче програмне забезпечення — основний показник прогресу [12].

- 8) Стійкий розвиток: Agile процеси сприяють сталому розвитку. Замовники, розробники та користувачі повинні мати можливість підтримувати постійний темп роботи [12].
- 9) Технічна досконалість: Постійна увага до технічної досконалості та гарного дизайну підвищує гнучкість Agile [12].
- 10) Самоорганізовані команди: Найкращі архітектури, вимоги та дизайни виникають із самоорганізованих команд [12].
- 11) Ретроспективи та вдосконалення: Регулярно, команда розмірковує над тим, як стати більш ефективною, та відповідно коригує свою поведінку [12].

2.1.3. Обґрунтування вибору методології Agile

Методологія Agile була обрана для розробки EchoAid з кількох причин:

- 1) Гнучкість у розробці: Проект EchoAid потребує швидкого реагування на зміну вимог та умов. Agile дозволяє легко адаптуватися до нових вимог, що з'являються в процесі розробки.
- 2) Ітеративний підхід: Agile передбачає розбиття проекту на короткі цикли або спринти, кожен з яких включає всі етапи розробки — від планування до тестування. Це дозволяє регулярно отримувати зворотний зв'язок від користувачів та вносити необхідні корективи.
- 3) Фокус на цінності: Методологія Agile орієнтована на доставку цінності для користувачів через створення функціонального програмного забезпечення на кожному етапі.
- 4) Задоволення користувачів: Agile підходить для проектів, де важливо постійно враховувати відгуки користувачів та швидко адаптувати продукт відповідно до їх потреб. У проекті EchoAid, де користувачами є волонтери та організатори, це особливо важливо.

2.1.4. Реалізація Agile в проекті EchoAid

Процес розробки EchoAid за методологією Agile включає декілька важливих етапів, які допомагають ефективно управляти проектом і забезпечити високу якість кінцевого продукту. Перший етап, планування спринтів, є надзвичайно важливим для встановлення чітких цілей та завдань на найближчі 2-3 тижні. Це дозволяє сфокусувати зусилля на конкретних аспектах проекту, забезпечуючи поступове та послідовне просування вперед. На цьому етапі я ретельно аналізую вимоги, визначаю пріоритети та формую план дій, що сприяє досягненню запланованих результатів у межах кожного спринту [12-14].

Щоденні стендап-зустрічі, хоча і проводяться мною самостійно, є важливою складовою Agile-процесу. Кожного дня я роблю короткі оцінки свого прогресу, аналізую виконані завдання, поточний стан проекту та можливі перешкоди. Ці щоденні оцінки допомагають мені підтримувати високий рівень організації та продуктивності, дозволяючи вчасно виявляти проблеми та оперативно реагувати на них. Такий підхід сприяє безперервному покращенню процесу розробки та забезпечує плавний перебіг проекту [12,13].

Розробка та тестування є основними етапами, які займають більшу частину часу кожного спринту. Під час цих етапів я займаюся створенням нових функцій, їх інтеграцією та тестуванням для забезпечення їхньої коректної роботи. Кожен новий елемент проходить ретельну перевірку, щоб гарантувати, що всі функції додатку працюють належним чином та відповідають вимогам користувачів. Тестування включає як модульне, так і інтеграційне та системне тестування, що дозволяє виявити та виправити можливі помилки на ранніх стадіях розробки.

Наприкінці кожного спринту я проводжу демонстрацію результатів. Це важливий етап, під час якого я отримую зворотний зв'язок від потенційних користувачів або інших зацікавлених сторін. Така практика дозволяє оцінити досягнення, виявити недоліки та внести необхідні корективи до подальшого плану дій. Отриманий зворотний зв'язок є цінним джерелом інформації для вдосконалення

продукту, оскільки дозволяє краще зрозуміти потреби користувачів та адаптувати розробку відповідно до їхніх очікувань.

Завершальним етапом кожного спринту є ретроспектива. Під час ретроспективи я аналізую досягнення, проблеми, з якими зіткнувся під час спринту, та можливі вдосконалення процесу. Цей етап дозволяє мені рефлексувати над власною роботою, виявляти ефективні практики та усувати недоліки. Ретроспектива сприяє постійному вдосконаленню процесу розробки та допомагає підвищити якість кінцевого продукту [1,2].

Вибір методології Agile для проекту EchoAid обґрунтований потребою в гнучкості, швидкій адаптації до змін та постійному вдосконаленні продукту на основі зворотного зв'язку від користувачів. Методологія Agile дозволяє ефективно управляти процесом розробки, забезпечуючи можливість швидко реагувати на зміни та адаптуватися до нових вимог. Це особливо важливо при розробці додатку однією людиною, оскільки дозволяє зберігати високий рівень організації та продуктивності, забезпечуючи успішне завершення проекту та високу якість кінцевого продукту.

2.2.Збір вимог до проекту

2.2.1. Методи збору вимог

Для успішної розробки мобільного додатку EchoAid було важливо зібрати повні та актуальні вимоги [4]. Для цього використовувалися різні методи, що дозволили отримати всебічне розуміння потреб та очікувань майбутніх користувачів.

Одним із основних методів були інтерв'ю з потенційними користувачами, серед яких були волонтери, організатори волонтерських заходів та представники волонтерських організацій. Проведення серії інтерв'ю дозволило виявити конкретні потреби, очікування та проблеми, з якими стикаються ці користувачі у своїй діяльності. Цей метод забезпечив глибоке розуміння контексту використання додатку

та допоміг визначити ключові функції, необхідні для задоволення потреб користувачів.

Крім інтерв'ю, було створено та розповсюджено онлайн-опитування, яке дозволило зібрати кількісні дані від ширшої аудиторії. Це опитування включало питання про функціональні можливості, які користувачі хотіли б бачити в додатку, а також про їхній досвід використання інших волонтерських платформ. Зібрані дані допомогли підтвердити результати інтерв'ю та виявити додаткові вимоги, які можуть бути важливими для розробки [15,16].

Ще одним важливим методом був аналіз існуючих рішень на ринку. Було проведено детальний огляд волонтерських додатків та платформ, таких як United24, Razom for Ukraine, Dobro.ua, HelpUkraine.Center та інші. Аналіз існуючих рішень допоміг визначити, які функції є найбільш затребуваними серед користувачів, а також виявити їхні недоліки та можливості для покращення. Цей підхід дозволив врахувати найкращі практики та уникнути поширених помилок при розробці EchoAid.

Завдяки використанню цих методів вдалося зібрати вичерпну інформацію, яка стала основою для визначення вимог до мобільного додатку EchoAid. Це дозволило створити продукт, який відповідає реальним потребам користувачів та має потенціал для успішного впровадження на ринку волонтерських послуг.

2.2.2. Документація вимог

Всі зібрані вимоги були задокументовані у вигляді користувацьких історій (user stories). Користувацькі історії містять короткі описи функціональності з точки зору кінцевих користувачів та слугують основою для подальшої розробки [15].

Приклади користувацьких історій:

- Як волонтер, я хочу мати можливість реєструватися та входити в систему, щоб мати доступ до всіх функцій додатку.
- Як організатор волонтерських заходів, я хочу мати можливість створювати нові завдання, щоб залучати волонтерів до виконання конкретних задач.
- Як користувач, я хочу мати можливість переглядати та фільтрувати завдання, щоб швидко знаходити завдання, які мене цікавлять.

2.2.3. Пріоритизація вимог

Після збору та документування вимог було проведено їхню пріоритизацію, щоб визначити, які функції мають бути реалізовані в першу чергу [16,17]. Пріоритизація виконувалася з урахуванням кількох факторів:

- 1) Важливість для користувачів: Функції, які користувачі вважають найбільш важливими для їхньої роботи та які найбільше впливають на їхній досвід використання додатку.
- 2) Технічна складність: Оцінка складності реалізації кожної функції, щоб забезпечити реалістичне планування та управління ресурсами.
- 3) Вплив на проект: Оцінка того, як кожна функція впливає на загальний успіх проекту, включаючи можливість отримання зворотного зв'язку та швидке вдосконалення продукту.

Функції було розподілено за трьома категоріями пріоритету:

- Високий пріоритет: Ключові функції, необхідні для мінімально життєздатного продукту (MVP).
- Середній пріоритет: Функції, які додають значну цінність, але можуть бути реалізовані після запуску MVP.
- Низький пріоритет: Функції, які можуть бути додані на пізніших етапах розробки.

Збір вимог до проекту EchoAid був важливим етапом для забезпечення успішної розробки додатку, орієнтованого на потреби користувачів. Використання різних методів збору даних, документування вимог у вигляді користувацьких історій та пріоритизація функцій дозволили створити чітке уявлення про те, які функціональні можливості необхідні для забезпечення високої якості та зручності використання додатку.

2.3. Проектування архітектури системи

2.3.1. Визначення архітектури

Для забезпечення гнучкості, масштабованості та підтримки простоти розвитку проекту було обрано об'єктно-орієнтовану архітектуру системи. Ця архітектура дозволяє розділити систему на окремі модулі та компоненти, що значно спрощує процес розробки, тестування та підтримки програмного забезпечення [16-18].

Основними принципами об'єктно-орієнтованої архітектури є модульність, гнучкість, масштабованість та повторне використання. Модульність передбачає розподіл системи на незалежні модулі, кожен з яких відповідає за виконання конкретних функцій. Це забезпечує можливість розробки та тестування кожного модуля окремо, що підвищує загальну ефективність і якість роботи над проектом.

Гнучкість архітектури забезпечується можливістю легко адаптуватися до змін у вимогах. Це дозволяє додавати нові функції або вносити зміни в існуючі компоненти без значних зусиль та ризиків для стабільності системи. Такий підхід гарантує, що система залишається актуальною та може швидко реагувати на потреби користувачів або зміни в ринкових умовах.

Масштабованість системи досягається через можливість додавання нових модулів або розширення існуючих. Це означає, що система може рости разом з проектом, підтримуючи збільшення обсягу даних або кількості користувачів без

зниження продуктивності. Така властивість є критично важливою для забезпечення довготривалої успішності проекту.

Повторне використання компонентів та бібліотек підвищує ефективність розробки, дозволяючи використовувати перевірені рішення та зменшувати час на розробку нових функцій. Використання стандартних компонентів також сприяє підвищенню надійності та якості програмного забезпечення, оскільки зменшується кількість помилок та забезпечується краща інтеграція між різними частинами системи.

Загалом, об'єктно-орієнтована архітектура забезпечує високу якість, гнучкість та ефективність розробки програмного забезпечення, що є ключовими факторами для успішного розвитку проекту EchoAid.

2.3.2. Створення діаграм

Для візуалізації архітектури системи були створені UML-діаграми, які допомагають краще зрозуміти структуру системи, взаємодію між її компонентами та основні сценарії використання. Нижче наведені основні типи діаграм, що використовувалися у проектуванні EchoAid [18,19].

1) Діаграма класів

Діаграма класів відображає структуру системи, включаючи класи, їх атрибути, методи та зв'язки між класами. Вона дозволяє визначити основні об'єкти системи та їхні взаємозв'язки. Ця діаграма є важливим інструментом для розробників, оскільки надає візуальне представлення архітектури програмного забезпечення. Вона допомагає чітко зрозуміти, які класи існують у системі, які дані вони зберігають та які методи використовують для обробки цих даних.

Діаграма класів також дозволяє розробникам бачити зв'язки між класами, що включають асоціації, агрегації, композиції та наслідування. Це допомагає забезпечити правильне проектування системи та уникнути можливих помилок у взаємодії між об'єктами. Завдяки діаграмі класів можна ефективніше планувати

розробку, тестування та підтримку програмного забезпечення, оскільки всі ключові компоненти та їх взаємодії чітко визначені та задокументовані.

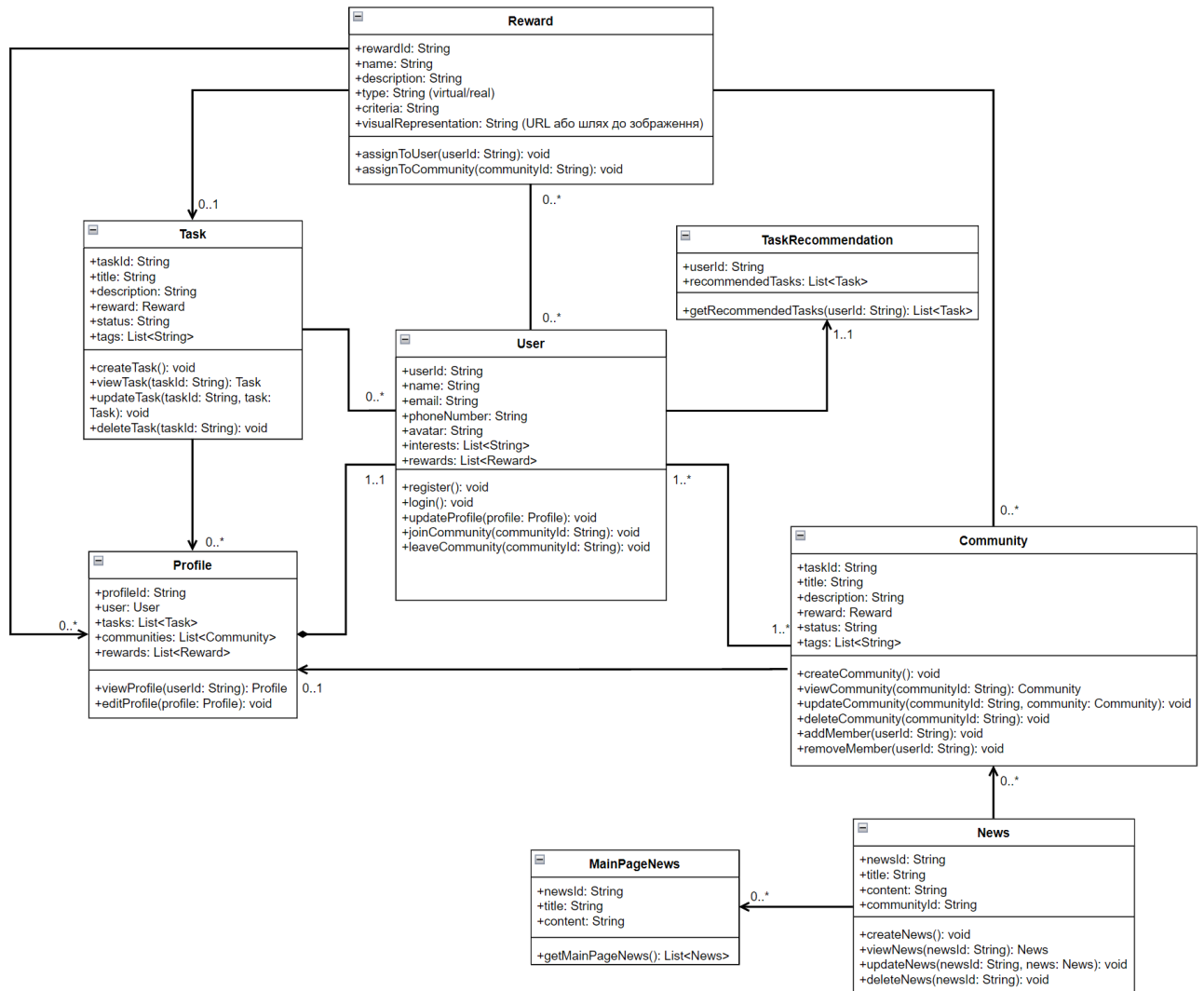


Рис. 2.1. Діаграма класів

2) Діаграма компонентів

Діаграма компонентів відображає різні компоненти системи та їхні зв'язки. Вона дозволяє зрозуміти, як компоненти взаємодіють один з одним та з зовнішніми системами. Ця діаграма є важливим інструментом для розробників та архітекторів

програмного забезпечення, оскільки надає візуальне представлення структурної організації системи на рівні компонентів.

Діаграма компонентів допомагає визначити основні блоки системи, їхні функції та взаємозв'язки. Це дозволяє побачити, які компоненти є незалежними, які з них взаємодіють безпосередньо та які компоненти потребують інтеграції з зовнішніми системами або сервісами. Вона також показує інтерфейси, через які компоненти взаємодіють між собою, що є ключовим для розуміння комунікаційних потоків та залежностей.

Завдяки діаграмі компонентів розробники можуть ефективніше планувати процес розробки, інтеграції та тестування системи. Вона забезпечує чітке уявлення про архітектуру програмного забезпечення, сприяючи кращому управлінню складністю проекту. Діаграма також допомагає виявити потенційні проблеми у взаємодії компонентів та визначити області, де може знадобитися додаткова оптимізація або рефакторинг.

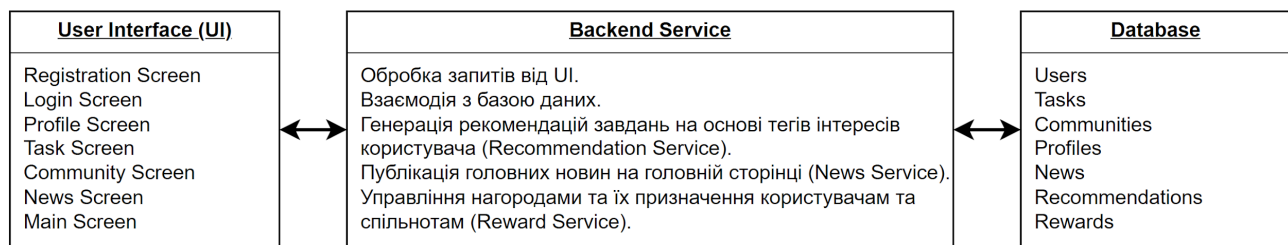


Рис. 2.2. Діаграма компонентів

3) Діаграми послідовностей

Діаграма послідовностей відображає взаємодії між об'єктами в системі у вигляді послідовності обміну повідомленнями. Вона дозволяє зрозуміти, як об'єкти взаємодіють один з одним протягом виконання конкретного процесу або сценарію.

Ця діаграма є важливим інструментом для аналізу та документування динамічних аспектів системи.

Діаграма послідовностей показує об'єкти, що беруть участь у процесі, та порядок, в якому вони обмінюються повідомленнями. Вона включає вертикальні лінії життєвих циклів об'єктів та горизонтальні стрілки, що представляють повідомлення, які передаються між об'єктами. Це допомагає виявити, які об'єкти взаємодіють між собою, у якій послідовності та за допомогою яких методів.

Завдяки діаграмі послідовностей розробники можуть детально аналізувати логіку виконання окремих функцій або сценаріїв, виявляти можливі проблеми у взаємодії об'єктів та оптимізувати процеси.

Діаграма послідовностей дозволяє забезпечити правильну реалізацію вимог, оскільки чітко показує, як мають виконуватися різні процеси в системі. Вона також сприяє виявленню неефективних або зайвих взаємодій, що можуть негативно вплинути на продуктивність системи, та допомагає їх усунути.



Рис. 2.3. Діаграма послідовності «Реєстрація користувача»

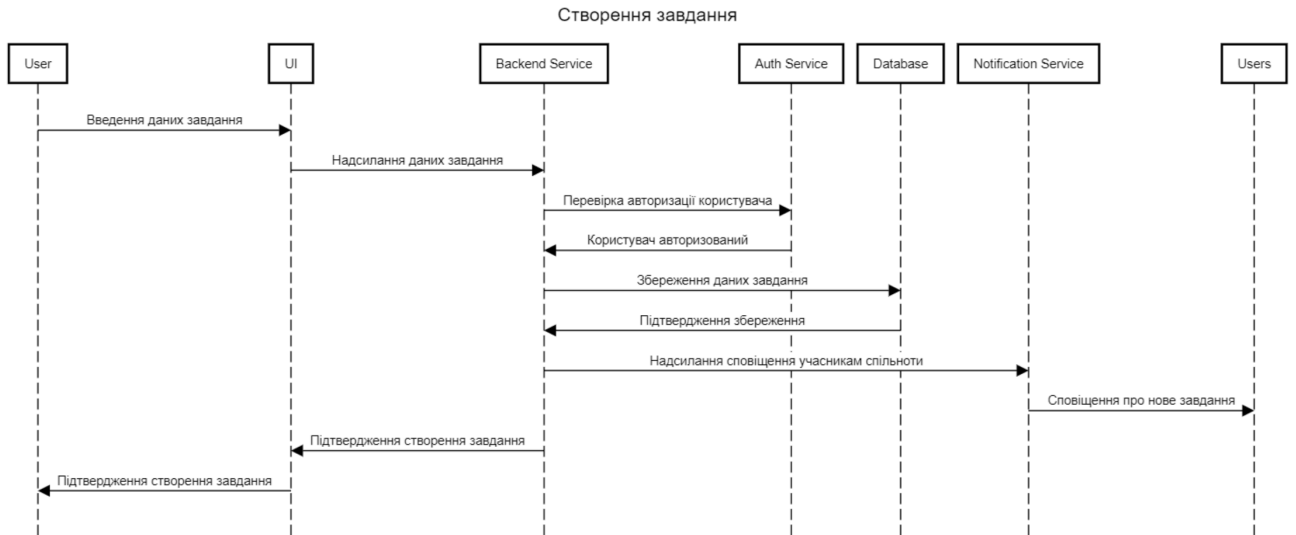


Рис. 2.4. Діаграма послідовності «Створення завдання»

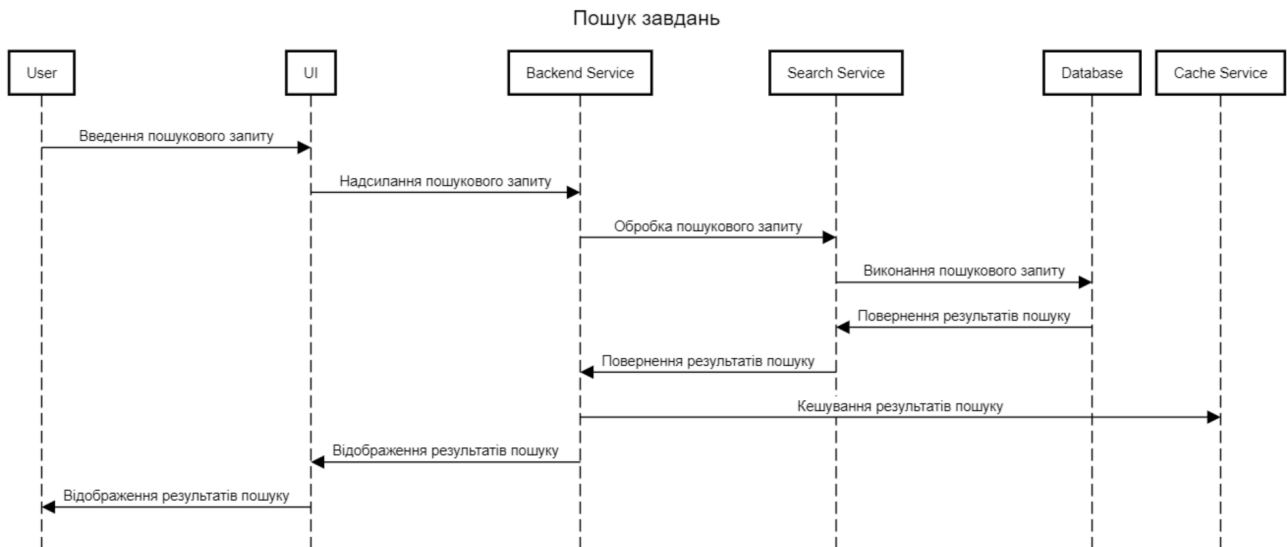


Рис. 2.5. Діаграма послідовності «Пошук завдань»

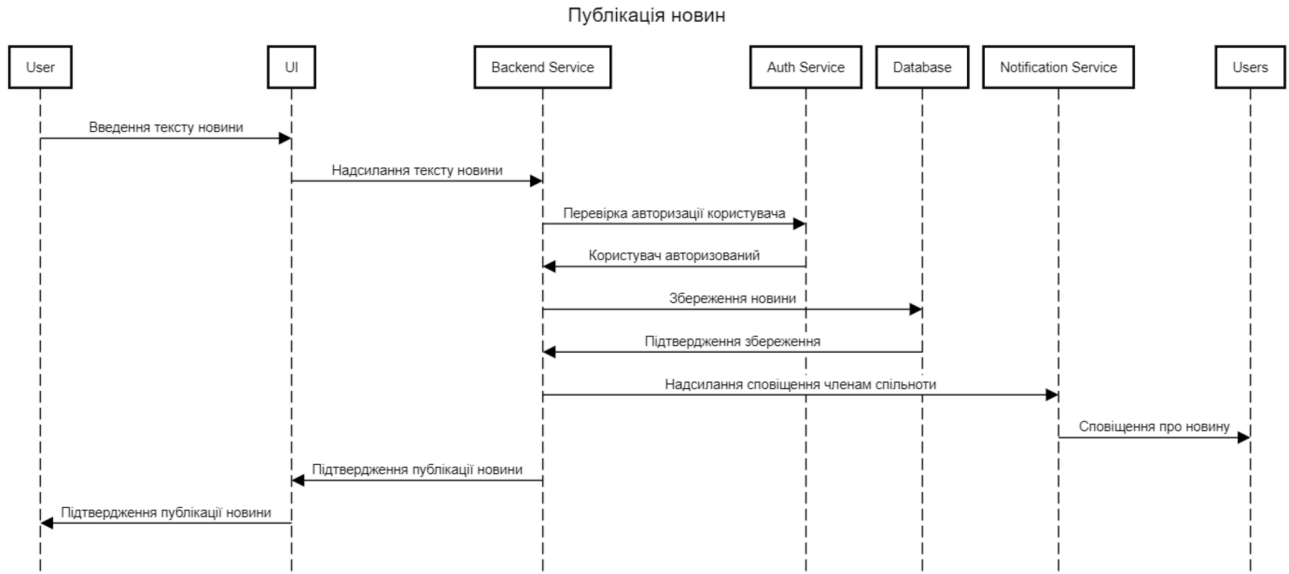


Рис. 2.6. Діаграма послідовності «Публікація новин»



Рис. 2.7. Діаграма послідовності «Отримання рекомендацій завдань»

4) Діаграми випадків використання

Діаграма випадків використання відображає функціональні можливості системи з точки зору користувача, показуючи взаємодію різних типів користувачів (акторів) із системою для виконання завдань або досягнення цілей. Вона складається з акторів, випадків використання та їхніх взаємозв'язків. Актори представляють різні ролі, такі як кінцеві користувачі, адміністратори або зовнішні системи, а випадки використання — конкретні дії або процеси, які вони можуть виконувати.

Ця діаграма допомагає визначити основні функції системи та зв'язки між ними, забезпечуючи спільне розуміння вимог користувачів. Вона є корисним інструментом для планування розробки, тестування та впровадження системи, допомагаючи виявити та уточнити вимоги на ранніх етапах проекту. Діаграма випадків використання сприяє створенню продукту, який відповідає потребам та очікуванням користувачів.

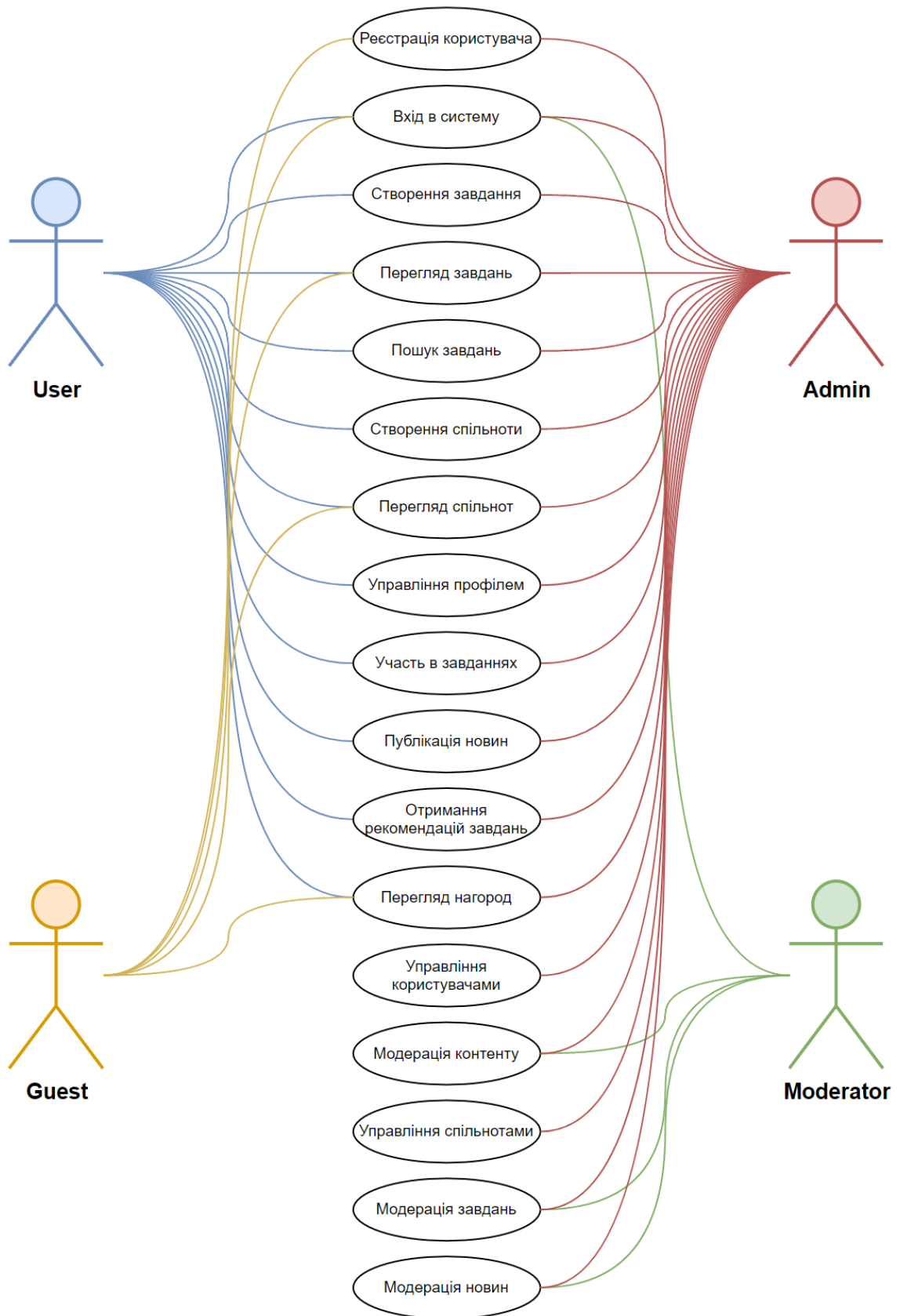


Рис. 2.8. Діаграма випадків використання

Проектування архітектури системи EchoAid включало створення детальних UML-діаграм, які допомагають візуалізувати структуру та функціональність системи. Використання об'єктно-орієнтованого підходу дозволило забезпечити модульність, гнучкість, масштабованість та повторне використання компонентів, що значно спрощує процес розробки, тестування та підтримки програмного забезпечення.

2.4. Бібліотеки та фреймворки для розробки додатку

Для реалізації мобільного додатку EchoAid були обрані різні бібліотеки та фреймворки, які допомогли забезпечити ефективну розробку, підтримку та масштабованість додатку. Нижче наведені основні з них та причини їх вибору.

2.4.1. Android SDK

Android SDK (Software Development Kit) є основним набором інструментів для розробки додатків для операційної системи Android. Він включає в себе компілятор, бібліотеки, емулятори та інші інструменти, необхідні для створення, тестування та налагодження додатків [20].

Android Studio є офіційним середовищем розробки (IDE) для Android, яке забезпечує інтеграцію з Android SDK, надаючи розробникам зручний інтерфейс для створення додатків. Android Emulator дозволяє тестувати додатки на різних пристроях та версіях Android, що є важливим для забезпечення сумісності та стабільності додатків. Android Debug Bridge (ADB) є інструментом командного рядка, який дозволяє взаємодіяти з пристроями Android для налагодження та інших завдань.

Використання Android SDK забезпечує всі необхідні інструменти для розробки додатків в одному місці. SDK постійно оновлюється та підтримується Google, що

гарантує актуальність та доступ до новітніх технологій і функцій. Крім того, Android SDK інтегрується з іншими інструментами розробки та тестування, що спрощує процес створення якісних додатків.

2.4.2. Retrofit

Retrofit є бібліотекою для виконання HTTP-запитів та обробки відповідей у форматі JSON. Вона значно спрощує роботу з RESTful API, дозволяючи легко визначати інтерфейси для запитів та автоматично перетворювати відповіді в об'єкти. Основною функцією Retrofit є перетворення JSON-відповідей у Java-об'єкти за допомогою бібліотеки Gson або Moshi. Бібліотека підтримує різні типи запитів, включаючи GET, POST, PUT та DELETE, і може обробляти як асинхронні, так і синхронні запити [21].

Використання Retrofit значно спрощує роботу з мережевими запитами, зменшуючи кількість коду, необхідного для взаємодії з API. Крім того, вона легко інтегрується з іншими бібліотеками для обробки JSON, що робить її зручним і ефективним інструментом для розробників.

2.4.3. Room

Room є частиною архітектурних компонентів Android Jetpack і використовується для роботи з базою даних SQLite. Вона забезпечує об'єктно-реляційне відображення (ORM), що дозволяє розробникам використовувати об'єкти для взаємодії з базою даних, спрощуючи цю взаємодію та зменшуючи кількість необхідного коду.

Основною функцією Room є генерація запитів SQL на основі анотацій в Java-класах, що дозволяє розробникам писати менш складний код для взаємодії з

базою даних. Крім того, Room підтримує потоки даних через LiveData та Flow, що забезпечує реактивну обробку даних. Також вона автоматично обробляє зміни в структурі бази даних, що полегшує процес міграції даних.

Використання Room спрощує роботу з базами даних, забезпечуючи безпечні та ефективні операції з даними. Вона інтегрується з іншими архітектурними компонентами Android, що робить її зручним і потужним інструментом для розробки додатків [22].

2.4.4. Glide

Glide є бібліотекою для завантаження та кешування зображень, яка дозволяє легко завантажувати зображення з URL-адрес та показувати їх у компонентах користувацького інтерфейсу. Ця бібліотека підтримує завантаження зображень з різних джерел, включаючи URL, локальні файли та ресурси додатку. Glide автоматично кешує зображення, що зменшує навантаження на мережу та покращує продуктивність додатку.

Крім основних функцій завантаження та кешування, Glide підтримує різноманітні трансформації та анімації зображень, що дозволяє розробникам легко налаштовувати вигляд зображень у додатку. Використання Glide значно спрощує роботу із завантаженням та відображенням зображень, роблячи цей процес більш ефективним та менш трудомістким. Бібліотека легко інтегрується з RecyclerView та іншими компонентами інтерфейсу, що робить її зручним інструментом для розробників додатків [23].

2.4.5. Hilt

Hilt є бібліотекою для впровадження залежностей (Dependency Injection, DI), яка є частиною архітектурних компонентів Android Jetpack. Вона базується на Dagger і спрощує управління залежностями у додатку, забезпечуючи автоматичне впровадження необхідних об'єктів.

Hilt використовує анотації для автоматичного впровадження залежностей, що значно спрощує процес написання коду. Вона підтримує життєвий цикл компонентів Android, автоматично створюючи графи залежностей, що полегшує управління залежностями в різних частинах додатку.

Використання Hilt спрощує управління залежностями, забезпечуючи модульність та тестованість коду. Це дозволяє розробникам створювати більш гнучкі та легко підтримувані додатки. Hilt також інтегрується з іншими архітектурними компонентами Android, що робить її зручним і потужним інструментом для розробки сучасних Android-додатків [24].

2.4.6. Navigation Component

Navigation Component є частиною архітектурних компонентів Android Jetpack і використовується для навігації між екранами у додатку. Вона спрощує визначення навігаційних шляхів та обробку переходів, роблячи процес створення складних навігаційних структур більш зрозумілим та керованим.

Однією з основних функцій Navigation Component є графічний редактор для створення графу навігації, що дозволяє розробникам візуально визначати зв'язки між екранами. Вона підтримує навігацію через фрагменти та активності, забезпечуючи гнучкість у побудові додатку. Крім того, Navigation Component обробляє переходи та анімації, що покращує користувацький досвід і додає візуальної привабливості.

Використання Navigation Component значно спрощує навігацію у додатку, зменшуючи кількість коду, необхідного для обробки навігаційних переходів. Інтеграція з іншими архітектурними компонентами Android робить її потужним інструментом для розробки сучасних, зручних у використанні додатків [25].

Використання цих бібліотек та фреймворків дозволило значно спростити процес розробки мобільного додатку EchoAid, забезпечуючи високу якість, ефективність та масштабованість додатку. Кожна з обраних бібліотек має свої унікальні переваги та допомагає вирішувати конкретні задачі, що робить процес розробки більш зручним та продуктивним.

2.5.Опис функціональних особливостей

2.5.1. Реєстрація користувача

Опис:

Користувач вводить основні дані (ім'я, email, пароль) для створення облікового запису.

Кроки:

- 1) Користувач переходить на екран реєстрації.
- 2) Вводить своє ім'я, email та пароль.
- 3) Натискає кнопку "Реєстрація".
- 4) Система перевіряє правильність введених даних та наявність облікового запису з таким email.
- 5) Якщо дані коректні, користувач отримує лист підтвердження на email.

- 6) Після підтвердження email обліковий запис активується, і користувач може увійти в систему.

Вимоги:

- Введення унікального email.
- Введення паролю, що відповідає вимогам безпеки (мінімальна довжина, наявність великих і малих літер, цифр).

2.5.2. Вхід в систему

Опис:

Користувач вводить email та пароль для входу в обліковий запис.

Кроки:

- 1) Користувач переходить на екран входу.
- 2) Вводить свій email та пароль.
- 3) Натискає кнопку "Увійти".
- 4) Система перевіряє правильність введених даних.
- 5) Якщо дані коректні, користувач отримує доступ до свого облікового запису.

Вимоги:

- Введення правильного email та паролю.
- Обробка помилок при неправильному введенні даних.

2.5.3. Створення завдання

Опис:

Користувач може створити нове завдання, вказуючи назву, опис, теги, нагороду та інші деталі.

Кроки:

- 1) Користувач переходить на екран створення завдання.
- 2) Вводить назву завдання, опис, теги, нагороду та інші необхідні деталі.
- 3) Натискає кнопку "Створити".
- 4) Система зберігає завдання в базі даних та робить його доступним для інших користувачів.

Вимоги:

- Введення всіх необхідних даних.
- Перевірка наявності обов'язкових полів.

2.5.4. Перегляд завдань

Опис:

Користувач може переглядати список доступних завдань.

Кроки:

- 1) Користувач переходить на екран завдань.
- 2) Система завантажує та відображає список доступних завдань.

- 3) Користувач може переглядати завдання, скролити список, обирати завдання для детального перегляду.

Вимоги:

- Відображення списку завдань.
- Можливість перегляду деталей завдання.

2.5.5. Пошук завдань

Опис:

Користувач може здійснювати пошук завдань за ключовими словами та тегами.

Кроки:

- 1) Користувач переходить на екран пошуку завдань.
- 2) Вводить ключові слова або обирає теги для пошуку.
- 3) Натискає кнопку "Пошук".
- 4) Система здійснює пошук та відображає результати.

Вимоги:

- Можливість пошуку за ключовими словами та тегами.
- Відображення релевантних результатів пошуку.

2.5.6. Створення спільноти

Опис:

Користувач може створити нову спільноту, вказуючи її назву, опис та інші параметри.

Кроки:

- 1) Користувач переходить на екран створення спільноти.
- 2) Вводить назву, опис та інші параметри спільноти.
- 3) Натискає кнопку "Створити".
- 4) Система зберігає спільноту в базі даних та робить її доступною для інших користувачів.

Вимоги:

- Введення всіх необхідних даних.
- Перевірка унікальності назви спільноти.

2.5.7. Перегляд спільнот

Опис:

Користувач може переглядати список доступних спільнот.

Кроки:

- 1) Користувач переходить на екран спільнот.
- 2) Система завантажує та відображає список доступних спільнот.

3) Користувач може переглядати спільноти, скролити список, обирати спільноту для детального перегляду.

Вимоги:

- Відображення списку спільнот.
- Можливість перегляду деталей спільноти.

2.5.8. Управління профілем

Опис:

Користувач може редагувати свій профіль, змінюючи ім'я, аватар, теги інтересів, контактні дані тощо.

Кроки:

- 1) Користувач переходить на екран управління профілем.
- 2) Вносить зміни до свого профілю.
- 3) Натискає кнопку "Зберегти".
- 4) Система зберігає зміни в базі даних.

Вимоги:

- Можливість редагування всіх полів профілю.
- Перевірка наявності обов'язкових полів.

2.5.9. Публікація новин

Опис:

Користувач може публікувати новини в спільнотах, до якої він належить.

Кроки:

- 1) Користувач переходить на екран публікації новин.
- 2) Вводить текст новини, додає зображення або інші медіа-файли.
- 3) Натискає кнопку "Опублікувати".
- 4) Система зберігає новину в базі даних та робить її доступною для учасників спільноти.

Вимоги:

- Введення тексту новини.
- Можливість додавання медіа-файлів.

2.5.10. Отримання рекомендацій завдань

Опис:

Система автоматично підбирає завдання для користувача на основі його інтересів та попередньої активності.

Кроки:

- 1) Користувач переходить на головну сторінку.
- 2) Система аналізує профіль користувача та його активність.
- 3) Відображає рекомендовані завдання.

Вимоги:

- Механізм аналізу профілю та активності користувача.
- Відображення релевантних рекомендацій.

2.5.10.1. Алгоритм підбору рекомендованих завдань

Для підбору рекомендованих завдань на основі тегів ми використовуємо наступний алгоритм, що включає кілька етапів: збір даних, підготовка даних, розрахунок релевантності завдань та вибір найкращих завдань для користувача.

Крок 1: Збір даних

На цьому етапі збираються всі доступні дані про завдання та їх теги, а також інформація про користувача і його вподобання.

Збір завдань:

$Z = \{z_1, z_2, \dots, z_n\}$ – множина всіх завдань

$T(z_1) = \{t_{i1}, t_{i2}, \dots, t_{im}\}$ – множина тегів завдання z_1

Збір уподобань користувача:

$U = \{u_1, u_2, \dots, u_n\}$ – множина тегів, що цікавлять користувача

Крок 2: Підготовка даних

На цьому етапі відбувається підготовка даних для розрахунку релевантності завдань.

Визначення релевантності тегів завдання до уподобань користувача:

Визначається множина спільних тегів між тегами завдання $T(z_i)$ та уподобаннями користувача U :

$$S(z_i) = T(z_i) \cap U$$

Розрахунок ваги кожного тегу:

Вага кожного тегу $w(t_j)$ визначається на основі частоти його використання в завданнях та важливості для користувача. Для кожного тегу t_j з множини тегів T , вага розраховується за формулою:

$$w(t_j) = \frac{N(t_j)}{N_{total}} \times I(t_j)$$

Де $N(t_j)$ – кількість завдань, що містять тег t_j , N_{total} – загальна кількість завдань, а $I(t_j)$ – важливість тегу для користувача, що визначається як частота використання цього тегу в уподобаннях користувача.

Крок 3: Розрахунок релевантності завдань

На цьому етапі розраховується загальна релевантність кожного завдання для користувача на основі ваг тегів.

Визначення релевантності завдання:

Релевантність завдання $R(z_i)$ визначається як сума ваг спільних тегів:

$$R(z_i) = \sum_{t_j \in S(z_i)} w(t_j)$$

Крок 4: Вибір найкращих завдань

На цьому етапі обираються завдання з найвищою релевантністю для користувача.

Відбувається сортування завдань за зменшенням релевантності $R(z_i)$

І обираються N завдань з найвищою релевантністю для рекомендації користувачу.

Приклад розрахунку:

Припустимо, що у нас є такі завдання з тегами:

z_1 з тегами $\{t_1, t_2\}$

z_2 з тегами $\{t_2, t_3\}$

z_3 з тегами $\{t_1, t_3\}$

Та вподобання користувача $U = \{t_1, t_3\}$

Визначимо спільні теги:

$$S(z_1) = \{t_1\}$$

$$S(z_2) = \{t_3\}$$

$$S(z_3) = \{t_1, t_3\}$$

Припустимо, що ваги тегів $w(t_1) = 1.5$, $w(t_2) = 1.0$, $w(t_3) = 2.0$

Розрахунок релевантності:

$$R(z_1) = 1.5$$

$$R(z_2) = 2.5$$

$$R(z_3) = 1.5 + 2.0 = 3.5$$

Сортування та вибір топ-2 завдань:

z_3 з релевантністю 3.5

z_2 з релевантністю 2.0

Таким чином, користувачу будуть рекомендовані завдання z_3 та z_2

2.5.11. Перегляд нагород

Опис:

Користувач може переглядати свої нагороди, отримані за виконання завдань.

Кроки:

- 1) Користувач переходить на екран нагород.
- 2) Система завантажує та відображає список нагород.
- 3) Користувач може переглядати деталі кожної нагороди.

Вимоги:

- Відображення списку нагород.
- Можливість перегляду деталей нагород.

2.5.12. Вступ до спільноти

Опис:

Користувач може приєднатися до спільноти, яка його цікавить.

Кроки:

- 1) Користувач переглядає список спільнот.
- 2) Обирає спільноту та натискає кнопку "Приєднатися".
- 3) Система додає користувача до спільноти та надає доступ до її контенту.

Вимоги:

- Перевірка доступності спільноти для приєднання.
- Додавання користувача до списку учасників спільноти.

2.5.13. Вихід зі спільноти

Опис:

Користувач може покинути спільноту, до якої він належить.

Кроки:

- 1) Користувач переходить на сторінку спільноти.
- 2) Натискає кнопку "Вийти".
- 3) Система видаляє користувача зі списку учасників спільноти.

Вимоги:

- Видалення користувача зі списку учасників.
- Видалення доступу до контенту спільноти.

2.5.14. Участь у завданні

Опис:

Користувач може приєднатися до виконання завдання, яке його цікавить.

Кроки:

- 1) Користувач переглядає список завдань.
- 2) Обирає завдання та натискає кнопку "Приєднатися".
- 3) Система додає користувача до списку виконавців завдання та надає доступ до його деталей.
- 4) Користувач виконує завдання згідно з інструкціями.
- 5) Після успішного виконання завдання користувач отримує нагороду, якщо вона передбачена.

Вимоги:

- Перевірка доступності завдання для участі.
- Додавання користувача до списку виконавців.
- Надання доступу до деталей завдання.

3. РОЗДІЛ 3. Розробка програмного забезпечення

3.1. Загальні положення

У цьому розділі розглядається процес програмної реалізації мобільного додатку EchoAid, що включає детальний опис вибору програмних засобів, реалізації основних компонентів, архітектури системи, логіки роботи, інсталяції, налаштування та тестування. Кожен з цих аспектів є важливим для забезпечення успішного впровадження та функціонування додатку.

3.2. Вибір програмних засобів

Під час розробки мобільного додатку EchoAid були обрані і використані кілька ключових програмних засобів. Кожен з них виконував конкретні функції, необхідні для забезпечення ефективної, надійної та зручної роботи додатку.

Основним інструментом для створення додатку EchoAid став Android SDK. Він надав усі необхідні компоненти для розробки, тестування та налагодження додатку. Зокрема, Android Studio, яка є інтегрованою середою розробки для Android, забезпечила зручний і ефективний процес розробки. За допомогою Android Emulator вдалося тестувати додаток на різних версіях Android та різних конфігураціях пристроїв, а Android Debug Bridge (ADB) забезпечив можливість налагодження додатку на реальних пристроях.

Для виконання HTTP-запитів та обробки відповідей у форматі JSON була використана бібліотека Retrofit. Вона значно спростила взаємодію з сервером, забезпечивши здійснення HTTP-запитів для отримання та надсилання даних, таких як створення завдань, реєстрація користувачів та отримання списку завдань. Retrofit також автоматично перетворював JSON-відповіді з серверу в Java-об'єкти, що значно полегшило роботу з даними.

Room, як частина архітектурних компонентів Android Jetpack, використовувався для роботи з базою даних SQLite. Він забезпечив збереження інформації про користувачів, завдання, спільноти та інші дані локально на пристрої. Завдяки Room вдалося виконувати CRUD-операції (створення, читання, оновлення, видалення) з використанням об'єктно-реляційного відображення (ORM). Інтеграція з LiveData дозволила автоматично оновлювати інтерфейс користувача при зміні даних у базі.

Для завантаження та кешування зображень у додатку використовувалася бібліотека Glide. Вона забезпечила ефективне завантаження зображень профілів користувачів, завдань та спільнот з мережі, а також їх автоматичне кешування, що покращило продуктивність додатку та зменшило навантаження на мережу. Glide також дозволила застосовувати різні трансформації до зображень, такі як обрізка, зміна розміру та закруглення кутів, що покращило вигляд зображень в інтерфейсі користувача.

Hilt, як бібліотека для впровадження залежностей (DI), значно спростила управління залежностями у додатку. Вона автоматично впроваджувала залежності у компоненти Android, такі як Activity, Fragment та ViewModel, що покращило модульність коду та його тестованість. Автоматичне створення та управління графами залежностей забезпечило зручність та безпеку роботи з залежностями у проекті.

Navigation Component, що також є частиною архітектурних компонентів Android Jetpack, використовувався для навігації між екранами у додатку. Він спростив визначення навігаційних шляхів та обробку переходів, включаючи анімації та навігаційні дії. Використання графічного редактора в Android Studio дозволило створювати графи навігації, що значно спростило процес проектування навігації.

Navigation Component також забезпечив автоматичне управління стеком навігації, що покращило користувацький досвід.

Використання цих програмних засобів та бібліотек забезпечило ефективну, надійну та гнучку розробку мобільного додатку EchoAid. Кожен з обраних інструментів виконував конкретні функції, що дозволило досягти високого рівня продуктивності та якості кінцевого продукту. Завдяки використанню сучасних технологій, EchoAid забезпечує користувачам зручний та надійний інструмент для організації та участі у волонтерських заходах.

3.3. Реалізація компонентів програмного забезпечення

У цьому підрозділі розглянемо реалізацію основних компонентів додатку EchoAid, включаючи функціональні можливості, такі як реєстрація користувача, вхід в систему, створення завдання, перегляд завдань та інші. Кожен з цих компонентів був реалізований з використанням обраних програмних засобів та бібліотек, описаних у попередньому підрозділі. Нижче наведені описи фрагментів коду для деяких функцій [26].

3.3.1. Реєстрація користувача

Реєстрація користувача є однією з ключових функцій додатку EchoAid. Користувач вводить свої основні дані, такі як ім'я, email та пароль. Після цього система перевіряє правильність введених даних та наявність облікового запису з таким email у базі даних. Якщо дані коректні, користувач отримує лист підтвердження на вказаний email. Після підтвердження email обліковий запис активується, і користувач може увійти в систему.

Опис фрагмента коду реєстрації користувача:

Використовується технологія Room для управління локальною базою даних на платформі Android. Room є частиною Android Jetpack, що забезпечує простий спосіб роботи з SQLite базами даних у додатках Android.

Клас, який представляє користувача, включає такі властивості, як ідентифікатор, ім'я, електронна адреса, пароль і статус верифікації електронної пошти. Цей клас позначений анотацією `@Entity`, що визначає його як таблицю в базі даних.

Інтерфейс для доступу до даних користувача позначений анотацією `@Dao`. Він містить методи для додавання нового користувача до бази даних та отримання користувача за його електронною адресою. Анотації `@Insert` і `@Query` вказують на відповідні SQL операції для взаємодії з базою даних.

База даних визначається у класі, який позначений анотацією `@Database`. Цей клас містить абстрактний метод для отримання DAO, що забезпечує доступ до таблиці користувачів. Для створення екземпляру бази даних використовується метод `Room.databaseBuilder`.

На інтерфейсі користувача є екран реєстрації, створений за допомогою XML макету. Активність, що відповідає за цей екран, ініціалізує елементи інтерфейсу, такі як поля для введення імені, електронної адреси та пароля, а також кнопку реєстрації. При натисканні кнопки викликається метод, який перевіряє введені дані на коректність. Якщо дані правильні, здійснюється перевірка наявності користувача з такою електронною адресою в базі даних. Якщо користувач з такою адресою вже існує, відображається повідомлення про помилку. Якщо ні, новий користувач створюється і зберігається в базі даних, після чого користувачу відправляється повідомлення про успішну реєстрацію.

3.3.2. Вхід в систему

Для входу в систему користувач вводить свій email та пароль. Система перевіряє правильність введених даних, здійснюючи запит до бази даних, і якщо дані коректні, користувач отримує доступ до свого облікового запису.

Опис фрагмента коду входу в систему:

Знову використовується технологія Room для управління локальною базою даних на платформі Android.

У цьому коді створюється активність для входу користувачів. Активність ініціалізує елементи інтерфейсу, такі як поля для введення електронної адреси та пароля, а також кнопку для входу. База даних налаштовується за допомогою методу `Room.databaseBuilder`, який створює екземпляр бази даних.

При натисканні кнопки входу викликається метод `loginUser()`, який зчитує введені користувачем дані. Введені дані перевіряються на коректність у методі `validateInputs()`. Якщо дані правильні, запускається новий потік, де здійснюється перевірка наявності користувача в базі даних за допомогою методу `getUserByEmail()`. Якщо користувач не знайдений або введений пароль не співпадає з паролем у базі даних, користувач отримує повідомлення про помилку. Якщо електронна адреса користувача не верифікована, також відображається відповідне повідомлення.

У випадку успішного входу, користувач отримує повідомлення про успішний вхід і відбувається перехід до головної сторінки додатку. Це реалізується за допомогою створення `Intent`, який запускає `MainActivity`, і завершення поточної активності за допомогою `finish()`.

3.3.3. Створення завдання

Користувачі можуть створювати нові завдання, вказуючи назву, опис, теги, нагороду та інші деталі. Реалізація цієї функції включала створення форми для введення даних про завдання, обробку введених даних на сервері та збереження завдання у базі даних за допомогою Room.

Опис фрагмента коду створення завдання:

Цей код реалізує функціональність відображення списку завдань у мобільному додатку, використовуючи RecyclerView для відображення елементів списку та Room для роботи з базою даних. Він складається з двох частин: адаптера для RecyclerView (TaskAdapter.java) та активності для відображення списку завдань (TaskListActivity.java).

TaskAdapter.java

TaskAdapter є адаптером для RecyclerView, який відповідає за відображення списку завдань. Адаптер використовує клас TaskViewHolder для зберігання посилань на елементи інтерфейсу кожного окремого завдання.

Конструктор TaskAdapter приймає список завдань, який буде відображатися. Метод updateTasks дозволяє оновлювати цей список і повідомляти адаптер про зміну даних для оновлення відображення.

Методи onCreateViewHolder та onBindViewHolder відповідають за створення та зв'язування вьюхолдерів з даними завдань відповідно. В onCreateViewHolder створюється новий екземпляр вьюхолдера, а в onBindViewHolder зв'язуються дані конкретного завдання з елементами інтерфейсу вьюхолдера.

В класі TaskViewHolder зберігаються посилання на текстові поля для назви та опису завдання, а також на зображення завдання. Метод bind зв'язує дані завдання з

відповідними елементами інтерфейсу. Для завантаження зображення використовується бібліотека Glide.

TaskListActivity.java

TaskListActivity є активністю, яка відображає список завдань за допомогою RecyclerView. В методі onCreate активність ініціалізує RecyclerView, встановлює для нього лінійний менеджер компоновки та створює адаптер TaskAdapter з порожнім списком завдань. Адаптер встановлюється в якості адаптера для RecyclerView.

База даних налаштовується за допомогою методу Room.databaseBuilder, який створює екземпляр бази даних. Метод loadTasks завантажує завдання з бази даних в окремому потоці і оновлює адаптер на головному потоці, щоб відобразити завантажені завдання.

3.3.4. Управління профілем

Користувачі можуть редагувати свій профіль, змінюючи ім'я, аватар, теги інтересів та контактні дані. Реалізація цієї функції включала створення інтерфейсу для редагування профілю, збереження змін у базі даних та оновлення інформації в реальному часі за допомогою LiveData.

Опис фрагмента коду управління профілем:

Цей код реалізує функціональність перегляду та редагування профілю користувача в мобільному додатку, використовуючи технологію Room для управління локальною базою даних на платформі Android.

ProfileActivity.java

Активність ProfileActivity відповідає за відображення та редагування профілю користувача. Під час створення активності (onCreate метод) відбувається ініціалізація елементів інтерфейсу, таких як поля для введення імені та тегів, зображення аватара і

кнопка збереження. База даних налаштовується за допомогою методу `Room.databaseBuilder`, який створює екземпляр бази даних.

Метод `loadUserProfile` завантажує профіль користувача з бази даних в окремому потоці. Дані користувача, такі як ім'я, теги та URL аватара, відображаються на екрані за допомогою відповідних елементів інтерфейсу. Для завантаження зображення аватара використовується бібліотека `Glide`.

Кнопка збереження (`btnSave`) ініціалізує метод `saveUserProfile`, який зчитує введені користувачем дані, перевіряє їх на коректність і зберігає оновлену інформацію в базу даних у окремому потоці. Якщо дані валідні, користувач отримує повідомлення про успішне оновлення профілю.

`UserDao.java` (Оновлений з методом `update`)

Інтерфейс `UserDao` розширено методом `update`, який дозволяє оновлювати дані користувача в базі даних. Крім цього, інтерфейс містить методи для вставки нового користувача (`insert`) та отримання користувача за електронною адресою (`getUserByEmail`). Анотації `@Dao`, `@Insert`, `@Update` та `@Query` вказують на відповідні SQL операції для взаємодії з базою даних.

У цьому підрозділі ми розглянули реалізацію основних компонентів додатку `EchoAid`, таких як реєстрація користувача, вхід в систему, створення завдання, перегляд завдань та управління профілем. Наведені фрагменти коду демонструють використання бібліотеки `Room` для взаємодії з базою даних, забезпечуючи зручність та надійність для кінцевих користувачів. Це лише частина всіх функцій, які використовуються в додатку. Повний функціонал включає багато інших аспектів, які забезпечують комплексне управління волонтерськими завданнями та взаємодію між користувачами.

3.4. Фінальна архітектура системи

3.4.1. Клієнтський додаток

Клієнтський додаток EchoAid був розроблений для платформи Android з використанням Android SDK. Основні функції додатку включають реєстрацію та авторизацію користувачів, створення та перегляд завдань, управління профілем користувача, а також взаємодію з іншими користувачами та спільнотами. Завдяки використанню стандартних компонентів Android було досягнуто зручності та інтуїтивності користувацького інтерфейсу.

Одним з ключових аспектів клієнтського додатку є реєстрація користувачів. Ця функція дозволяє новим користувачам створити обліковий запис, вводячи свої особисті дані, такі як ім'я, email та пароль. Після введення даних система перевіряє їх правильність та наявність облікового запису з таким email у базі даних. Якщо дані коректні, користувач отримує лист підтвердження на вказаний email, після чого його обліковий запис активується.

Для забезпечення взаємодії з сервером та базою даних, додаток використовує бібліотеку Retrofit, яка спрощує процес виконання HTTP-запитів та обробки відповідей у форматі JSON. Це дозволяє легко надсилати запити на реєстрацію користувача, авторизацію, створення завдань та отримання списку завдань. Використання Retrofit забезпечує зручність та ефективність при роботі з сервером [28].

Крім того, для роботи з локальною базою даних використовувалася бібліотека Room. Вона забезпечує збереження та управління даними про користувачів, завдання, спільноти та інші сутності прямо на пристрої користувача. Це дозволяє

додатку працювати автономно, без постійного підключення до інтернету, зберігаючи при цьому всі необхідні дані.

Важливою частиною інтерфейсу користувача є перегляд завдань. Користувачі можуть переглядати список доступних завдань, використовуючи фільтри пошуку для знаходження завдань за ключовими словами та тегами. Інтерфейс для перегляду завдань був реалізований з використанням RecyclerView, що дозволяє ефективно відображати великі списки даних. Завдяки бібліотеці Glide, завантаження та відображення зображень, пов'язаних із завданнями, здійснюється швидко та без проблем.

Управління профілем користувача також є важливою функцією додатку. Користувачі можуть редагувати свій профіль, змінюючи ім'я, аватар, теги інтересів та контактні дані. Інтерфейс для редагування профілю був створений таким чином, щоб забезпечити максимально простий і зрозумілий процес внесення змін. Використання LiveData дозволило автоматично оновлювати інтерфейс користувача при зміні даних у базі.

Окрім зазначених функцій, додаток EchoAid підтримує інтерактивність та соціальні взаємодії між користувачами. Вони можуть створювати спільноти, приєднуватися до них, публікувати новини та обмінюватися повідомленнями. Це забезпечує активну участь користувачів у житті спільнот, сприяє обміну досвідом та знаннями, а також підвищує залученість користувачів у волонтерську діяльність.

3.4.2. Серверна частина

Серверна частина системи EchoAid є ключовою ланкою, що забезпечує обробку запитів від клієнтського додатку та взаємодію з базою даних. Вона була реалізована з використанням RESTful API, що дозволяє клієнтському додатку надсилати HTTP-запити для виконання різних операцій, таких як створення нових завдань, отримання списку завдань, оновлення профілю користувача тощо. Сучасні фреймворки та бібліотеки, що використовувалися для розробки серверної частини, забезпечують надійність, масштабованість та безпеку системи [27].

Одним з основних завдань серверної частини є обробка запитів на реєстрацію та авторизацію користувачів. Коли користувач реєструється, сервер отримує запит з необхідними даними, перевіряє їх правильність, зберігає у базі даних та надсилає лист підтвердження на вказаний email. Після підтвердження email, сервер активує обліковий запис користувача, дозволяючи йому входити в систему та використовувати всі можливості платформи. Аналогічно, при авторизації користувача сервер перевіряє введені дані та, у випадку їх відповідності, дозволяє доступ до облікового запису.

Для реалізації цих функцій використовувалися сучасні технології та фреймворки, такі як Spring Boot. Це дозволило створити ефективну та легко масштабовану серверну частину, яка здатна обробляти велику кількість запитів одночасно. Spring Boot забезпечує зручну конфігурацію та налаштування, що значно спрощує розробку та підтримку серверного додатку.

Важливою частиною серверної архітектури є взаємодія з базою даних. Серверна частина отримує запити від клієнтського додатку, обробляє їх та виконує відповідні операції з базою даних, такі як створення нових записів, оновлення

існуючих або видалення. Для цього використовувалася реляційна база даних, яка забезпечує високу надійність та продуктивність. SQL-запити дозволяють ефективно управляти великими обсягами даних та виконувати складні операції з ними.

Крім того, серверна частина забезпечує обробку запитів на створення та управління завданнями. Коли користувач створює нове завдання, сервер отримує запит з даними про завдання, перевіряє їх правильність та зберігає у базі даних. Після цього нове завдання стає доступним для інших користувачів, які можуть його переглядати, брати участь у його виконанні та отримувати нагороди. Серверна частина також забезпечує можливість редагування та видалення завдань, що дозволяє користувачам ефективно управляти своїми проектами.

Для забезпечення безпеки даних та захисту від несанкціонованого доступу, серверна частина використовує сучасні методи автентифікації та авторизації. Це включає використання токенів для управління сесіями користувачів, шифрування даних та інші методи захисту. Таким чином, серверна частина забезпечує високу ступінь безпеки та конфіденційності даних користувачів.

Окрім основних функцій, серверна частина також забезпечує обробку запитів на управління профілями користувачів та спільнотами. Користувачі можуть редагувати свої профілі, змінювати налаштування, створювати та управляти спільнотами. Серверна частина отримує ці запити, обробляє їх та виконує необхідні операції з базою даних, забезпечуючи актуальність та цілісність даних.

3.4.3. База даних

База даних є ключовим компонентом системи EchoAid, що забезпечує збереження та управління всією необхідною інформацією. Вона включає таблиці для зберігання даних про користувачів, завдання, спільноти, нагороди та інші сутності. Для реалізації бази даних використовувалася реляційна база даних на серверній стороні та бібліотека Room на клієнтській стороні, що забезпечує зручний та ефективний доступ до даних.

На серверній стороні використовувалася реляційна база даних, яка забезпечує високу надійність та продуктивність. Вибір реляційної бази даних обумовлений її здатністю ефективно управляти великими обсягами даних та забезпечувати цілісність і консистентність даних. Завдяки підтримці SQL-запитів можна виконувати складні операції з даними, такі як створення нових записів, оновлення існуючих та видалення, що забезпечує гнучкість у управлінні даними.

Для зберігання даних про користувачів створюється таблиця, що включає такі поля, як ідентифікатор користувача, ім'я, email, пароль та статус підтвердження email. Це дозволяє ефективно управляти інформацією про користувачів, забезпечуючи можливість швидкого доступу до необхідних даних. Крім того, забезпечується захист паролів шляхом їх хешування перед зберіганням у базі даних, що підвищує рівень безпеки системи.

Дані про завдання зберігаються в окремій таблиці, що включає поля для ідентифікатора завдання, назви, опису, тегів, нагороди та іншої інформації. Це дозволяє користувачам створювати та управляти завданнями, переглядати детальну інформацію про них та брати участь у їх виконанні. Завдяки реляційній базі даних

можна легко здійснювати пошук завдань за різними критеріями, такими як ключові слова, теги або статус завдання.

Для управління спільнотами створюється таблиця, що містить інформацію про кожну спільноту, включаючи її ідентифікатор, назву, опис та інші параметри. Це дозволяє користувачам створювати нові спільноти, приєднуватися до існуючих та брати активну участь у їхньому житті. Реляційна база даних забезпечує можливість швидкого пошуку та фільтрації спільнот за різними критеріями, що сприяє зручному та ефективному управлінню спільнотами.

На клієнтській стороні використовувалася бібліотека Room, яка є частиною архітектурних компонентів Android Jetpack. Room забезпечує зручний інтерфейс для взаємодії з локальною базою даних SQLite, що дозволяє ефективно зберігати та управляти даними прямо на пристрої користувача. Це особливо важливо для додатку, який може працювати автономно, без постійного підключення до інтернету.

Room використовує об'єктно-реляційне відображення (ORM), що дозволяє працювати з даними у вигляді об'єктів Java, а не писати складні SQL-запити. Це значно спрощує процес розробки та забезпечує зручність при роботі з даними. Завдяки підтримці LiveData можна автоматично оновлювати інтерфейс користувача при зміні даних у базі, що забезпечує зручність та інтерактивність користувацького інтерфейсу.

Використання Room дозволяє зберігати локально на пристрої користувача інформацію про користувачів, завдання, спільноти та інші дані. Це забезпечує

можливість доступу до даних навіть у режимі офлайн, що є важливою перевагою для мобільних додатків. Коли з'єднання з сервером відновлюється, локальні дані можуть бути синхронізовані з сервером, забезпечуючи актуальність та консистентність даних.

3.5. Логіка роботи програмного забезпечення

3.5.1. Алгоритм реєстрації користувачів

Процес реєстрації користувачів починається з введення особистих даних, таких як ім'я, email та пароль. Після цього система перевіряє правильність введених даних та наявність облікового запису з таким email у базі даних. Якщо дані коректні і обліковий запис з таким email ще не існує, користувач отримує лист підтвердження на вказаний email. Після підтвердження email, обліковий запис користувача активується, і він може увійти в систему. Цей алгоритм забезпечує безпеку та запобігає створенню дубльованих облікових записів.

3.5.2. Алгоритм авторизації

Алгоритм авторизації включає перевірку введених користувачем даних для входу в систему. Користувач вводить свій email та пароль, після чого система перевіряє ці дані на відповідність збереженим у базі даних. Якщо дані коректні і email користувача підтверджений, користувач отримує доступ до свого облікового запису. У випадку неправильно введених даних або непідтвердженого email система виводить відповідне повідомлення про помилку.

3.5.3. Алгоритм створення завдання

Алгоритм створення завдання дозволяє користувачам додавати нові завдання у систему. Користувач вводить необхідні дані про завдання, такі як назва, опис, теги та нагорода. Після цього система перевіряє правильність введених даних та зберігає

нове завдання у базі даних. Завдання стає доступним для перегляду іншим користувачам, які можуть його виконати. Цей алгоритм забезпечує структурованість та зручність управління завданнями.

3.5.4. Алгоритм перегляду завдань

Алгоритм перегляду завдань дозволяє користувачам переглядати список доступних завдань з можливістю використання фільтрів пошуку. Користувач може шукати завдання за ключовими словами, тегами або іншими критеріями. Система обробляє запит користувача, виконує пошук у базі даних та повертає результати, що відповідають заданим критеріям. Це забезпечує швидкий та зручний доступ до необхідної інформації.

3.5.5. Алгоритм управління профілем користувача

Користувачі можуть редагувати свій профіль, змінюючи особисті дані, аватар, теги інтересів та контактні дані. Алгоритм управління профілем включає перевірку введених даних та їх збереження у базі даних. Після внесення змін профіль користувача автоматично оновлюється, і ці зміни стають доступними для інших користувачів. Це дозволяє користувачам підтримувати актуальність своєї інформації.

3.5.6. Алгоритм синхронізації даних

Додаток EchoAid підтримує можливість роботи в режимі офлайн, що означає, що дані можуть бути збережені локально на пристрої користувача. Алгоритм синхронізації даних забезпечує узгодження локальних даних з сервером, коли з'єднання з інтернетом відновлюється. Це включає відправлення нових або оновлених даних на сервер та отримання актуальних даних з сервера. Такий підхід забезпечує цілісність та актуальність даних у системі.

Логіка роботи програмного забезпечення EchoAid побудована таким чином, щоб забезпечити високу продуктивність, надійність та зручність у використанні для кінцевих користувачів. Використання сучасних технологій та підходів дозволило створити ефективну систему, яка відповідає потребам користувачів та забезпечує стабільну роботу додатку.

3.6. Інсталяція та налаштування

3.6.1. Інсталяція додатку через APK

На даному етапі мобільний додаток EchoAid розповсюджується через файл формату APK (Android Package). Це дозволяє користувачам встановити додаток безпосередньо на свої пристрої, не використовуючи офіційний магазин додатків Google Play. Інсталяція через APK є зручним способом розповсюдження додатку на початкових етапах його розробки та тестування.

Кроки для інсталяції додатку через APK:

- 1) Завантаження файлу APK: Користувачі можуть отримати файл APK, завантаживши його з офіційного веб-сайту EchoAid або отримавши посилання на завантаження через email.
- 2) Налаштування пристрою: Перед інсталяцією файлу APK, користувачам необхідно дозволити встановлення додатків з невідомих джерел. Для цього потрібно перейти до налаштувань пристрою, обрати розділ "Безпека" та активувати опцію "Невідомі джерела".
- 3) Інсталяція APK: Після завантаження файлу APK користувач має відкрити файл за допомогою файлового менеджера або браузера, після чого з'явиться запит на підтвердження інсталяції. Користувач повинен натиснути "Інстальювати" для завершення процесу.
- 4) Запуск додатку: Після успішної інсталяції користувач може знайти додаток EchoAid у списку встановлених додатків та запустити його.

3.6.2. Налаштування середовища виконання

Після встановлення додатку, користувачам можуть бути доступні додаткові налаштування для оптимальної роботи додатку. Це включає налаштування облікового запису, вибір мовних параметрів, налаштування повідомлень та інші параметри, що підвищують зручність використання додатку.

Кроки для налаштування додатку:

- 1) Налаштування облікового запису: Після першого запуску додатку користувачеві буде запропоновано зареєструвати новий обліковий запис або увійти до існуючого. Для цього необхідно ввести особисті дані, такі як ім'я, email та пароль.
- 2) Вибір мовних параметрів: Додаток EchoAid підтримує кілька мов, що дозволяє користувачам обирати мову інтерфейсу. Це можна зробити у налаштуваннях додатку, обравши потрібну мову зі списку доступних.
- 3) Налаштування повідомлень: Користувачі можуть налаштувати повідомлення додатку, обираючи, які типи повідомлень вони хочуть отримувати (наприклад, про нові завдання, повідомлення від спільнот тощо). Це допомагає користувачам залишатися в курсі важливих подій та активностей.
- 4) Інші налаштування: Додаток також може містити інші параметри налаштування, такі як налаштування приватності, управління даними, синхронізація з сервером тощо. Користувачі можуть налаштувати ці параметри відповідно до своїх потреб та вподобань.

3.6.3. Плани щодо розповсюдження через Google Play

У майбутньому планується розповсюдження додатку EchoAid через офіційний магазин додатків Google Play. Це дозволить значно розширити аудиторію користувачів та забезпечити зручний процес встановлення та оновлення додатку. Підготовка до розміщення у Google Play включає виконання вимог щодо безпеки та якості додатку, підготовку маркетингових матеріалів, таких як іконки, скріншоти та опис додатку, а також тестування на різних пристроях для забезпечення сумісності.

Публікація додатку у Google Play також передбачає регулярне оновлення та підтримку, що дозволить оперативно виправляти виявлені помилки, додавати нові функції та покращувати загальну якість додатку. Крім того, користувачі зможуть залишати відгуки та пропозиції, що сприятиме подальшому розвитку та вдосконаленню EchoAid.

3.7. Тестування програмного забезпечення

У цьому підрозділі розглянемо процес тестування мобільного додатку EchoAid, включаючи методи та інструменти, використані для забезпечення якості та стабільності роботи додатку. Тестування є важливим етапом розробки програмного забезпечення, що дозволяє виявити та виправити помилки, а також переконатися, що всі функції додатку працюють належним чином.

3.7.1. Процес тестування

Тестування мобільного додатку EchoAid пройшло кілька ключових етапів, кожен з яких сприяв забезпеченню високої якості та надійності додатку. Під час тестування були використані різні типи тестування, включаючи модульне, інтеграційне, системне тестування та тестування користувацького інтерфейсу.

Модульне тестування: На першому етапі тестування було проведено модульне тестування, яке включало перевірку окремих функцій та методів додатку. Для цього

використовувалися бібліотеки JUnit та Mockito. Модульне тестування дозволило виявити та виправити помилки на рівні коду, забезпечуючи правильну роботу кожного компонента додатку [29,31].

Інтеграційне тестування: Після успішного модульного тестування було проведено інтеграційне тестування, яке перевіряло взаємодію між різними модулями та компонентами системи. Інтеграційне тестування допомогло переконатися, що всі частини додатку працюють разом належним чином, забезпечуючи узгодженість даних та коректну роботу всіх функцій. Для цього використовувалися JUnit та Espresso.

Системне тестування: Наступним етапом було системне тестування, яке передбачало перевірку всієї системи як єдиного цілого. Це включало перевірку всіх функціональних можливостей додатку, забезпечення коректної роботи під навантаженням та тестування на різних пристроях. Для системного тестування використовувалися автоматизовані тести з використанням інструментів, таких як Appium.

Тестування користувацького інтерфейсу: Останнім етапом було тестування користувацького інтерфейсу, яке перевіряло зручність та коректність роботи інтерфейсу користувача. Тестування користувацького інтерфейсу допомогло переконатися, що всі елементи інтерфейсу відображаються правильно, працюють належним чином та забезпечують зручний досвід для користувачів. Для цього використовувалися інструменти, такі як Espresso та UI Automator [30,33].

3.7.2. Висновки за результатами тестування

Процес тестування мобільного додатку EchoAid пройшов успішно, забезпечуючи високу якість та стабільність роботи додатку. Під час тестування були

виявлені та виправлені численні помилки, що сприяло підвищенню надійності та продуктивності додатку.

Результати модульного тестування показали, що всі окремі функції та методи додатку працюють належним чином, без помилок та збоїв. Модульне тестування дозволило виявити та виправити помилки на ранніх етапах розробки, що знизило ризик виникнення проблем на наступних етапах.

Інтеграційне тестування підтвердило, що всі компоненти системи працюють разом належним чином, забезпечуючи узгодженість даних та коректну роботу всіх функцій. Це свідчить про високу якість розробки та правильну інтеграцію всіх частин додатку.

Системне тестування показало, що додаток стабільно працює під навантаженням, забезпечуючи високу продуктивність та швидкий відгук на дії користувачів. Додаток успішно пройшов тестування на різних моделях пристроїв та версіях операційної системи Android, що забезпечує його сумісність з широким спектром пристроїв.

Тестування користувацького інтерфейсу підтвердило зручність та інтуїтивність використання додатку. Всі елементи інтерфейсу відображаються правильно, працюють належним чином та забезпечують зручний досвід для користувачів. Це свідчить про високу якість дизайну та реалізації користувацького інтерфейсу.

4. РОЗДІЛ 4. Ергономіка інформаційних систем

4.1.Ергономіка ІТ

4.1.1. Вимоги до програмного забезпечення з точки зору користувача та проектування компонентів інтерфейсу

Програмне забезпечення EchoAid розроблялося з особливою увагою до потреб та очікувань користувачів, а також з використанням сучасних методів проектування інтерфейсу. Основними критеріями стали простота використання, швидкість виконання завдань, мінімізація помилок, висока стандартизація інтерфейсу та візуальна привабливість. Ці аспекти тісно переплітаються з процесом проектування компонентів інтерфейсу, що дозволило створити зручний та ефективний додаток.

Простота використання була одним з головних пріоритетів при розробці EchoAid. Інтерфейс мав бути інтуїтивно зрозумілим, що дозволяє користувачам швидко освоїти роботу з додатком без необхідності довгого навчання. Використання зрозумілих іконок, чіткої навігації та логічної структури екранних форм допомогло досягти цієї мети. При створенні дизайну використовувались інструменти для прототипування, що дозволило створити головний екран з доступом до основних функцій в один клік, забезпечуючи легкий доступ до основних інструментів та виконання завдань [34,36].

Швидкість виконання завдань є ще одним ключовим аспектом. У сучасному світі користувачі очікують миттєвої реакції від програмного забезпечення. Програма повинна бути оптимізованою для швидкої роботи, що включає мінімізацію кількості кроків для виконання завдань та швидку реакцію інтерфейсу на дії користувача. У процесі розробки створювались інтерактивні прототипи, що дозволяло тестувати швидкість виконання основних завдань і вносити необхідні корективи для підвищення продуктивності.

Мінімізація помилок була важливим аспектом при розробці інтерфейсу. Дизайн мав бути спрямований на зниження ймовірності виникнення помилок користувача через надання чітких інструкцій, попередження та підтвердження дій, а також

забезпечення зворотного зв'язку. Наприклад, було створено покрокові інструкції для створення нового завдання, що дозволяє користувачам уникнути помилок на кожному етапі.

Висока стандартизація інтерфейсу є ключовим елементом у забезпеченні зручності використання. Використання загальноприйнятих стандартів та принципів дизайну полегшує навчання користувачів. Були розроблені уніфіковані стилі, розташування елементів та знайомі патерни взаємодії, які створюють відчуття комфорту та передбачуваності. Це особливо важливо для користувачів, які часто працюють з різними додатками і цінують знайомі інтерфейсні рішення.

Візуальна привабливість також відіграє значну роль у сприйнятті програмного забезпечення. Естетичний дизайн не тільки приваблює користувачів, але й підвищує їхню задоволеність від використання додатку. Використання гармонійної колірної палітри, сучасної типографіки та високоякісної іконографіки створює позитивний візуальний досвід. Були створені детальні макети, що демонструють використання кольорів, шрифтів та іконок для створення привабливого інтерфейсу.

Процес створення дизайну включав кілька етапів. Спочатку проводилося дослідження та аналіз потреб користувачів, що дозволило визначити основні вимоги до інтерфейсу. На основі цього створювалися низько- та високоякісні прототипи, які тестувалися з реальними користувачами для збору зворотного зв'язку. Отримані дані використовувалися для вдосконалення дизайну та створення остаточних макетів [34,38].

Таким чином, врахування цих вимог та процес проектування інтерфейсу дозволив створити зручний, швидкий та привабливий додаток, який відповідає очікуванням сучасних користувачів та забезпечує ефективну взаємодію.

4.1.2. Параметри при розробці інтерфейсу користувача

При розробці інтерфейсу користувача для EchoAid було враховано декілька важливих параметрів, які мали на меті забезпечити максимальну зручність та ефективність використання додатку. Ці параметри охоплюють ергономічні цілі, показники якості, комфортність інтерфейсу, проблеми та особливості розробки, а також принципи реалізації.

Ергономічні цілі ставлять за мету досягнення максимальної зручності та ефективності використання додатку. Інтерфейс має бути розроблений таким чином, щоб мінімізувати час, необхідний для виконання завдань, і забезпечити легкість у навчанні та використанні. Наприклад, для досягнення цієї мети було вирішено використовувати прості та зрозумілі елементи керування, які дозволяють користувачам швидко знаходити потрібні функції та інструменти [34,36].

Показники якості є ключовим аспектом, який включає вимірювання таких параметрів, як час виконання завдань, кількість помилок, задоволеність користувачів та ефективність взаємодії з інтерфейсом. Наприклад, було проведено тестування з реальними користувачами для визначення, наскільки швидко вони можуть виконувати основні завдання, такі як створення нового завдання чи перегляд спільнот. Ці дані допомогли ідентифікувати проблемні моменти та вдосконалити дизайн.

Комфортність інтерфейсу включає адаптацію до різних умов використання, включаючи різні пристрої та розміри екранів. Інтерфейс має бути однаково зручним як на великому екрані комп'ютера, так і на маленькому екрані смартфона. Для цього було використано адаптивний дизайн, що дозволяє інтерфейсу автоматично підлаштовуватися під розмір екрану та забезпечувати зручність використання на будь-якому пристрої.

Розробка інтерфейсу не обходиться без вирішення певних проблем та врахування особливостей. Основні проблеми, з якими довелося зіткнутися, включають забезпечення інтуїтивної зрозумілості інтерфейсу, зниження кількості помилок користувачів та забезпечення швидкої реакції інтерфейсу. Кожна з цих

проблем була вирішена через ретельне планування, тестування та вдосконалення дизайну [35,37].

Принципи реалізації відіграють важливу роль у створенні якісного інтерфейсу. Включення сучасних методів та інструментів для створення зручного та функціонального інтерфейсу стало основою успішної розробки. Використання прототипування та тестування з користувачами допомогло створити інтерфейс, який відповідає високим стандартам якості та задовольняє потреби користувачів.

4.2. Дизайн додатку

4.2.1. Загальні положення

Дизайн додатку EchoAid був розроблений з урахуванням найсучасніших тенденцій і технологій, що дозволяють створити інтуїтивно зрозумілий та візуально привабливий інтерфейс. Основною метою дизайну було забезпечення зручності використання та підвищення ефективності взаємодії користувачів з додатком. Для реалізації цих завдань було обрано інструмент Figma, який надав можливість створити інтерактивні прототипи, тестувати їх на різних пристроях та отримувати зворотний зв'язок від користувачів.

Figma була обрана як основний інструмент для дизайну з кількох причин. По-перше, цей інструмент дозволяє працювати над проектом в режимі реального часу, що значно спрощує процес співпраці між дизайнерами та іншими членами команди. Це особливо важливо для великих проектів, де залучені багато людей з різними обов'язками [36,39].

По-друге, Figma забезпечує високу гнучкість і масштабованість дизайну. Інструмент дозволяє створювати адаптивні макети, які автоматично підлаштовуються під різні розміри екранів, що є критично важливим для мобільних додатків. Крім того, можливість створення інтерактивних прототипів у Figma дозволяє тестувати дизайн безпосередньо на користувачах, отримуючи цінний зворотний зв'язок і вносячи необхідні корективи на ранніх етапах розробки.

Використання Figma також забезпечує зручне управління компонентами інтерфейсу. Всі елементи дизайну можуть бути організовані в компоненти, що дозволяє легко вносити зміни в один елемент і автоматично оновлювати всі екземпляри цього елемента в проєкті. Це значно спрощує процес підтримки і оновлення дизайну.

Основною метою дизайну EchoAid було створення інтерфейсу, який би максимально відповідав потребам користувачів. Для цього проводилося детальне дослідження цільової аудиторії, що включало аналіз потреб та очікувань користувачів, а також вивчення існуючих рішень на ринку. Це дозволило визначити основні вимоги до інтерфейсу та розробити дизайн, який забезпечує високу зручність використання та привабливий зовнішній вигляд.

4.2.2. Основні елементи дизайну

Дизайн додатку EchoAid складається з декількох ключових елементів, які визначають його зовнішній вигляд та функціональність. До них відносяться колірна палітра, типографіка, іконографіка та компоненти інтерфейсу [34-37].

1) Палітра кольорів

Палітра кольорів була обрана таким чином, щоб забезпечити високий контраст та зручність використання додатку в різних умовах освітлення. Основні кольори включають відтінки синього та зеленого, які асоціюються з надійністю та безпекою. Допоміжні кольори, такі як сірий та білий, використовуються для створення візуальної ієрархії та акцентування важливих елементів інтерфейсу.



Рисунок 4.1. Палітра кольорів EchoAid

2) Типографіка

Для додатку було обрано сучасний та легко читаний шрифт, який забезпечує високу зручність сприйняття тексту на різних пристроях. Основний шрифт використовується для заголовків та основного тексту, а допоміжний шрифт – для підзаголовків та менш важливих елементів інтерфейсу.

Основний шрифт – a_FuturaBs (20, 90A17D, 100%)

Допоміжний шрифт 1 – Wix Madefor Display (16, 000000, 100%)

Допоміжний шрифт 2 – Wix Madefor Display (10, 000000, 50%)



Рисунок 4.2. Типографіка EchoAid

3) Іконографіка

Іконки є важливою частиною дизайну, оскільки вони допомагають користувачам швидко орієнтуватися в додатку та знаходити необхідні функції. Для EchoAid було створено набір унікальних іконок, які гармонійно вписуються в загальний стиль додатку та забезпечують зрозумілість і функціональність інтерфейсу.



Рисунок 4.3. Іконкографіка EchoAid

4) Компоненти інтерфейсу

Компоненти інтерфейсу, такі як кнопки, форми, картки та навігаційні елементи, були розроблені з урахуванням принципів зручності використання та візуальної привабливості. Кожен компонент був протестований на зручність використання та адаптований для роботи на різних пристроях.



Рисунок 4.4. Компоненти інтерфейсу EchoAid

4.2.3. Особливості дизайну EchoAid

Дизайн додатку EchoAid має кілька особливостей, які роблять його зручним та привабливим для користувачів.

1) Головний екран

На головному екрані користувач може побачити декілька компонентів, а саме «Новини волонтерства», які пишуть самі користувачи, «Термінова допомога» для завдань срок виконання яких добігає кінця, а також «Запити для вас» з рекомендаціями завдань. Також внизу є панель навігації з конпками «Дім», «Знайти завдання», «Створити завдання», «Спільноти» та «Профіль».

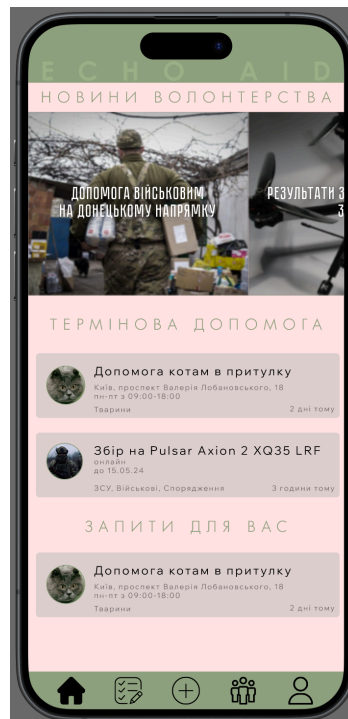


Рисунок 4.5. Головний екран додатку EchoAid

2) Сторінка завдань

На сторінці завдань користувач бачить повний список завдань з фільтрами та сортуванням пошуку.

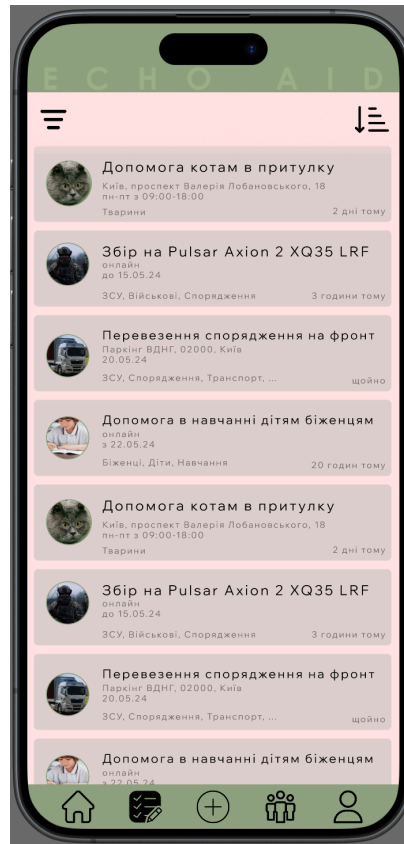


Рисунок 4.6. Список завдань EchoAid

а. Завдання детальніше



Рисунок 4.7. Завдання детальніше

3) Створення завдання

На сторінці створення завдання користувач може використовувати наступні поля: Назва, Теги, Фото, Опис, Місцезнаходження, Термін виконання, Контакти та Нагороди для створення завдання

The image displays two mobile application screens for creating a task. The top of both screens features the text "ECHO AID" in a light green font. The left screen contains the following sections:

- Вкажіть назву:** A text input field.
- Додайте теги:** A text input field.
- Додайте фото:** A section with a sub-note "Перше фото буде на обкладинці оголошення." and a large square area with a white plus sign for adding a photo.
- Додайте опис:** A large text input area with a character count "Введіть щонайменше 40 символів" and "0/9000".
- Вкажіть місцезнаходження:** A text input field.

The right screen contains the following sections:

- Вкажіть термін виконання:** A section with a sub-label "До:" and a date input field.
- Вкажіть ваші контакти:** A section with sub-labels "Контактна особа*", "Email-адреса", and "Номер телефону*", each followed by a text input field.
- Вкажіть нагороди:** A section with sub-labels "Назва нагороди" (with a text input field) and "Зображення нагороди" (with a square area containing a white plus sign).

At the bottom of both screens is a navigation bar with five icons: a home icon, a list icon, a plus icon, a group of people icon, and a person icon. A green button labeled "ОПУБЛІКУВАТИ" is located at the bottom of the right screen.

Рисунок 4.8. Створення завдання

4) Сторінка спільнот

На сторінці спільнот користувач може перейти до різних функцій спільнот натиснувши відповідні конпки.



Рисунок 4.9. Сторінка спільнот EchoAid

а. Створити спільноту



Рисунок 4.10. Створення спільноти

в. Моя спільнота



Рисунок 4.11. Моя спільнота

5) Профіль

В профілі користувач зможе переглянути інформацію про себе, подивитись свої нагороди, а також зайти до налаштувань, подивитись історію виконаних завдань, звернутись до онлайн-підтримки (модератора) та вийти з профілю.

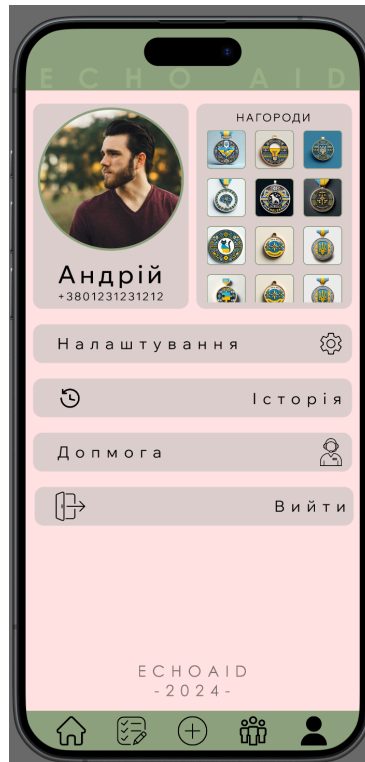


Рисунок 4.12. Особистий профіль EchoAid

а. Налаштування профілю

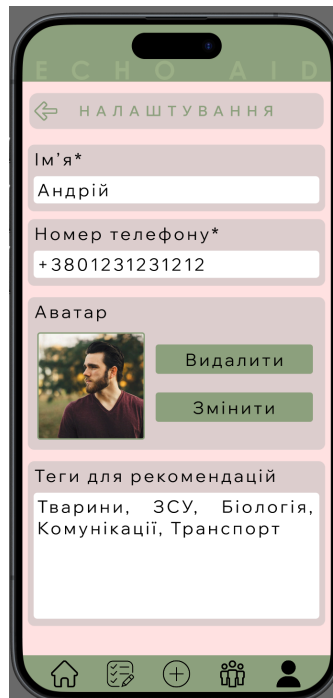


Рисунок 4.13. Налаштування профілю

в. Історія виконаних завдань

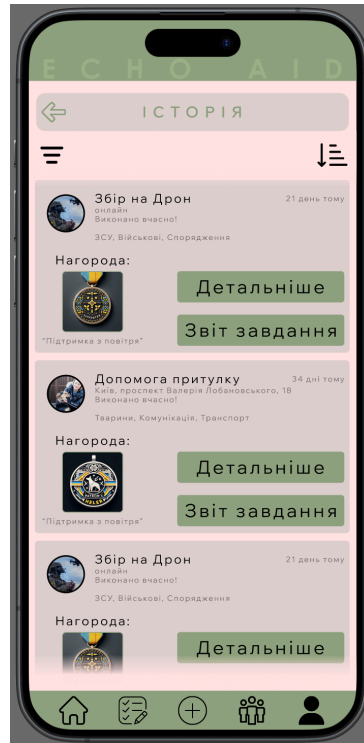


Рисунок 4.14. Історія виконаних завдань

б) Загальний вигляд дизайну додатку

В загалом додаток має наступні сторінки:

- Add Group Announcements Page
- Create Group Page
- Details Page
- Existing Groups Page
- Group Announcements Page
- Group Members Page
- Group Page
- Group Tasks Page
- History Page

- Home Page
- Login Page
- My Group Page
- News Page
- Profile Page
- Register Page
- Request Page
- Settings Page
- Tasks Page

Всі ці сторінки так або інакше пов'язані анімаціями одна з одною та допомагають користувачу з легкістю пересуватися по додатку. Також планується розробка додаткових сторінок для поліпшення користувацького досвіду та додавання нового функціоналу

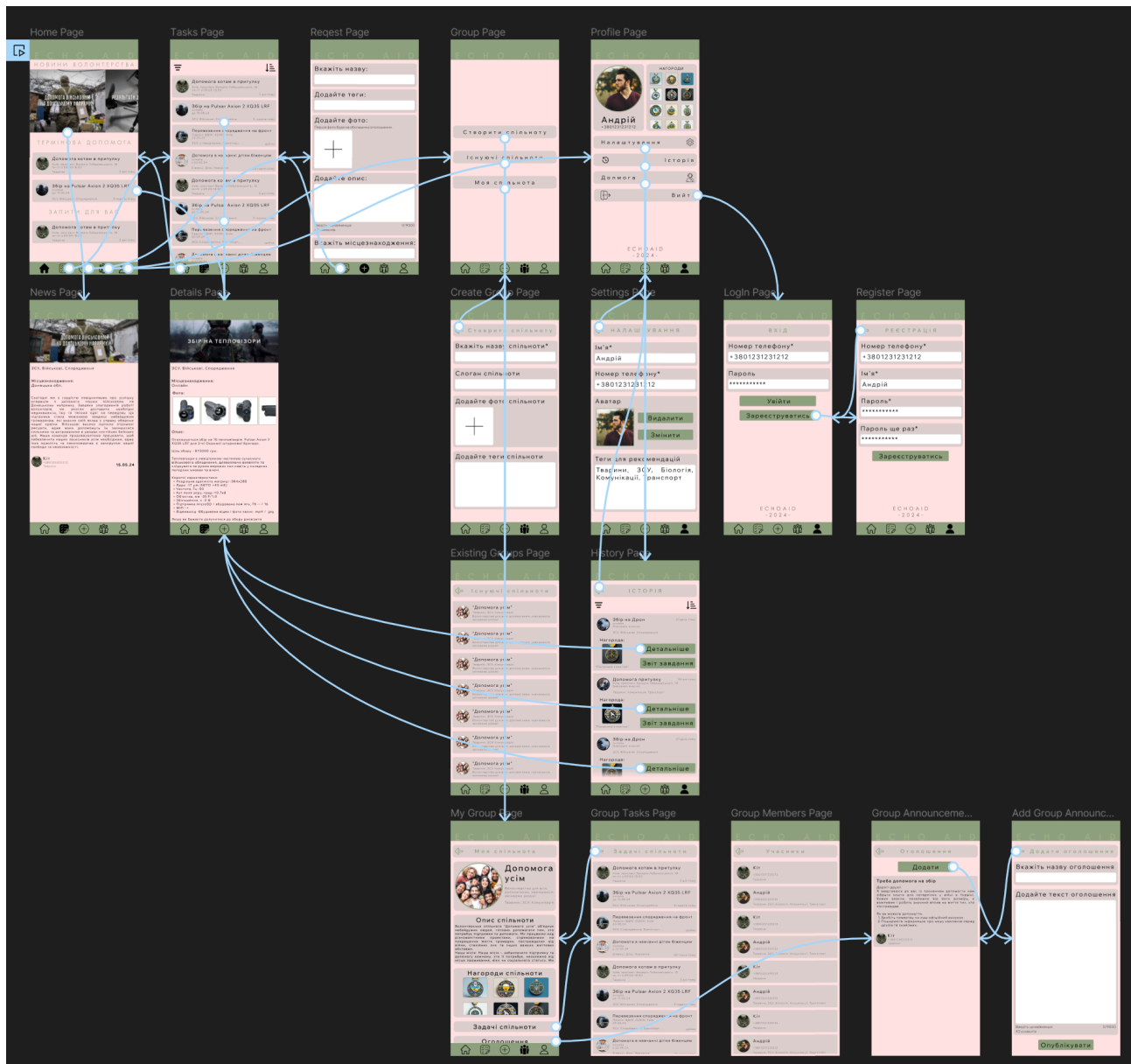


Рисунок 4.14. Загальний вигляд дизайну додатку

4.3. Техніко-економічне обґрунтування розробки

4.3.1. Резюме проекту

Додаток EchoAid є мобільним додатком для віртуального волонтерства, розробленим для полегшення організації та участі у волонтерських заходах. Основною метою проекту є підвищення залученості та мотивації волонтерів через використання гейміфікаційних елементів. Додаток надає користувачам можливість

створювати та виконувати завдання, приєднуватися до спільнот, отримувати нагороди та відстежувати свої досягнення.

4.3.2. Опис проектованого продукту або виду послуг

EchoAid – це мобільний додаток, який пропонує користувачам широкий спектр функціональних можливостей, створених для полегшення організації та участі у волонтерських заходах. Однією з основних можливостей є реєстрація та вхід в систему. Користувачі можуть легко створити свій обліковий запис або увійти в систему, використовуючи електронну пошту та пароль. Ця функція забезпечує безпеку та конфіденційність даних користувачів, що є критично важливим для будь-якого додатку, який працює з особистою інформацією.

Додаток також пропонує можливість створення та перегляду завдань. Користувачі можуть легко створювати нові волонтерські завдання, вказуючи необхідні деталі, такі як заголовок, опис, дата та час, категорія та навіть додавання зображень для кращої візуалізації. Перегляд завдань дозволяє користувачам знаходити та фільтрувати завдання за різними критеріями, такими як категорії, дата або популярність, що значно спрощує процес пошуку відповідних завдань для участі [38].

Участь у завданнях та отримання нагород є ще однією важливою функцією EchoAid. Користувачі можуть брати участь у різних волонтерських заходах та отримувати за це нагороди, що підвищує їх мотивацію та залученість. Нагороди можуть бути як віртуальними, так і реальними, що додає елемент гейміфікації та робить процес волонтерства більш захопливим.

Створення та управління спільнотами є ще однією унікальною можливістю EchoAid. Користувачі можуть створювати власні волонтерські спільноти, запрошувати до них інших користувачів та організовувати спільні заходи. Це сприяє розвитку волонтерського руху та створює умови для ефективної взаємодії між волонтерами.

4.3.3. Оцінка ринку збуту

Цільова аудиторія додатку EchoAid включає волонтерів, організаторів волонтерських заходів та представників волонтерських організацій. Враховуючи зростаючий попит на онлайн-платформи для волонтерства, особливо у кризових умовах, потенціал ринку є значним. EchoAid може залучити велику кількість користувачів, забезпечуючи їм зручний інструмент для організації та участі у волонтерських заходах.

4.3.4. Аналіз конкуренції

Основні конкуренти EchoAid включають існуючі волонтерські платформи, такі як UNITED24, Razom for Ukraine та Dobro.ua. Однак, конкурентні переваги EchoAid полягають у інтеграції гейміфікаційних елементів, міжнародній доступності та високій зручності використання. Ці особливості дозволяють EchoAid виділитися серед інших рішень на ринку та залучити більше користувачів.

4.3.5. Стратегія маркетингу

Стратегія маркетингу EchoAid охоплює ключові підходи для просування додатку та залучення користувачів. Основним інструментом будуть соціальні мережі, такі як Facebook, Instagram та Twitter (X), для поширення новин, історій успіху волонтерів, організації конкурсів та акцій [38].

Партнерства з волонтерськими організаціями підвищать довіру до додатку та залучать більше користувачів. Організації використовуватимуть EchoAid для організації заходів та рекомендацій своїм волонтерам.

Реклама у цифрових медіа через Google Ads та Facebook Ads допоможе досягти цільової аудиторії. Таргетована реклама приверне увагу користувачів, які шукають можливості для волонтерства.

Таким чином, стратегія маркетингу EchoAid включає використання соціальних мереж, партнерства та цифрову рекламу що забезпечить ефективне просування додатку.

4.3.6. План виробництва

Розробка EchoAid здійснюється за методологією Agile, що дозволяє забезпечити гнучкість у розробці та швидку адаптацію до змін. Основні етапи виробництва включають планування спринтів, розробку функціональності, тестування та релізи. Тестування проводиться на різних етапах розробки, включаючи альфа- та бета-тестування, що дозволяє виявити та виправити помилки до релізу.

4.3.7. Організаційний та юридичний план

Розробка EchoAid здійснюється за методологією Agile, що дозволяє забезпечити гнучкість у розробці та швидку адаптацію до змін. Основні етапи виробництва включають планування спринтів, розробку функціональності, тестування та релізи. Тестування проводиться на різних етапах розробки, включаючи альфа- та бета-тестування, що дозволяє виявити та виправити помилки до релізу.

4.3.8. Фінансовий план та стратегія фінансування

Фінансовий план проекту включає розрахунок бюджету на розробку, маркетинг та підтримку додатку. Основні джерела фінансування включають власні кошти, інвестори та краудфандинг. Для залучення інвесторів буде розроблено детальний бізнес-план та проведено презентації проекту [38].

4.3.9. Оцінка ризику та страхування

Аналіз можливих ризиків включає технічні, фінансові та ринкові ризики. Для мінімізації ризиків розроблено план управління, що включає стратегії для виявлення

та реагування на можливі проблеми. Страхування проекту дозволить зменшити фінансові втрати у разі настання непередбачених обставин.

ВИСНОВКИ

Розробка мобільного додатку EchoAid спрямована на вирішення актуальних проблем залученості та мотивації волонтерів шляхом впровадження ігрових елементів у неігровий простір. Використання гейміфікації дозволяє перетворити процес волонтерства на захоплюючий та мотивуючий досвід, що сприяє збільшенню кількості учасників та їхньої активності.

Дослідження показало, що впровадження ігрових механік, таких як бали, медалі, рівні та виклики, допомагає створити стимули для волонтерів, що підвищує їхню залученість та продуктивність. Гейміфікація також сприяє покращенню координації та співпраці між волонтерами, що є критично важливим для успішного виконання завдань.

Розробка додатку EchoAid включала детальний аналіз існуючих волонтерських платформ, збір вимог від користувачів, проектування архітектури системи та впровадження сучасних технологій для забезпечення інтуїтивно зрозумілого інтерфейсу та швидкої реакції на дії користувачів. Використання методології Agile дозволило швидко адаптуватися до змінних вимог та забезпечити постійне вдосконалення продукту на основі зворотного зв'язку від користувачів.

Основні результати роботи включають створення функціонального та привабливого мобільного додатку, який відповідає потребам волонтерів та організаторів волонтерських заходів. Додаток забезпечує ефективну організацію волонтерських заходів, підвищення мотивації волонтерів та зниження кількості помилок при використанні.

Загалом, розробка EchoAid демонструє, що впровадження ігрових елементів у неігровий простір може значно покращити ефективність та залученість у волонтерській діяльності, забезпечуючи при цьому високий рівень зручності та задоволення для користувачів.

Список використаних джерел

1. Jun Chen, Mo Liang. Play hard, study hard? The influence of gamification on students' study engagement. *frontiers*. [Онлайновий] <https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2022.994700/full>.
2. Georgios Lampropoulos, Antonis Sidiropoulos. Impact of Gamification on Students' Learning Outcomes and Academic Performance: A Longitudinal Study Comparing Online, Traditional, and Gamified Learning. *MDPI*. [Онлайновий] <https://www.mdpi.com/2227-7102/14/4/367>.
3. Kim, S., Song, K., Lockee, B., & Burton, J. *Enjoy learning like gaming*. Springer. 2018.
4. Deterding, S., Dixon, D., Khaled, R., & Nacke, L. *rom game design elements to gamefulness: Defining "gamification"*. In *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments (pp. 9-15)*. 2011.
5. Huotari, K., & Hamari, J. *A definition for gamification: Anchoring gamification in the service marketing literature*. *Electronic Markets*, 27(1), 21-31. 2017.
6. *UNITED24. Офіційний сайт*. [Онлайновий] <https://u24.gov.ua>.
7. *Razom for Ukraine. Офіційний сайт*. [Онлайновий] <https://www.razomforukraine.org>.
8. *Dobro.ua. Офіційний сайт*. . [Онлайновий] <https://dobro.ua>.
9. *HelpUkraine.Center. Офіційний сайт*. [Онлайновий] <https://helpukraine.center>.
10. *UN Volunteers (UNV) Online Volunteering Service. Офіційний сайт*. [Онлайновий] <https://www.onlinevolunteering.org>.
11. *VolunteerMatch. Офіційний сайт*. [Онлайновий] <https://www.volunteermatch.org>.
12. Beck, K., Beedle, M., van Bennekum, A., et al. *Manifesto for Agile Software Development*. 2001.
13. Schwaber, K., & Sutherland, J. *The Scrum Guide*. 2020.
14. Cohn, M. *User Stories Applied: For Agile Software Development*. Addison-Wesley Professional. 2004.
15. Robertson, S., & Robertson, J. *Mastering the Requirements Process: Getting Requirements Right*. Addison-Wesley Professional. 2012.
16. Fowler, M. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Professional. 2003.
17. Booch, G., Rumbaugh, J., & Jacobson, I. *he Unified Modeling Language User Guide*. Addison-Wesley Professional. 2005.
18. Lucidchart. [Онлайновий] 2024 p. <https://www.lucidchart.com/pages/?>.

19. Documentation, StarUML. [Онлайновий] <https://staruml.io>.
20. Developers, Android. *Android SDK*. [Онлайновий] 2024 р. <https://developer.android.com/studio>.
21. Documentation, Retrofit. [Онлайновий] 2024 р. <https://square.github.io/retrofit/>.
22. Library, Room Persistence. [Онлайновий] 2024 р. <https://developer.android.com/training/data-storage/room>.
23. Documentation, Glide. [Онлайновий] 2024 р. <https://www.glideapps.com/docs>.
24. Injection, Hilt Dependency. [Онлайновий] 2024 р. <https://developer.android.com/training/dependency-injection/hilt-android>.
25. Component, Navigation. [Онлайновий] 2024 р. <https://developer.android.com/guide/navigation>.
26. Nielsen, J. *10 Usability Heuristics for User Interface Design*. 1995.
27. Krug, S. *Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability*. New Riders. 2014.

28. *Android Developer Documentation*. [Онлайновий] developer.android.com.
29. *Spring Boot Reference Guide*. [Онлайновий] spring.io/projects/spring-boot.
30. *Retrofit Documentation*. [Онлайновий] square.github.io/retrofit.
31. *JUnit 5 User Guide*. [Онлайновий] junit.org/junit5/docs/current/user-guide.
32. *Espresso Testing Framework*. [Онлайновий] developer.android.com/training/testing/espresso.
33. *Mockito Documentation*. [Онлайновий] site.mockito.org.
34. Appium Documentation. [Онлайновий] appium.io/docs/en/about-appium/intro.
35. *UI Automator Documentation*. [Онлайновий] developer.android.com/training/testing/ui-automator.

36. Норман, Д. *Дизайн звичних речей*. місце видання невідоме : New York: Basic Books, 2002.
37. Cooper, A. *About Face 3: The Essentials of Interaction Design*. місце видання невідоме : Wiley Publishing, 2007.
38. Nielsen, J. *Usability Engineering*. місце видання невідоме : Morgan Kaufmann, 1994.
39. Тефтелев, С. Н. *Проектування інтерфейсів користувача*. місце видання невідоме : К.: Техніка, 2018.
40. Гордон, Дж. *Гейміфікація у бізнесі*. місце видання невідоме : М.: Альпина Паблішер, 2020.

41. *Figma Documentation*. [Онлайновий] [figma.com](https://www.figma.com).