

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Сумський державний університет

Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

«До захисту допущено»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

_____ (підпис)

19 грудня 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня магістр

зі спеціальності 122 - Комп'ютерних наук,

освітньо-професійної програми «Інформатика»

на тему: «Інформаційна технологія проектування веб-додатку для організації роботи волонтерської організації»

здобувача групи ІН.м – 24 Матюшенка Ярослава Павловича

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

Ярослав МАТЮШЕНКО

_____ (підпис)

Керівник,

асистент кафедри комп'ютерних наук,

кандидат фізико-математичних наук

Олександр ВЛАСЕНКО

_____ (підпис)

Суми – 2023

Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра комп'ютерних наук

«Затверджую»

В.о. завідувача кафедри

Ігор ШЕЛЕХОВ

(підпис)

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня магістра

зі спеціальності 122 - Комп'ютерних наук, освітньо-професійної програми «Інформатика»
здобувача групи ІН.м-24 Матюшенка Ярослава Павловича

1. Тема роботи: «Інформаційна технологія проєктування веб-додатку для організації роботи волонтерської організації.»

затверджую наказом по СумДУ від «06» грудня 2023 року №1412-VI

2. Термін здачі здобувачем кваліфікаційної роботи до 19 грудня 2023 року

3. Вхідні дані до кваліфікаційної роботи _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1) Аналіз проблеми предметної області, постановка й формування завдань дослідження.

2) Огляд технологій, що використовуються для розробки мобільних застосунків.

3) Практична реалізація інформаційної технології. 4) Аналіз результатів.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____

6. Консультанти до проєкту (роботи), із значенням розділів проєкту, що стосується їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання «06» листопада 2023 р.

Завдання прийняв до виконання _____

(підпис)

Керівник _____

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання	Примітка
1	<i>Аналіз проблеми предметної області, постановка й формування завдань дослідження</i>		
2	<i>Огляд технологій, що використовуються для для розробки мобільних застосунків.</i>		
3	<i>Практична реалізація інформаційної технології</i>		
4	<i>Оформлення пояснювальної записки до кваліфікаційної роботи</i>		

Здобувач вищої освіти _____

(підпис)

Керівник _____

(підпис)

АНОТАЦІЯ

Записка: 45 стор., 13 рис., 10 табл., 1 додаток, 18 джерел.

Об'єкт дослідження – процес розробки мобільного застосунку.

Мета роботи – розробка мобільного застосунку для центра волонтерів, що забезпечує оперативний обмін інформацією при організації заходів.

Метод дослідження – написання коду додатку, його тестування на помилки та відповідність визначеним вимогам. Тестування може включати як автоматизовані тести, так і ручне тестування користувачами.

Результати роботи: мобільний додаток, що знаходиться на стадії розробки, призначений для центра волонтерів. У додатку реалізовано функції головного меню, форму створення заходу, форму редагування, форму перегляду та редагування фотоматеріалів, можливість реєстрації волонтера-користувача.

ВЕБ-ДОДАТОК, ВОЛОНТЕРСЬКА ОРГАНІЗАЦІЯ,
ВЕБ-ПРОГРАМУВАННЯ, БАЗИ ДАНИХ, МОБІЛЬНІ ТЕХНОЛОГІЇ

ЗМІСТ

ВСТУП	5
1. ІНФОРМАЦІЙНИЙ ОГЛЯД	6
1.1. Техніко-економічна характеристика предметної галузі	6
1.2. Концептуальне моделювання мобільного додатка.....	7
1.3.Виявлення недоліків чинного бізнес-процесу	9
1.4. Постановка задачі	9
1.5. Розробка моделі бізнес-процесу	11
1.6. Аналіз відомих застосунків.....	13
1.7. Моделювання мобільного додатка для волонтерського центру	15
2. ПРОЄКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ	18
2.1 Вимоги до мобільного додатку.....	18
2.2 Моделювання мобільного додатка.....	21
2.3 Інформаційне забезпечення мобільного застосунку	25
2.4 Проєктування бази даних мобільного додатка	27
2.4.3 Фізичне моделювання бази даних мобільного додатка	29
3. ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ	31
3.1 Функціональні вимоги до мобільного додатку.....	31
3.2 Вибір технології розробки програмного забезпечення.....	35
3.3 Опис роботи мобільного додатка та тестування.....	36
ВИСНОВКИ.....	39
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	40
ДОДАТОК А.....	42

ВСТУП

Актуальність. Мобільний застосунок (англ. Mobile app) – програмне забезпечення, призначене для роботи на смартфонах, планшетах та інших мобільних пристроях. Більшість мобільних програм є стандартними й встановлені на пристрій, а решту програм можна завантажити й придбати з онлайн магазинів, таких як App Store, BlackBerry, App World, Google Play та інших. Напочатку, функціональні можливості мобільних програм обмежувалися можливістю швидко перевірки електронної пошти або просто пограти в ігри з достатньо примітивною графікою. Проте високий попит на мобільні пристрої призвів до розширення їх функціоналу в інших областях, таких як, GPS, спілкування, перегляд відео з використанням інтернету.

З початком повномасштабного вторгнення в Україну волонтерський рух набрав вагомого значення, тому логічним є те, що кількість волонтерських організацій зростає. У рамках волонтерського руху якого створюються волонтерські табори та міжнародні групи, що реалізують головну ідею волонтерства – об'єднати людей з різних країн у рамках проєкту для допомоги.

Об'єктом дослідження є аналіз технічних рішень та інструментів для розробки мобільних застосунків.

Предмет дослідження – технологія створення мобільного застосунку для організації роботи волонтерського центру.

Гіпотеза. Всупереч наявності аналогів, проєктування та розробка мобільного застосунку має сенс, оскільки його повного та загальнодоступного аналога на цей час не існує. Розроблений застосунок має забезпечити оперативний обмін інформацією при організації заходів.

Структура. Випускна кваліфікаційна робота складається із вступу, трьох розділів, висновків, списку використаних джерел та додатку.

1. ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1. Техніко-економічна характеристика предметної галузі

Волонтерство або волонтерська діяльність (від латів. voluntarius – добровільний) – це широке коло діяльності, включаючи традиційні форми взаємодопомоги та самопомоги, офіційне надання послуг та інші форми громадянської участі, що здійснюється добровільно на благо широкого загалу без розрахунку на грошову винагороду. Добровольці, з погляду закону України – фізичні особи, які здійснюють благодійну діяльність у формі безоплатного виконання робіт, надання послуг (добровольчої діяльності)[1].

Діяльність благодійних організацій потребує підтримки добровольців. Для організації добровольчої діяльності, з одного боку, створюються вакансії добровольців у благодійних організаціях, з іншого боку – формується коло підтримувальних організацій та сам добровольчий корпус учасників благодійної діяльності, які згодні брати участь у безоплатній праці на благо нужденних. Така безоплатна праця є різновидом філантропії (безкорисливого дарування на користь тих, хто цього потребує – людей, природи тощо)[2].

Поняттям добровольча (волонтерська) діяльність в українській мові часто підміняють поняття «суспільна діяльність», якою позначають будь-яку корисну діяльність на благо суспільства. Діяльність добровольців спрямована в першу чергу на допомогу верствам населення, що гостро цього потребують і не мають можливості допомагати собі самостійно (старість, безпритульність, інвалідність, стихійні лиха, соціальні катаклізми і багато чого ще).

Місія центру волонтерів: позиціонування центру волонтерів як соціально активного університету у місцевій спільноті.

Метою діяльності центру волонтерів є розвиток соціально-професійного волонтерства у місті та його включення до міжнародних волонтерських програм.

Основними завданнями роботи центру є[3]:

- об'єднання чинних волонтерських команд у рамках центру

волонтерського студентського руху;

- організаційний супровід створення нових волонтерських команд у навчальних структурних установах;
- розробка та реалізація освітньої програми навчання волонтерів;
- організація університетських, регіональних, всеукраїнських та міжнародних заходів та подій для розвитку волонтерського студентського руху;
- формування регіонального соціального замовлення студентським волонтерським об'єднанням;
- сприяння реалізації волонтерських проєктів на користь міста та регіону;
- створення умов для включення волонтерів міжнародні волонтерські програми та міжнародну взаємодію.

Культурно-масовий відділ займається організацією заходів та створенням умов для повноцінної та плідної роботи. Він займається роботою груп, спрямованою на всі аспекти роботи волонтерів[4].

Із аналізу діяльності об'єднання видно, що є завдання, яким необхідна автоматизація. Виходячи з цього, було вирішено розробити мобільний додаток, який допомагатиме добровольцям шукати заходи, у яких вони можуть брати участь, а творцям заходів відповідно добирати добровольців.

1.2. Концептуальне моделювання мобільного додатка

Концептуальне моделювання предметної області – це структуризація предметної області, на яку розробляється додаток. Метою є виявлення переліку та ієрархії предметів, об'єктів, факторів та явищ, повний набір яких дозволяє реалізувати поставлені перед додатком мети та завдання[5].

Концептуальна модель (змістовна модель) – це абстрактна модель, що визначає склад і структуру об'єкта, властивості елементів і причинно-наслідкові зв'язки, притаманні об'єкту, що аналізується, і суттєві для

досягнення цілей моделювання. У концептуальній моделі, зазвичай у словесній формі, наводяться відомості про природу та параметри (характеристики) елементарних явищ досліджуваного об'єкта, про вид та рівень взаємодії між ними, про місце та значення кожного елементарного явища в загальному процесі функціонування об'єкта[6].

Концептуальне моделювання виконується за допомогою графічної мови для візуалізації, специфікації конструювання та документування систем, у якій велика роль належить програмному забезпеченню – Unified Modeling Language (UML). За допомогою UML можна розробити детальний план майбутньої системи, у якій будуть міститися не лише концептуальні елементи, такі як системні функції та бізнес-процеси, але й особливості, такі як класи, написані спеціальними мовами програмування, схеми бази даних.

UML є графічною нотацією, яка призначена для моделювання та опису всіх процесів, що протікають у процесі розробки. Основу UML представляють діаграми, які різняться за типами та призначені для моделювання різних аспектів розробки.

Для моделювання бізнес-процесів інформаційно-аналітичної системи використовується інструмент для моделювання, аналізу документування та оптимізації бізнес-процесів Ramus. Він дозволяє проводити опис, аналіз та моделювання бізнес-процесів, а також будувати систему класифікації та кодування[5].

Метою моделювання є аналіз бізнес-процесів здійснення моніторингу організації волонтерських заходів з точки зору керівника та волонтерів.

На вхід процесу «Формування заходу» надходять дані про заходи, на виході виходить сформований захід. У ролі механізму виступає той, хто створює захід, тобто керівник. Як учасники виступають волонтери.

Декомпозиція контекстної діаграми включає такі процеси.

1. Процес «Ввести інформацію про захід» передбачає введення даних про захід до текстового редактора або електронної таблиці для подальшої її систематизації.

2. Процес «Обробити отримані дані» передбачає уточнення, систематизацію та впорядкування інформації про захід, який проводить волонтерський центр.

3. Процес «Проаналізувати отримані дані» передбачає аналіз обробленого сценарію заходу щодо помилок.

4. Процес «Сформуванню заходів» передбачає, власне, складання готової програми та цілі заходу.

Таким чином, було відображено стандартний процес формування заходу у волонтерському центрі.

1.3.Виявлення недоліків чинного бізнес-процесу

Процеси другого рівня «Обробити отримані дані» та «Сформуванню заходів», що входять до складу бізнес-процесу «Формування заходів волонтерського центру», мають такі недоліки[7]:

1. Неefективний документообіг, що виявляється у необхідності ручної обробки даних. Цей недолік може бути усунений шляхом автоматизації обробки даних за допомогою додавання до проєктованого додатка відповідних функцій.

2. Неefективний документообіг, який виявляється у необхідності ручного формування списку волонтерів. Цей недолік може бути усунений шляхом додавання до додатка функцій автоматизованого формування списку волонтерів, додавання доступу до керівника проєкту з метою його перевірки.

Таким чином, всі недоліки чинного бізнес-процесу виявлено, що дозволяє перейти до побудови моделі «ЯК ПОВИННО БУТИ».

1.4. Постановка задачі

Ведення моніторингу заходів та їх організація, а також звітність про них ґрунтується на паперових документах. З року в рік збільшується обсяг

паперової роботи, що створює складність моніторингу, знижує швидкість організації та проведення заходів. Найчастіше повільна обробка даних відділом із традиційним паперовим архівом знижує оперативність інших відділів. Певною мірою ці процеси поволі переходять на рівень електронного документообігу, проте не на стільки щоб перестати бути проблемою.

Для збільшення продуктивності необхідно виконати автоматизацію певних процесів, тобто перенести все навантаження по організації на мобільний додаток. Швидкість виконання завдання у розробленому додатку істотно має відрізнятися від пошуку у паперовій картотеці чи відомостях. Виходячи з величезного комплексу завдань, які виконує керівник волонтерського центру, автоматизація поставленого завдання має увійти в єдиний процес з автоматизації організації заходів. Для цього розглянемо бізнес-процеси діяльності керівника центру волонтерів, проаналізувавши виконувані ним процеси[8].

Мобільний додаток, що проектується, повинен забезпечувати підтримку організації заходів, підсумки проведених заходів, а також виконувати створення наступних документів: звіт проведеного заходу, із повним зазначенням місця проведення, датами проведення, доданими фотографіями і так далі. Додаткові функції та особливості мобільного додатка виражаються у автоматизації процесу організації заходів та скороченні тимчасових витрат за проведення підготовки заходів[9].

У зв'язку з необхідністю виконання поставленого завдання потрібно здійснювати організацію та моніторинг проведення заходу, що й робитиме пропонуваній мобільний застосунок.

Для досягнення цієї мети потрібно вирішити такі завдання:

- виконати аналіз предметної галузі та врахувати її особливості;
- спроектувати та реалізувати програмний продукт;
- дати економічне обґрунтування розробленого мобільного додатку;

Призначенням мобільного додатка є[8]:

- прискорення та автоматизація процесів обробки інформації;

- виключення розбіжностей або неточностей у розрахунках при організації заходу;
- скорочення часу обробки та отримання даних для організації та моніторингу заходу.

Тому необхідно зберігати всю важливу інформацію в електронному вигляді, щоб надалі її можна було будь-якої миті часу використовувати, наприклад, при квартальній звітності.

Мобільний застосунок, що розробляється, має полегшити комунікації і швидке оповіщення всіх учасників заходу.

У проєктованому застосунку розв'язувані завдання буде реалізовано відповідним програмним забезпеченням, яке матиме можливість виконання низки операцій, як-от реєстрація учасників, формування звітів, моніторинг заходу.

Окрім того будуть використані екранні форми з усіма необхідними полями виведення необхідної інформації, а також елементами керування (наприклад, кнопками), що виконують основні операції, такі як збереження або скасування операцій та навігації звісно що навігація додатком. Для забезпечення комфортної роботи будуть використані екранні форми, що дають змогу задавати різні діапазони значень при створенні заходу.

Джерелом оперативної інформації для реалізації вирішення поставлених завдань є вихідні дані волонтерів та внесена інформація для створення чергового заходу.

Для розрахунку підсумкової інформації будуть застосовуватись звіти, які формуватимуться залежно від мети заходу, що проводиться.

1.5. Розробка моделі бізнес-процесу

Для виділення та подальшого опису основних вимог, яким має задовольняти застосунок, було побудовано контекстну модель "ЯК ПОВИННО БУТИ" ("TO-BE"), яка відображає передбачуваний стан

предметної області. Тепер при обробці даних, а також при формуванні заходу застосовується автоматизована робота програми.

Для визначення вимог до програми, що впроваджується, скористаємося класифікацією FURPS. Ця класифікація є аббревіатурою, яка розшифровується наступним чином[10]:

- F – Functionality (функціональність);
- U – Usability (зручність використання);
- R – Reliability (надійність);
- P – Performance (продуктивність);
- S – Supportability (супроводження).

У таблиці 1.1 представлені функціональні вимоги, виявлені під час аналізу вищезазначених контекстів, а таблиці 1.2 – нефункціональні.

Таблиця 1.1 – Функціональні вимоги до додатка

№	Вимога	Статус	Корисність	Ризик
1	Додавання нового волонтера	Схвалено	Критична	Середній
2	Видалити існуючого волонтера	Схвалено	Критична	Середній
3	Редагувати існуючу подію	Схвалено	Критична	Середній
4	Різні варіанти створення заходів	Запропоновано	Критична	Середній

Таблиця 1.2 – Нефункціональні вимоги до системи

№	Вимога	Статус	Корисність	Ризик
Вимоги щодо зручності використання				
5	Кегль основного шрифту не нижче 12 пунктів	Схвалено	Критична	Низький
6	Дизайн інтерфейсу в єдиному стилі та колірній гамі	Схвалено	Критична	Низький
7	Елементи інтерфейсу повинні мати чітку структуру	Схвалено	Критична	Низький

8	Базовий дозвіл екрана смартфона 640×960	Запропоновано	Критична	Низький
Вимоги до надійності				
9	Час відновлення роботи додатку після відмови 24 год	Запропоновано	Критична	Середній
Вимоги до продуктивності				
10	Додаток повинен мати можливість забезпечити одночасну роботу мінімум 50 користувачів	Схвалені	Критична	Середній
11	Час відгуку програми не має перевищувати 1хвилини	Запропоновані	Критична	Середній
Вимоги до супроводу				
12	Усунення виявлених в додатка проблем протягом 24 годин	Запропоновано	Критична	Середній

Далі необхідно провести аналіз вже чинних ІТ-рішень щодо відповідності описаним вимогам.

1.6. Аналіз відомих застосунків

Аналіз найвідоміших інформаційних систем, що мають те чи інше відношення до автоматизації формування розкладу, необхідний для того, щоб точніше визначити подальші напрямки проектування, а також не повторити чужих помилок під час розробки. За підсумками проведеного аналізу необхідно визначити доцільність проектування та розробки нової інформаційної системи у предметній галузі, що розглядається.

Розглянемо три часто використовувані додатки з пошуком роботи, оскільки програми для пошуку волонтерів знайти не вдалося. Їх функціонал в цілому достатньо схожий на той, що нам необхідно реалізувати. Переваги додатку Work.ua:

1. на головній сторінці надана проста інструкція; є список збережених пошуків;

2. великий вибір роботи, що сортується за параметрами. Значних або критичних недоліків виявлено не було.

Додаток Robotia.ua, представляє більше міст у порівнянні з вищеназаним аналогом. Даний додаток один із найпоширеніших додатків України.

Його переваги:

- представлений вибір вакансії;
- огляд вакансії для міст з інших країн представлений іноземною мовою;
- є можливість заповнення анкети.

Urwork є одним із популярних додатків для пошуку роботи. Тут представлені вакансії по різних категоріях, а пошук максимально оптимізований: додаток сам підказуватиме варіанти написання назви вакансії, так щоб знайти їй відповідність у базі. Розширений пошук допоможе звузити область інтересу. Додаток містить велику базу вакансій та надає досить зручний інструментарій. У цьому додатку пошук можна проводити відразу по кількох містах, що розширює діапазон знайдених пропозицій і підвищує можливість знайти потрібну вакансію. Призначений більшою мірою для пошуку віддаленої роботи.

За підсумками проведеного аналізу необхідно визначити доцільність проектування та розробки нового додатка у предметній галузі, що розглядається.

Отже, запропонований до розробки додаток повинен мати можливість забезпечити одночасну роботу щонайменше 50 користувачам, у мережному режимі, час відгуку програми не повинен перевищувати 1 хвилини, усунення виявлених проблем має відбутися протягом 24 годин з моменту виявлення несправності.

Вимоги по дизайну: дизайн повинен бути в єдиному стилі, приємним на вигляд, не ускладнювати роботу програми, мати простий функціонал, зрозумілий будь-якому користувачеві.

1.7. Моделювання мобільного додатка для волонтерського центру

Під час розробки застосунку одним із найважливіших етапів є вибір мови програмування. Розглянемо кілька варіантів.

Для мови програмування Javascript можна виділити такі бібліотеки[11]:

- PhoneGap – безкоштовний open-source фреймворк для створення мобільних програм, створений Nitobi Software. Дозволяє створювати програми для мобільних пристроїв за допомогою JavaScript, HTML5 та CSS3 без знання «рідних» мов програмування (наприклад, Objective-C) та під усі мобільні операційні системи (iOS, Android, Bada тощо);

- ReactNative – розробка від Facebook, яка дозволяє писати під усі чинні платформи. Основна відмінність від PhoneGap, наприклад, що він використовує нативні компоненти системи, а не створює WebView компоненти;

- NativeScript – це бібліотека, що дозволяє робити кросплатформні програми, використовуючи XML, CSS, JavaScript. Native script вирішує те саме завдання, що і PhoneGap, але підходи у них різні. Phonegap використовує рушій браузера, щоб відобразити UI (фактично ви отримуєте веб-сторінку), NativeScript використовує нативний рендеринг, тобто елементи нативного UI.

Для мови програмування Python можна виділити такі бібліотеки:

- SL4A – бібліотека, яка дозволяє створювати та запускати скрипти, написані різними мовами сценаріїв прямо на Android-пристроях.

- Kivy – є бібліотекою для розробки додатків, що працюють на всіх платформах. Бібліотека розповсюджується з відкритим вихідним кодом. Бібліотека зосереджена на інноваційних інтерфейсах, призначених для пристроїв типу «touch». Бібліотека ґрунтується на Cython (C-розширення для Python). На цей момент підтримує Linux, Windows, Mac OS X та Android. Графічний двигун побудований на OpenGL ES 2.

Для мови програмування Ruby можна виділити такі бібліотеки:

- RubyMotion – це набір інструментів для розробки під iOS мовою Ruby.

Він складається з середовища виконання, яке реалізує функціонал Ruby всередині iOS.

– Ruboto – створення програм для Android пристроїв. Ця бібліотека використовує JRuby для запуску програм на Android

Для розробки було обрано платформу React, так як вона має більші можливості, про що більш детально буде розглянуто у наступному розділі.

1.7.1. Розробка об'єктної моделі мобільного додатка

У якості об'єктної моделі мобільного застосунку може виступати діаграма варіантів використання. Дана діаграма дозволяє виявити основні процеси, які у додатку та його взаємозв'язок. Крім того, діаграма варіантів використання сприяє виділенню функціональної структури цієї програми. На основі створеної діаграми складається план подальшої розробки програми.

На діаграмі представлені такі дійові особи (актори):

– Волонтер – особа, яка використовує додаток для пошуку заходів волонтерського руху;

– Керівник заходу – особа, яка використовує програми для створення заходу з метою залучення волонтерів.

У таблиці 1.3 наведемо коротку характеристику прецедентів (тобто самих варіантів використання) діаграми.

Таблиця 1.3 – Характеристика прецедентів

Прецедент	Характеристика
Переглянути розклад заходів	Перегляд волонтером заходів, що проводяться волонтерським центром
Ввести інформацію про проведені заходи	Введення керівником даних про місце, час та інші дані про проведення заходу
Обробка отриманих даних	Уточнення, систематизація та впорядкування інформації про захід
Аналіз оброблених даних	Аналіз обробленого заходу щодо помилок

Ввести інформацію про захід	Внесення нового заходу до списку керівником для пошуку волонтерами
Сформувати захід	Підготовка всіх аспектів заходу керівником

Таким чином, було описано основні технологічні етапи, які необхідно реалізувати для автоматизації чинного бізнес-процесу "Формування заходів волонтерського центру".

1.7.2. Розробка логічної моделі даних мобільного додатка

Для створення логічної моделі даних необхідно розробити її інфологічну модель, яка призначена для відображення складу інформаційних об'єктів, над яким проведено процедуру нормалізації та складається перелік інформаційних потреб мобільного додатка для створення та пошуку заходів.

За методологією IDEF1X як логічна модель даних мобільного додатка виступає діаграма «сутність-зв'язок». У мобільному додатку для створення та пошуку заходів буде кілька таблиць, одна з яких призначена для збереження текстових даних про заходи, а інші – для зберігання списку учасників[11].

2. ПРОЄКТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКУ

2.1 Вимоги до мобільного додатку

2.1.1 Вимоги до структури та функціонування мобільного додатка

Мобільний додаток має бути не просто красивим, головне завдання – розв'язувати проблеми користувачів. Бути зручним, практичним, простим у сприйнятті. Воно покликане полегшити систему пошуку кандидатів волонтерів на той чи інший захід. Інтерактивний доступ до програми не може бути ускладнений якимись технічними моментами, збоями, помилками[14].

Визначимо основні вимоги до майбутнього додатку. Вимоги описані методологією FURPS+ (таблиця 2.1). Вимоги визначають, що має бути реалізовано. Виділяються два основні типи вимог[15]:

1. Функціональна вимога – це принцип, за яким має функціонувати додаток (призначення додатку загалом).
2. Нефункціональні вимоги – особлива властивість або обмеження, що накладається на програму.

Вимоги до мобільного додатка, що розробляється, представлені в таблиці 2.1.

Таблиця 2.1 – Вимоги до додатка

ID	Вимога	Статус	Корисність	Ризик	Стабільність
Функціональні вимоги					
1	Додаток повинен підтримувати найпоширеніші веб-сервери, наприклад, Apache HTTP Server версії не нижче 2.0	Схвалені	Важлива	Середній	Середня
2	Додаток повинен встановлювати стабільне з'єднання зі сховищем даних, наприклад, базою даних MySQL	Схвалені	Критична	Середній	Середня

3	Додавати складені заходи до архіву	Схвалені	Важлива	Середній	Низька
4	Пошук заходів щодо дат	Запропоновані	Важлива	Середній	Низька
5	Можливість розгортання найпопулярніших операційних системах	Запропоновані	Критична	Критично	Низька
6	Наявність вільної ліцензії	Запропоновані	Критична	Середній	Низька
Вимоги щодо зручності використання					
7	Навігація повинна бути простою і зрозумілою в використанні	Схвалені	Критична	Низький	Низька
8	Здійснення контролю доступу до заходів	Схвалені	Критична	Низький	Низька
Вимоги до надійності					
9	Автономне створення резервної копії бази даних	Запропоновані	Критичне	Середній	Середня
Вимоги до продуктивності					
10	Час відповіді програми повинен становити не більше 5 секунд	Запропоновані	Важлива	Середній	Низька
Вимоги до підтримки					
11	Середній час відновлення працездатності додатків не більше 1 доби	Запропоновані	Критична	Низький	Низька
+ Обмеження					
Обмеження проєктування					
12	Управління МЗ здійснюється через мобільний пристрій, або ПК	Запропоновані	Критична	Середній	Середня
13	Для реалізації застосунку має використовуватися мова програмування java8 і вище	Запропоновані	Важлива	Низький	Середня

У мобільному додатку для центру волонтерів буде використовуватися триланкова архітектура клієнт-сервер, яка підтримується в Android studio.

Триланкова архітектура є сервером бази даних, яких зберігає всю інформацію, необхідну для роботи програми, мобільний додаток, відображає інтерфейс користувачеві, і сервер програми, що реалізує прикладну бізнес-логіку[15].

У розробці мобільного додатка необхідне використання сервера бази даних та клієнтське програмне забезпечення.

Метою автоматизації є:

- забезпечення швидкого обміну інформацією щодо організації заходів;
- пошуку даних шляхом впровадження бази даних за місце паперової картотеки;
- збирання персональних даних учасників заходу;
- реєстрація нових користувачів;
- швидке створення заходу з усією необхідною інформацією;
- створення фотогалереї;
- формування звітності

Вимоги до функціональності:

1. можливість створити новий захід;
2. проведення пошуку за вже готовими заходами;
3. можливість реєстрації у заході;
4. можливість додавання своїх фотоматеріалів;
5. можливість додавання коментарів;
6. можливість редагування інформації в описі;

Вимоги захисту цілісності та секретності[16]:

1. дані мобільного застосунку повинні розташовуватись на локальному сервері без доступу до зовнішньої мережі інтернет;
2. обробка персональних даних волонтера може здійснюватися виключно з метою забезпечення дотримання законів та інших нормативних актів, сприяння волонтерам у їх завданнях, навчанні, забезпечення особистої безпеки волонтерів, контролю кількості та якості виконуваної роботи та забезпечення збереження майна.

Періодичність розв'язання задачі – повсякденне використання.

Вибір архітектури вважається основним компонентом розробки застосунку, у роботі буде використовуватися триланкова архітектура клієнт-сервер, яка підтримується в системі «MySQL».

Архітектура мобільного додатка, що розробляється, для центру волонтерів представлена на рисунку 2.1.

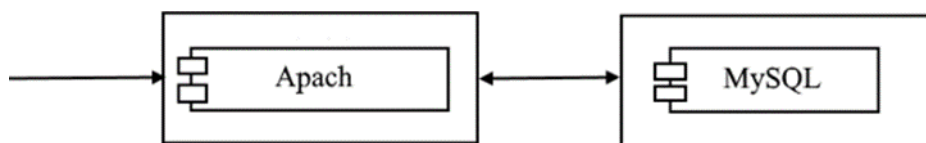


Рисунок 2.1 – Триланкова архітектура мобільного застосунку

Як «Тонкий клієнт» виступає мобільний пристрій. Для даної моделі, мобільний пристрій у більшій мірі відображає інтерфейс програми, який обробляється на сервері. Такий підхід зменшує навантаження на мобільний пристрій і часто використовується для забезпечення доступу до корпоративних додатків та даних через мобільні пристрої. Для зв'язку між компонентами є мережа інтернет. За здійснення процесів бізнес-логіки відповідає компонент Apache, він і є сервером програми. Сервер бази даних розташований на персональному комп'ютері керівника центру волонтерів[16].

Після визначення вимог до системи та вибору архітектури для реалізації програми, необхідно здійснити моделювання мобільного додатка, для визначення ключових функцій, які будуть реалізовані в застосунку, що проєктується.

2.2 Моделювання мобільного додатка

2.2.1 Функціональне моделювання мобільної програми

На рисунку 2.2 представлена функціональна модель проєктованого мобільного додатка, на якій відображені основні функції додатка, що проєктується.

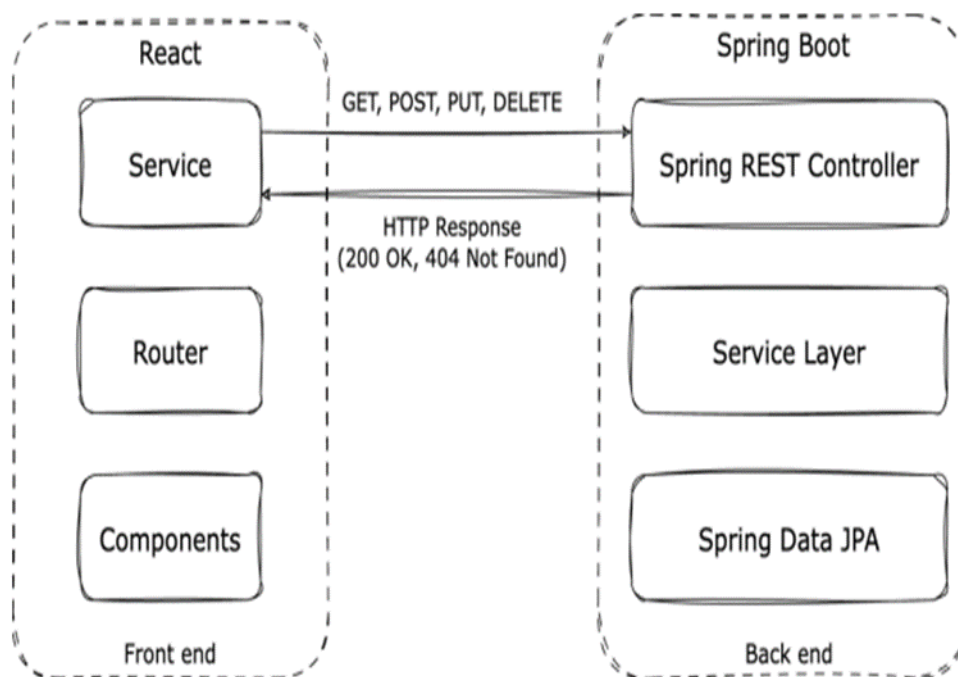


Рисунок 2.2 – Функціональна модель проектованого мобільного застосунку

Основні функції мобільного додатку:

- приймання заявок на реєстрацію у заході;
- вибір заходу для реєстрації;
- збір даних про зареєстрованих волонтерів;
- обробка даних;
- збір необхідної кількості учасників;
- формування звітів.

Спочатку керівник створює захід, у якому вказує точну дату, час та місце проведення заходу, додає до нього опис, та фото обкладинки заходу, потім волонтер-користувач додатка, який бажає взяти участь, реєструється на захід, після проведення якого він може поділитись фото матеріалами у розділ фото.

2.2.2 Логічне моделювання даних мобільного додатка

Логічне моделювання це здійснення перевірки функціонування логічної схеми. Основна мета полягає у здійсненні перевірки функції проектованого мобільного застосунку без повної реалізації на даному етапі розробки. Переваги даної моделі полягають у здатності перевіряти логічні функції в

додатках та їх тимчасові відносини, тобто як вони поведуться з плином часу та у відповідності до різних умов і подій. На рисунку 2.3 зображено діаграму послідовності, на якій відображено взаємодію об'єктів, а також обмін інформацією між ними.

Взаємодія між об'єктами та суб'єктами відбувається таким чином:

1. Керівник створює захід, заповнюючи всі необхідні дані, такі як дата, час проведення, місце проведення, опис, фото обкладинки.
2. Додаток створює готовий захід на основі внесених даних.
3. Волонтер-користувач отримує оповіщення про новий захід.
4. Волонтер-користувач подає заявку на участь у заході.
5. Керівник отримує дані про повноту списку учасників у режимі онлайн.
6. Після проведення заходу, волонтери-користувачі можуть завантажити фотоматеріали в альбом заходу.
7. Керівники волонтери можуть переглядати альбом з фотозвітом.

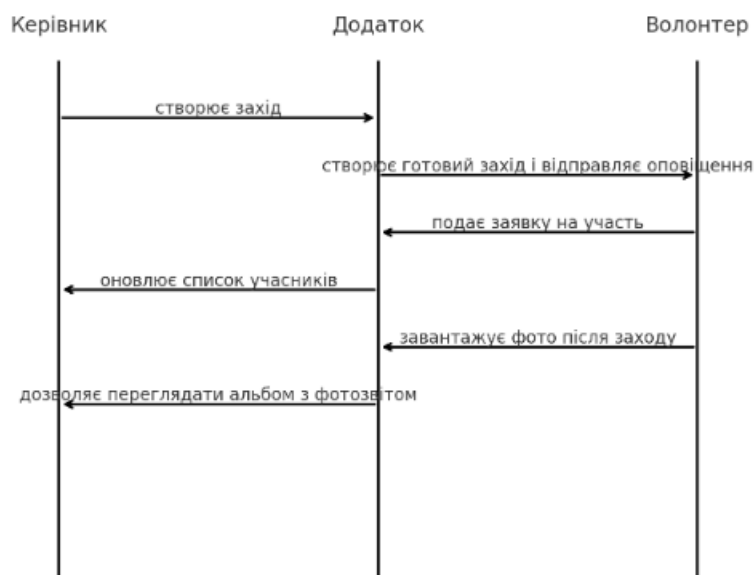


Рисунок 2.3 – Діаграма послідовності роботи мобільної програми

Таким чином, було здійснено перевірку логіки функціонування мобільного додатку. Описано послідовність організації роботи у додатку, та взаємозв'язок усіх учасників.

На рисунку 2.4 зображено діаграму активності. Ця діаграма є алгоритмом роботи програми.

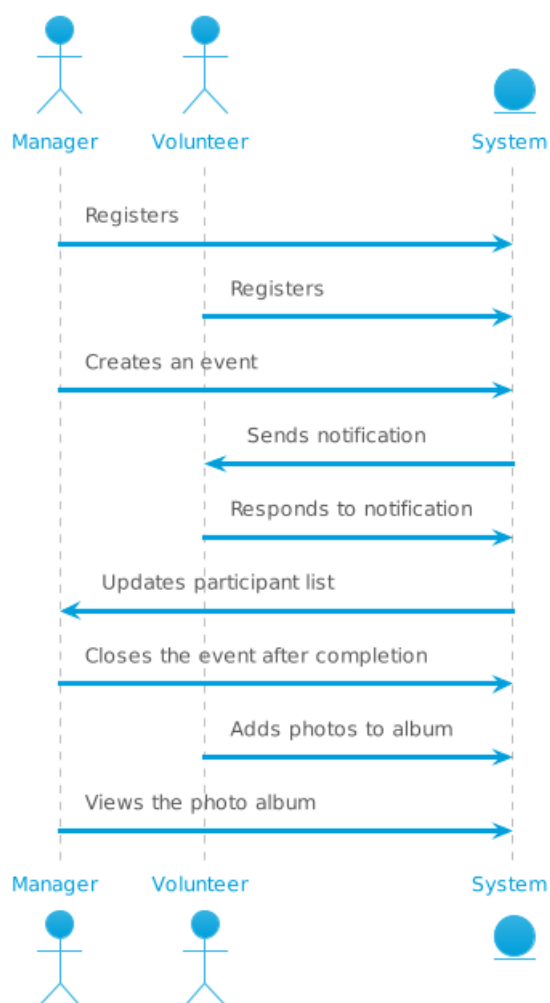


Рисунок 2.4 – Алгоритм роботи мобільного додатка

На діаграмі відображені ключові моменти роботи мобільного додатка, такі як: авторизація (керівник, він же адміністратор, волонтер, він же користувач), створення заходу адміністратором та заповнення даних, отримання оповіщення волонтером, обробка даних волонтера, заповнення повного списку реєстрованих, закриття заходу, а також додавання фотоматеріалів до загального альбому обох користувачів[18].

Наступним не менш важливим етапом логічного моделювання є складання діаграми класів мобільного додатку. Діаграма класів необхідна для відображення відносин між класами, тим самим представляючи логічну

модель створюваного проєкту.

Діаграма класів визначає різного роду статичні зв'язки та визначає типи об'єктів додатка, які існують між ними. На рисунку 2.5 продемонстровано діаграму класів, на якій представлені всі класи з атрибутами та методами.



Рисунок 2.5 – Діаграма класів

Таким чином, було описано основну логіку функціонування мобільного додатка, було виділено основні класи, які будуть використані у реалізації мобільного додатка для центру волонтерів. Для подальшої розробки застосунку необхідно здійснити аналіз інформаційного забезпечення.

2.3 Інформаційне забезпечення мобільного застосунку

У мобільному застосунку, що розробляється, буде присутня форма створення заходу, із заданими для неї фільтрами введення інформації. У таблиці 2.2 надано докладний опис уведення інформації у форму створення заходу.

Таблиця 2.2 – Опис полів форми створення заходу

Поле заповнення	Інформація, що вводиться, та формат введення
Кнопка додавання фотографії обкладинки заходів	Рисунок з мобільного пристрою формату jpg, jpeg, або gif
Послідовне введення цифрової інформації, кількість, що оголошує учасників заходу	Цифри від 1 до 100 залежно від потрібної кількості учасників
Час та дата заходу	Цифрове значення у форматі dd:mm:yy
Опис події, текстове поле	Текстова інформація про поточний захід, формат введення вільний.

Також у мобільному додатку, що розробляється, буде присутня форма авторизації, при запуску програми, підключеного до мережі інтернет. У таблиці 2.3 наведеної нижче, відображені поля для введення інформації під час авторизації користувача.

Таблиця 2.3 - Поля Авторизації

Поле заповнення	Введена інформація
Логін	Логін користувача, персонально створений їм під час реєстрації
Пароль	Пароль користувача, персонально створений ним під час реєстрації

Користувачам, не зареєстрованим у додатку, буде надано можливість реєстрації шляхом натискання на клавішу реєстрації, яка знаходиться у формі авторизації при запуску програми. У таблиці 2.4 наведеної нижче відображені поля для введення інформації при реєстрації нового користувача.

Таблиця 2.4 - Поля форми реєстрації

Поле заповнення	Введена інформація
Логін	Логін для авторизації у додатку Формат: Логін може складатися з латинських букв, цифр, знаків -, _, (Дефіс, нижнє підкреслення, крапка). Жодних інших символів, пробілів бути не повинно.

Пароль	Пароль для авторизації у додатку Формат: може складатися із латинських букв, цифр. Жодних інших символів, прогалін бути не повинно.
Ім'я	Справжнє ім'я реєстрованого
Прізвище	Справжнє прізвище реєстрованого
По-батькові	Справжнє по батькові реєстрованого
Номер для зв'язку	Мобільний номер реєстрованого

У пункті 2.3 було надано інформаційне забезпечення, необхідне для повноцінного функціонування програми.

2.4 Проектування бази даних мобільного додатка

2.4.1 Логічне моделювання бази даних мобільного додатка

Логічна модель служить для відображення логічних зв'язків між елементами даних незалежно від їхнього середовища зберігання та змісту. Логічна модель описує поняття предметної області, їх взаємозв'язок, і навіть обмеження на дані, накладені предметної області [14]. Логічна модель даних є початковим прототипом майбутньої бази даних проєктованого мобільного додатка.

Перед побудовою логічної моделі необхідно здійснити нормалізацію даних, представлених у концептуальній моделі, для скорочення суперечностей інформації, що зберігається, що міститься в БД. Для цього скористаємося «Нормальними формами».

На рисунку 2.6 представлена логічна модель даних, у якій відображено сутності з атрибутами, ключами сутностей та зв'язками між ними.

Сутність "Персональні дані" з атрибутами: "Прізвище", "Ім'я",

"По батькові", "Вік", "Телефон", "Керівник". Атрибут "Персональний номер" виступає в ролі первинного ключа. Сутність «Заходи» з атрибутами: «Номером прийняття», "Назва заходу", "Опис", "Місце", "Дата", "Фото обкладинки". Атрибут "Номер заходу" виступає в ролі первинного ключа, а атрибут "Фото обкладинки" виступає в ролі зовнішнього ключа.

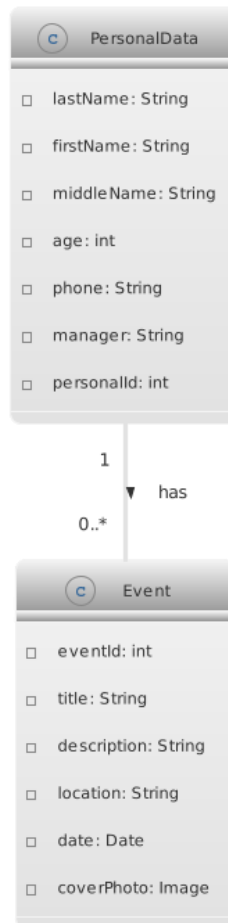


Рисунок 2.6 – Логічна модель даних

Таким чином, було описано реальні об'єкти предметної області, які відображають виділені сутності та їх взаємозв'язки між об'єктами.

2.4.2 Обґрунтування вибору системи керування базами даних

Під час розробки мобільного додатку однією з основних етапів є вибір системи управління базою даних (СУБД). Вибрана СУБД повинна відповідати вимогам, що висуваються до неї, і мати повну сумісність з додатком, розробленим під мобільну операційну систему Android[16].

Для зручнішого вибору СУБД була побудована порівняльна таблиця 2.5, у якій представлені критерії оцінки проведеного аналізу необхідних можливостей кожної СУБД для реалізації застосунку.

Таблиця 2.5 – Порівняльний аналіз СУБД

Критерії оцінки	Microsoft SQL Server 2016	Pervasive.SQL V12 SP1	Oracle 12C
Базові можливості мови SQL	+	+	+
Велика кількість механізмів зберігання даних	+	-	-
Відновлення баз даних з використанням журналу транзакцій	+	+	+
Контроль цілісності бази даних	+	+	+
Наявність графічного інтерфейсу користувача для управління базою даних	+	+	+
Наявність оптимізатора запитів	+	+	+
Низькі вимоги до ПК	-	+	+
Підтримка символів кирилиці на полях бази даних	+	+	+
Простота налаштування	+	-	-
Сума балів (+)	8	7	7

Для розробки мобільного додатка для центру волонтерів була обрана СУБД MySQL, оскільки вона задовольняє всі поставлені вимоги і має ряд переваг у порівнянні з аналогами.

2.4.3 Фізичне моделювання бази даних мобільного додатка

Наступним кроком розробки є фізичне проектування баз даних. На даному етапі розглядається організація файлів, індексів та основні відносини для забезпечення повноцінного доступу до даних, а також усі засоби захисту та пов'язані з цим обмеження цілісності.

У фізичній моделі сутності є таблицями, а екземплярами сутності є рядки таблиці. Атрибути виступають у ролі колонок таблиці. У фізичній моделі даних об'єкти містять тип даних, що залежить від обраної СУБД.

На рисунку 2.7 продемонстровано фізичну модель даних.

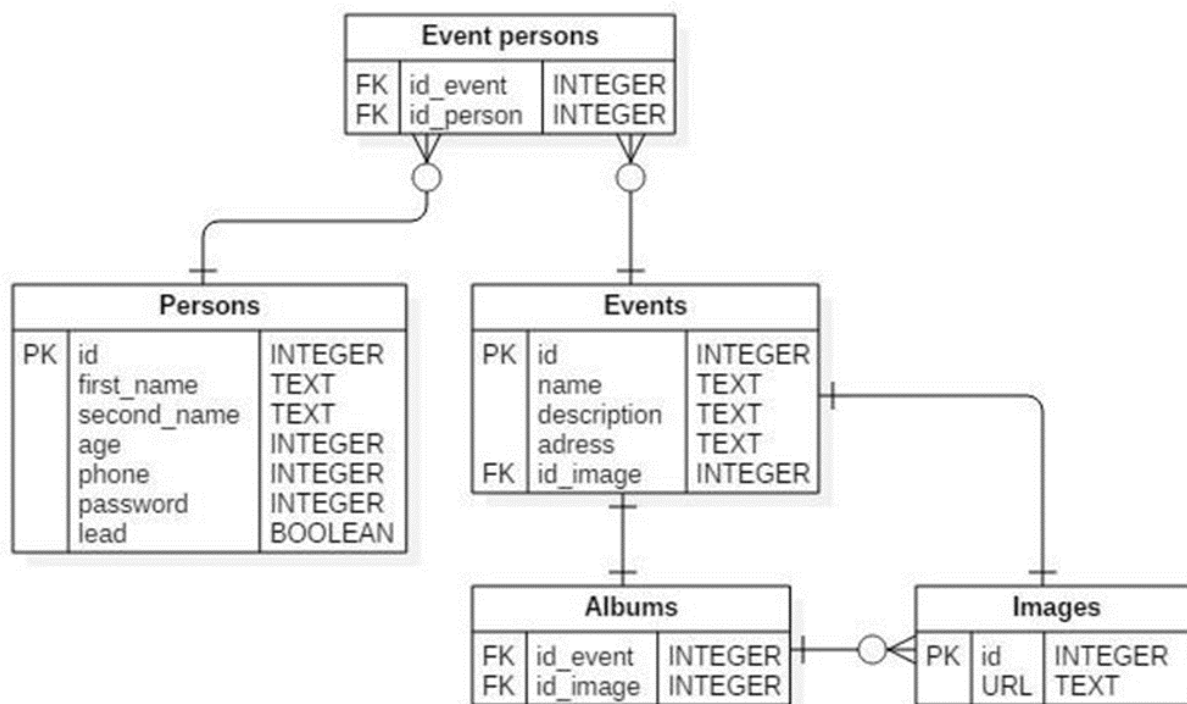


Рисунок 2.7 – Фізична модель даних

Фізична модель даних містить всю докладну інформацію, яка необхідна для реалізації бази даних у вибраній СУБД, а також у цій моделі містяться особливості синтаксису до необхідної СУБД[18].

Фізичне моделювання даних є останнім етапом реалізації проєктованої бази даних, на даному етапі розробник БД вибирає способи реалізації бази даних, що розробляється.

3. ПРОГРАМНА РЕАЛІЗАЦІЯ МОБІЛЬНОГО ЗАСТОСУНКУ

3.1 Функціональні вимоги до мобільного додатку

Функціональна структура програми повинна включати основні підпрограми, що виконують завдання збору, обробки отриманих даних, призначених для реалізації процесів збору даних від користувачів програми, приведення зазначених даних до виду, необхідного для наповнення підпрограми зберігання даних. Модуль для створення та візуалізації форм програми, який призначений для створення вікон програми, таких як форма заходу, форма авторизації та реєстрації.

Технічні вимоги до мобільного додатку, що розробляється:

1. Дозвіл при встановленні програми

Для виконання поставленого завдання додаток повинен мати необхідні дозволи для проведення відповідних операцій з пристроєм. Дотримуючись політики додатків Android OS, будь-який реалізований на даній платформі додаток при установці повинен запросити дозволи на види операцій, які проводяться програмою з мобільним пристроєм під час його роботи. Весь список дозволів, що запитуються, при установці виводиться на екран для користувача, щоб той міг вирішити, чи варто встановлювати цю мобільну програму на свій пристрій.

У випадку програми, яка розробляється виходячи з даного завдання, будуть запитуватися дозволи на роботу з мережевими підключеннями (забезпечується доступ до бездротових мереж для зв'язку з сервером і синхронізацією даних між сервером БД і сховищем на мобільному пристрої) і на доступ до файлової системи (для збереження налаштувань програми, для доступу до сховища).

Усі основні налаштування, властивості та запити на дозволи мобільного додатка, розробленого за допомогою Android Studio, описані у файлі AndroidManifest.xml та надані у форматі XML.

Дозволи мобільного додатку записуються розробником безпосередньо в цей файл. На рисунку 3.1 наведено дозволи, описані у файлі AndroidManifest.xml.

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Рисунок 3.1 – Дозволи для мобільного додатка, що розробляється

2. Вимоги до користувачів програми:

- кількість користувачів визначається поточними потребами центру волонтерів;
- користувачі (волонтери) повинні мати базові навички роботи з мобільною операційною системою Android будь-якої з версій.

3. Вимоги до ергономіки та технічної естетики. Вимоги до зовнішнього вигляду:

- реалізація графічного режиму;
- простий, зрозумілий і зручний інтерфейс, що не вимагає спеціальних знань.

4. Вимоги до інформаційної безпеки:

- обов'язково має бути забезпечений захист даних від краху при аваріях та збоях;
 - збереження інформації має виконуватися на наступних подіях:
 - відключення живлення на сервер баз даних;
 - відмова ліній зв'язку (мережі чи телефонії);
 - відмова апаратури сервера (периферії).

Засобами забезпечення безпеки інформації при будь-яких аваріях та збоях під час експлуатації сервера є:

- змінні, оптичні носії інформації;
- своєчасне створення резервної копії бази даних;
- створення резервної копії програмного забезпечення.

5. Вимоги до антивірусного захисту:

Для запобігання псуванню даних, включаючи несанкціонованого доступу до них, має використовуватися якісне, сучасне антивірусне забезпечення, що включає якісний між мережевий екран.

Основні вимоги до якісного антивірусного ПЗ:

- наявність величезної антивірусної бази з метою запобігання безлічі мережевих атак;
- можливість постійного аналізу всіх файлів, що передаються робочій машині.

6. Вимоги щодо стандартизації та уніфікації:

Для роботи з БД має використовуватися структурована мова запитів SQL. Зараз діє стандарт, прийнятий у 2003 році (SQL: 2003) з невеликими модифікаціями, внесеними пізніше (SQL: 2008).

7. Вимоги до програмного забезпечення:

Програмне забезпечення, що використовується при розробці, повинні мати широке поширення, бути загальнодоступним і використовуватися в промислових масштабах. Базовою платформою має бути операційна система Microsoft Windows 7-10 VLSC x64. Як мову програмування використовуватиметься React Native. Як мову запитів до бази даних використовуватиметься мовою SQL, а ролі СУБД – MySQL.

Процес декомпіляції потребує деякої кількості часу. Залежно від розміру БД необхідно забезпечити відповідну кількість системних ресурсів до виконання всіх етапів декомпіляції. Процес декомпіляції має проводитися робочому місці адміністратора. Рекомендується одна з наступних мінімальних конфігурацій, поданих у таблиці 3.1.

Таблиця 3.1 – Системні вимоги до ПК

Конфігурація	Модель – частота Гц (не менше)	Оперативна пам'ять (не менше, Мбайт)	Вільна пам'ять на системному диску (не менш, Гбайт)	Операційна система
Робоча станція адміністратора	Intel Core i5-2500	2048	30	Windows XP SP23 Windows 7
Робоче місце адміністратора на термінальному сервісі	Intel Xeon- 2600	2048	30	Windows Server 2008 R2

У таблиці 3.2 подано вимоги до сервера бази даних. Вимоги для двох рівнів архітектури такі самі, як і для трирівневої архітектури.

У разі необхідності збільшення кількості робочих місць рекомендується додати до системи сервер(и) додатків, що дозволить перейти від дворівневої архітектури до трирівневої. Також допускається паралельна інтеграція сервера бази даних та сервера додатків на одному ПК для досягнення необхідної продуктивності.

У разі необхідності збільшення користувачів більш як 10 рекомендується використовувати сервер бази даних із RAID-масивами, а для отримання складної звітності рекомендується додаткове встановлення сервера звітів.

Рекомендується сервер бази даних та сервер додатків з'єднати між собою високошвидкісним каналом зв'язку (1 Гбіт/сек).

Таблиця 3.2 - Системні вимоги до сервера

Конфігурація	Число користувачів	Модель (не нижче)	Оперативна пам'ять (не менше, Мбайт)	Вільна дискова пам'ять (не менш, Гбайт)
Мінімальна для сервера	-	PIII-933	512	1

Рекомендована для сервера	До 10 До 25 До 50	PIV-2600 2x PIV-2600 2x PIV-3200	1 Гбайт 2 Гбайта 2 Гбайти	2
---------------------------	-------------------------	--	---------------------------------	---

Було описано основні вимоги до апаратно-програмного забезпечення мобільного додатка, що розробляється, для центру волонтерів.

3.2 Вибір технології розробки програмного забезпечення

Android SDK надає розробнику великі можливості для роботи з мобільними пристроями на базі операційної системи Android OS за допомогою використання бібліотек, що підтримуються всіма пристроями даної операційної системи, дозволяючи охоплювати більшу частину ринку мобільних пристроїв, попри величезну фрагментацію ринку в плані виробників та підтримуваних систем, включаючи різні версії SDK

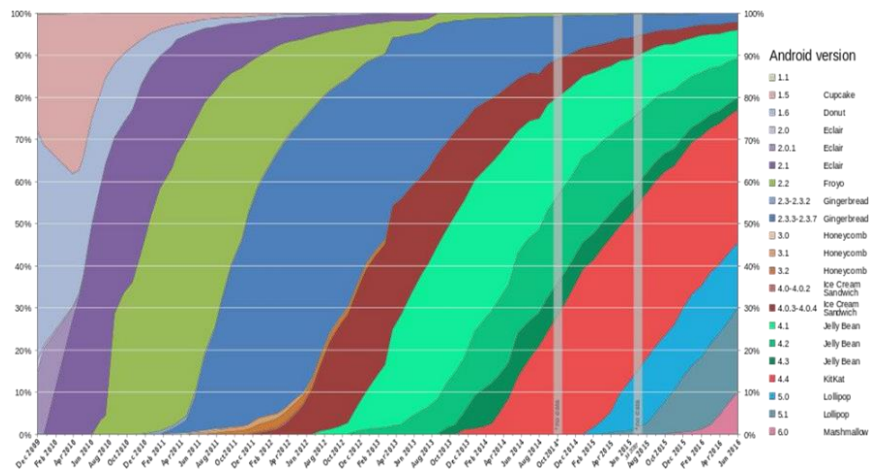


Рисунок 3.2 – Глобальне використання версій операційної системи Android.

Нині розробниками операційної системи Android рекомендовано використання Android SDK версії не нижче 19 (Android 4.4 Kitkat), але з розвитком операційної системи очікується чергове підвищення рекомендованої версії API. Однак, застарілі функції та бібліотеки не виводяться з використання при виході нових версій; у цьому випадку для

застарілих функцій закінчується підтримка, але у пакети, що виходить, вони все ще входять. Завдяки цьому програми, розроблені з використанням більш ранніх версій Android SDK, здатні запускатися та коректно працювати на сучасних версіях операційної системи Android OS.

На рисунку 3.2 показано розподіл встановлених на мобільних пристроях всіх версій ОС Android.

3.3 Опис роботи мобільного додатка та тестування

Розроблений мобільний додаток дозволить користувачам бачити майбутні події та заходи, які проводять центр волонтерів. Створювати та редагувати заходи зможе лише керівник центру волонтерів, він же адміністратор.

Для роботи необхідно здійснити авторизацію в додатку, якщо авторизація відбувається вперше, є можливість зареєструватися для подальшого використання. Модуль авторизації є вбудованим у додаток для центру волонтерів.

Далі користувач потрапляє в головне меню мобільного додатку, в якому він може бачити список вже створених заходів, які були завершені або перебувають у стадії планування. Тут авторизований користувач може переглядати запропоновані заходи, а також взяти участь у тих, які не були сформовані до кінця. А адміністратор може редагувати старі та створювати нові заходи. На рисунках 3.3 та 3.4 показані варіанти відображення головного меню, без заходів та з активними заходами відповідно.

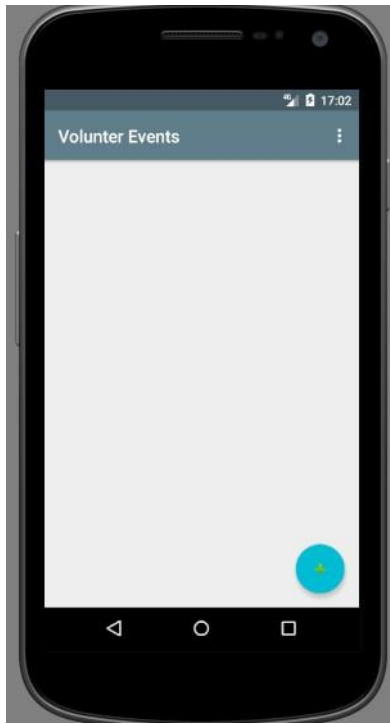


Рисунок 3.3 – Головне меню без заходів

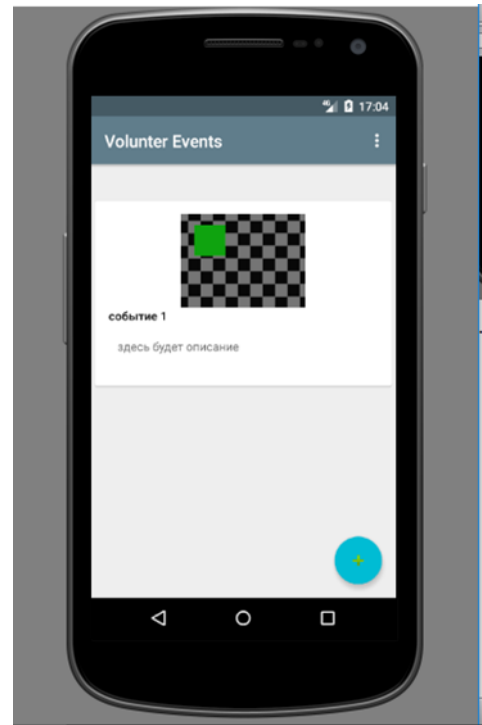


Рисунок 3.4 – Головне меню з вибором заходів

Далі, користувач-волонтер має змогу залишити головне меню та перейти у форму будь-якого заходу, та переглянувши його, подати заявку на участь, що ми можемо побачити на рисунку 3.5.

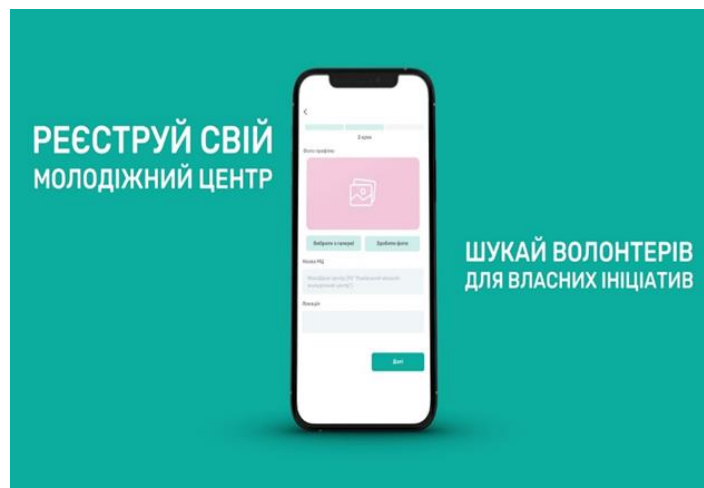


Рисунок 3.5 – Заявка на участь в заході

Адміністратор-керівник, перемістившись у форму заходу, має можливість редагувати його параметри, що надано на рисунку 3.6.

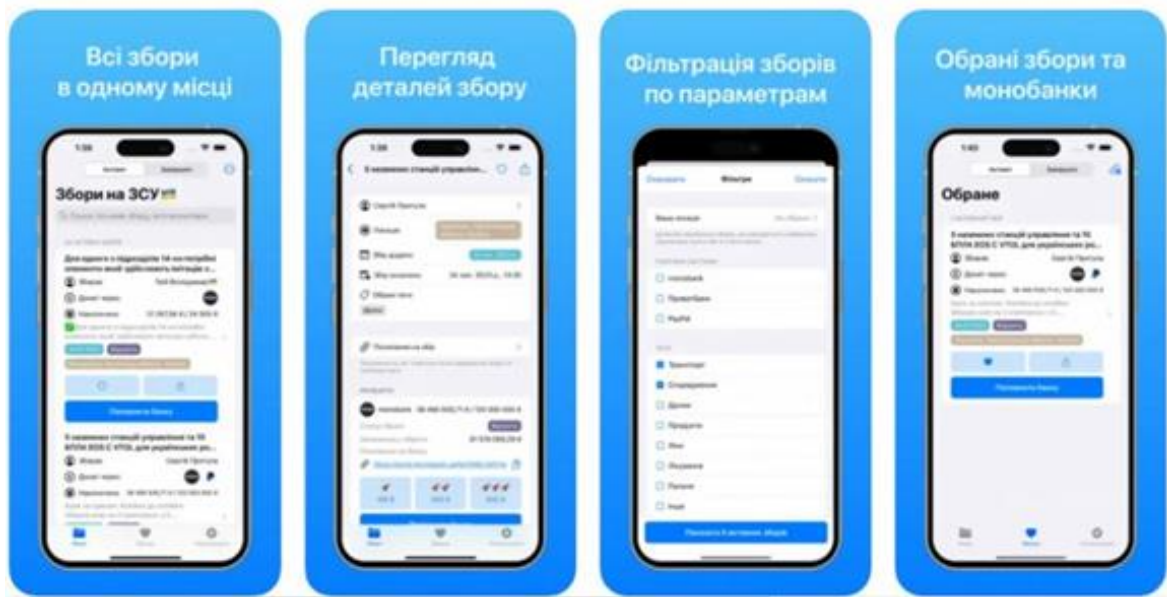


Рисунок 3.6 – Форма створення та редагування заходу

Також у функціонал мобільного додатку для центру волонтерів включено можливість перегляду та додавання фотоматеріалів, до цієї функції мають доступ як користувач-волонтер, так і адміністратор-керівник. Форму перегляду та редагування альбому надано на рисунку 3.7

Таким чином, переглянувши розділи створення форми заходу, розділ редагування форми заходу, форму заявки на реєстрацію у заході та розділ фотографії, можна сказати, що додаток функціонує та задовольняє заданим вимогам.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було реалізовано застосунок, для якого реалізовано функції головного меню, форму створення заходу, форму редагування, форму перегляду та редагування фотоматеріалів, можливість реєстрації волонтера-користувача.

Було проаналізовано діяльність центру волонтерів, спроектовано контекстну діаграму з подальшою її декомпозицією основних бізнес-процесів діяльності центру волонтерів. Було побудовано діаграму потоків даних, а також здійснено докладний розгляд роботи центру волонтерів, на основі аналізу, були сформовані основні вимоги до МП, що розробляється. Було виділено основні функції програми. А також було здійснено порівняльний аналіз таких додатків.

Було описано основні принципи роботи мобільним додатком. Здійснено тестування мобільного додатка на пристроях з мобільною операційною системою Android, Sony Xperia P та Galaxy.

Розроблений мобільний додаток дозволить здійснювати створення, моніторинг та редагування заходів для центру волонтерів, а також завантаження та перегляд фотоматеріалів із проведених заходів, буде можливе коментування.

Реалізована мобільна програма буде мати широкий функціонал для управління заходами і стане хорошим помічником в організації різних зустрічей і подій, тим самим забезпечить скорочення тимчасових витрат. Розвиток програми передбачає збільшення функціональних можливостей, надалі може стати початком створення цілої соціальної мережі для людей даного напрямку. До збільшення функціональних можливостей можна віднести можливість коментування заходів, прив'язка до GPS-систем, для відстеження учасників заходів і легшого збору, можливість залишати відгуки для вже проведених заходів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Волонтерська діяльність в Україні: соціально-правове дослідження: монографія / Павло Горінов, Ростислав Драпушко. Київ: Державний інститут сімейної та молодіжної політики, 2022. 240 с.
2. Леонова І. Волонтерський рух в Україні. Київ : 2019. 5с.
3. Липитчук О. Тенденції розвитку волонтерського руху в Україні. Одеса : 2022. 181 с.
4. Як створити успішний волонтерський проєкт. Посібник від БФ "Добро починається з тебе". Київ, 2020. 36 с.
5. Що таке Flutter та які його особливості. URL: <https://highload.today/uk/shho-take-flutter-ta-yaki-jogoosoblivosti>.
6. Apache Cassandra. URL: <https://blog.iteducenter.ua/articles/apache-cassandra>.
7. Що таке Firebase. URL: <https://qagroup.com.ua/publications/what-is-firebase>.
8. Чемерис Г. Ю. UX/UI дизайн : навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності "Дизайн" освітньо-професійної програми "Графічний дизайн". Запоріжжя : ЗНУ, 2021. 290 с
9. Куленко М. Я. Основи графічного дизайну : підручник : для студентів вищ. навч. закл. / Михайло Якович Куленко ; Київ. нац. ун-т будівництва та архітектури ; за ред. Є. А. Антонович. – Вид. 3-тє, перероб. та допов. – Київ : Кондор, 2018. 543 с.
10. Степаненко Л. Дизайн – синергія мистецтва та науки. Київ: 2021. 76 с.
11. Ярцев В.П. Організація баз даних та знань: навчальний посібник.- К. ДУТ 2018. 214с.
12. Резніченко В. Моделі та засоби систем баз даних і знань. Київ : 2021. 57 с.
13. Ситник Н. Організація баз даних. Київ : 2022 .38 с.
14. Руденко В. Бази даних. Харків: 2021. 54 с.

15. Створення додатку на Flutter. URL : <https://dou.ua/lenta/articles/flutter-first-steps>.
16. Розробка мобільних додатків на iOS та Android. URL: <https://dou.ua/lenta/articles/mobile-dev-guide/>
17. Створення Progressive Web Applications (PWA). URL: <https://dou.ua/lenta/articles/progressive-web-applications/>
18. Дворецький М. Л., Нездолій Ю. О., Дворецька С. В., Кандиба І. О. Розробка мобільних застосунків для OS Android : навч. посіб. – Миколаїв: Вид-во ЧНУ ім. Петра Могили, 2021. – 140 с.

ДОДАТОК А

```
import * as React from 'react';
import { NavigationContainer } from '@react-navigation/native';
import { createNativeStackNavigator } from '@react-navigation/native-stack';
import Login from './components/Login';
import Registration from './components/Registration';
import Home from './components/Home';
import EventDetails from './components/EventDetails';

const Stack = createNativeStackNavigator();

function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Home">
        <Stack.Screen name="Home" component={Home} options={{ title: 'Волонтерські Заходи' }} />
        <Stack.Screen name="Login" component={Login} options={{ title: 'Вхід' }} />
        <Stack.Screen name="Registration" component={Registration} options={{ title: 'Реєстрація' }} />
        <Stack.Screen name="EventDetails" component={EventDetails} options={{ title: 'Деталі Заходу' }} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}

export default App;
```

```
import React, { useState } from 'react';
import { View, TextInput, Button, Alert } from 'react-native';
import axios from 'axios';
import { useNavigation } from '@react-navigation/native';

const Login = () => {
  const navigation = useNavigation();
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');

  const validateInputs = () => {
    if (email.length === 0 || password.length === 0) {
      Alert.alert("Помилка", "Обидва поля повинні бути заповнені!");
      return false;
    }
    // Тут можна додати додаткові правила валідації
    return true;
  };

  const handleLogin = () => {
    if (validateInputs()) {
      axios.post('https://volunteer.com/login', {
        email,
        password
      })
        .then(response => {
          Alert.alert("Успішний вхід", "Ви увійшли до системи!");
          navigation.navigate('Home'); // Переходите до домашнього екрану або іншого екрану
        })
        .catch(error => {
          Alert.alert("Помилка входу", error.message);
        });
    }
  };

  return (
    <View>
      <TextInput placeholder="Email" onChangeText={setEmail} value={email} />
      <TextInput placeholder="Пароль" onChangeText={setPassword} value={password} secureTextEntry />
      <Button title="Вхід" onPress={handleLogin} />
    </View>
  );
};

export default Login;
```

```

import React, { useState } from 'react';
import { View, TextInput, Button, Alert } from 'react-native';
import axios from 'axios';
import { useNavigation } from '@react-navigation/native';

const Registration = () => {
  const navigation = useNavigation();
  const [email, setEmail] = useState('');
  const [password, setPassword] = useState('');
  const [username, setUsername] = useState('');

  const validateInputs = () => {
    if (username.length === 0 || email.length === 0 || password.length === 0) {
      Alert.alert("Помилка", "Всі поля повинні бути заповнені!");
      return false;
    }
    // Тут можна додати додаткові правила валідації
    return true;
  };

  const handleRegister = () => {
    if (validateInputs()) {
      axios.post('https://volunteer.com/register', {
        username,
        email,
        password
      })
      .then(response => {
        Alert.alert("Реєстрація успішна", "Ласкаво просимо!");
        navigation.navigate('Home'); // Переходите до домашнього екрану або іншого екрану
      })
      .catch(error => {
        Alert.alert("Помилка реєстрації", error.message);
      });
    }
  };

  return (
    <View>
      <TextInput placeholder="Ім'я користувача" onChangeText={setUsername} value={username} />
      <TextInput placeholder="Email" onChangeText={setEmail} value={email} />
      <TextInput placeholder="Пароль" onChangeText={setPassword} value={password} secureTextEntry />
      <Button title="Реєстрація" onPress={handleRegister} />
    </View>
  );
};

export default Registration;

```

```

import React, { useState, useEffect } from 'react';
import { View, Text, FlatList, TouchableOpacity, Image, StyleSheet } from 'react-native';
import axios from 'axios';
import { useNavigation } from '@react-navigation/native';

const Home = () => {
  const navigation = useNavigation();
  const [events, setEvents] = useState([]);

  useEffect(() => {
    axios.get('https://volunteer.com/events')
      .then(response => {
        setEvents(response.data.events);
      })
      .catch(error => {
        console.log(error);
      });
  }, []);

  const renderItem = ({ item }) => (
    <TouchableOpacity style={styles.card} onPress={() => navigation.navigate('EventDetails', { eventId: item.id })}>
      <Image source={{ uri: item.imageUrl }} style={styles.image} />
      <View style={styles.info}>
        <Text style={styles.title}>{item.name}</Text>
        <Text style={styles.description}>{item.description}</Text>
      </View>
    </TouchableOpacity>
  );

  return (
    <View style={styles.container}>
      <FlatList
        data={events}
        renderItem={renderItem}
        keyExtractor={item => item.id.toString()}
      />
    </View>
  );
};

const styles = StyleSheet.create({
  // ...
});

export default Home;

```

```

import React, { useState, useEffect } from 'react';
import { View, Text, Image, Button, Alert, StyleSheet } from 'react-native';
import axios from 'axios';
import { useNavigation, useRoute } from '@react-navigation/native';

const EventDetails = () => {
  const navigation = useNavigation();
  const route = useRoute();
  const { eventId } = route.params;
  const [eventDetails, setEventDetails] = useState(null);

  useEffect(() => {
    axios.get(`https://volunteer.com/events/${eventId}`)
      .then(response => {
        setEventDetails(response.data);
      })
      .catch(error => {
        console.log(error);
      });
  }, [eventId]);

  const handleRegister = () => {
    // Перевірте тут умови реєстрації, якщо потрібно
    axios.post(`https://volunteer.com/events/register`, { eventId })
      .then(response => {
        Alert.alert("Реєстрація на захід", "Ви успішно зареєструвалися на захід!");
        navigation.goBack(); // Поверніться до попереднього екрану або перенаправте куди-небудь інше
      })
      .catch(error => {
        Alert.alert("Помилка реєстрації", error.message);
      });
  };

  if (!eventDetails) return <View><Text>Завантаження...</Text></View>;

  return (
    <View style={styles.container}>
      <Image source={{ uri: eventDetails.imageUrl }} style={styles.image} />
      <Text style={styles.title}>{eventDetails.name}</Text>
      <Text style={styles.description}>{eventDetails.description}</Text>
      <Button title="Зареєструватися на захід" onPress={handleRegister} />
    </View>
  );
};

const styles = StyleSheet.create({
});

export default EventDetails;

```