

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерних наук,
управління та адміністрування
Кафедра інформаційних технологій

Кваліфікаційна робота бакалавра

на тему: Розробка мобільного за стосунку для редагування та систематизації
зображень за їх контентом

Виконав студент групи К-19
спеціальності 122 «Комп'ютерні науки»
Бабаєв Мекан

Керівник к.геогр.н., доцент
Кузніченко Світлана Дмитрівна

Консультант _____

Рецензент к.техн.н., доцент
Гнатовська Ганна Арнольдівна

Одеса 2023

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ	6
ВСТУП	7
1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ МОДИФІКАЦІЇ	9
ЗОБРАЖЕНЬ ТА ТЕХНОЛОГІЙ ЇХ РОЗРОБЛЕННЯ.....	9
1.1 Аналіз алгоритму масштабування зображень.....	9
1.2 Аналіз існуючих мобільних за стосунків для масштабування та модифікації зображень	11
1.3 Вибір інструментів для проектування	14
1.4 Вибір архітектурного шаблону.....	17
1.5 Вибір мови програмування	21
1.6 Вибір середовища розробки.....	24
1.7 Вибір фреймворків та бібліотек	27
1.8 Вибір системи керування базами даних	28
2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ	32
2.1 Мета та задачі ІС	32
2.2 Типи користувачів.....	33
2.3 Функціональні вимоги.....	33
2.4 Нефункціональні вимоги.....	37
2.5 Ідентифікація архетипу ІС	39
2.6 Користувацький інтерфейс (UI View).....	39
2.7 Уявлення розгортання ІС (Deployment View).....	43
2.8 Уявлення даних ІС (DataView)	43
3. РЕАЛІЗАЦІЯ СИСТЕМИ.....	45
3.1 Уявлення про структуру проекту ІС	45
3.2 Уявлення про класи ІС	45
3.3 Інфраструктурне уявлення ІС (Infrastructure View).....	48
3.4 Документація ІС	48
ВИСНОВКИ.....	52

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ 53

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ОС – операційна система

ПЗ – програмне забезпечення

ADT – Android Development Tools – Інструменти розробки в ОС Android

API – Application Programming Interface – програмний інтерфейс
програми

AR – Augmented Reality – доповнена реальність

IDE – Integrated Development Environment – інтегроване середовище
розробки

SDK – Software Development Kit – набір засобів розробки

ВСТУП

Останнім часом спостерігається тенденція накопичення великої кількості графічної інформації, тому необхідність її ефективної систематизації (каталогізація) для зручного зберігання, швидкого доступу, додавання нових файлів у існуючі каталоги, обробки та редагування робить актуальною задачу розробки відповідних програмних засобів.

Зміна розміру зображення є стандартним інструментом у багатьох застосунках для обробки зображень. Прості методи включають масштабування та обрізку, які призводять до втрати важливих особливостей зображення або спотворень. Ефективніша зміна розміру зображення може бути досягнута завдяки врахуванню контенту зображення, а не лише геометричних обмежень. Одним з таких підходів є алгоритм Seam carving, що дозволяє змінювати розмір зображення за рахунок пошуку так званих сімів найменш важливих пікселів та подальшого їх видалення чи дублювання. Тобто Seam carving дозволяє видалити пікселі з нецікавих частин зображення, зберігаючи при цьому важливий вміст. Seam carving може змінювати розмір зображення, вирізаючи чи вставляючи пікселі в різні частини зображення. Цей алгоритм використовує функцію енергії, яка визначає, наскільки важливі пікселі. Окрім зміни розміру цей підхід застосовується і для видалення зайвих об'єктів, визначених користувачем.

Зміна розміру зображень може бути обумовлена різними факторами, такими як перегляд на різних пристроях з різними роздільними здатностями, розміщення у Інтернеті, використання у дизайнерських рішеннях тощо. Зручним є забезпечення подібної функціональності шляхом використання мобільних технологій, так як практично у кожної сучасної людини є смартфон, який в будь-який момент часу може бути використаний користувачем застосунку. Таким чином, розробка програмного забезпечення для редагування зображень з врахуванням їх контенту у вигляді мобільного застосунку є актуальним завданням.

Метою кваліфікаційної роботи є розробка мобільного застосунка для ОС Android, що реалізує алгоритм модифікації зображень з урахуванням їх змістовного наповнення.

Для досягнення поставленої мети необхідно вирішити наступні завдання:

- провести порівняльний аналіз існуючих мобільних застосунків аналогів та виявити функціонал, який доцільно запозичити для власної розробки;
- виконати порівняльний аналіз існуючих методів модифікації та масштабування зображень та обґрунтувати вибір алгоритмів, що будуть використовуватися для вирішення завдання;
- обґрунтувати вибір програмних засобів розробки та технологій;
- виконати проектування мобільного застосунку;
- розробити алгоритми та програмні реалізації методів модифікації зображень;
- виконати реалізацію клієнтської та серверної частин мобільного застосунку;
- виконати тестування розробленого програмного забезпечення.

Структура кваліфікаційної роботи складається вступу, трьох розділів, висновків, переліку посилань на 15 найменувань, додатків. Повний обсяг роботи становить 54 сторінок, містить 24 рисунки і 3 таблиці.

1 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ДЛЯ МОДИФІКАЦІЇ ЗОБРАЖЕНЬ ТА ТЕХНОЛОГІЙ ЇХ РОЗРОБЛЕННЯ

1.1 Аналіз алгоритму масштабування зображень

Масштабуванням зображення – це збільшення або зменшення розміру зображення зі збереженням його пропорцій. Масштабування може здійснюватися кількома методами в залежності від типу графіки. З векторною графікою проблем з масштабуванням не відбувається. Однак, при роботі з растровим зображенням потрібно використовувати спеціальні алгоритми.

Стандартне масштабування зображення не є достатнім, оскільки воно погіршує зміст зображення, і зазвичай його можна застосовувати рівномірно. Ефективне стиснення зображень повинне не тільки використовувати геометричні обмеження, але й враховувати зміст зображення. Тому в кваліфікаційній роботі пропонується реалізувати алгоритм масштабування зображень Seam carving [1, 5], який підтримує зміну розміру зображень для вмісту як для зменшення, так і для розширення. Сім (seam) – оптимальний з'єднаний шлях пікселів на одному зображенні зверху вниз або зліва направо, де оптимальність визначається енергетичною функцією зображення, яка розраховує важливість пікселів. Тобто сім – це сполучений шлях з пікселями з низькою енергією. Завдяки багаторазовому вирізанню або вставці сімів у одному напрямку можна змінити співвідношення сторін зображення (рис.1.1). Застосовуючи ці оператори в обох напрямках, можна перетворити зображення до нового розміру. Вибір і порядок сімів захищають зміст зображення, як це визначено функцією енергії. При зменшенні зображення, вибір сімів гарантує, що при збереженні структури зображення буде вилучено більше пікселів з низькою енергією (рис.1.2). При збільшенні зображення вставки сімів забезпечує баланс між вихідним вмістом зображення та штучно вставленими пікселями. Фактично, ці оператори створюють змістовне змінення розмірів зображень.



Рисунок 1.1 – Зображення з одним горизонтальним і одним вертикальним сі-
мом



а



б

Рисунок 1.2 – Результати масштабування: а – масштабування алгоритмом
Seam carving; б – стандартне масштабування

Алгоритм Seam carving також може бути використаний для поліпшення вмісту зображення та видалення об'єкта. Зберігаючи порядок сімів у зображенні, можна створювати багаторозмірні зображення, здатні постійно змінюватися в режимі реального часу, щоб відповідати заданому розміру [2].

1.2 Аналіз існуючих мобільних за стосунків для масштабування та модифікації зображень

В даний час існує досить велика кількість мобільних Android застосунків для масштабування та модифікації зображень. Більшість має простий інтерфейс і дуже схожий функціонал. Далі розглянемо деякі з них.

Photo Resizer [3] – це безкоштовний, швидкий і простий у використанні інструмент для оптимізації та масштабування зображень для широкого спектра застосувань (рис.1.3). Застосунок дозволяє користувачу зменшити розмір зображення у форматі JPEG чи PNG шляхом обрізки вказаних значень пікселів, або процентів для розміщення в Facebook, Twitter, Pinterest, Reddit, Tumblr або іншій соціальній мережі.



Рисунок 1.3 – Інтерфейс мобільного застосунку Photo Resizer

До переваг Photo Resizer слід віднести:

- наявність пакетної зміни розміру (зміна розміру кількох фотографій);
- простий у використанні інтерфейс;
- функція порівняння зображень поруч (оригінал і модифікація);
- визначені найбільш часто використовувані розміри фотографій;
- можливість визначити власний розмір зображення;
- можливість поділитися зображенням в соціальних мережах (Instagram, Facebook, Flickr, Google+, ВКонтакте, KakaoTalk та ін.).

До недоліків програми можна віднести

- відсутність функції модифікування зображення, наприклад, видалення зайвих об'єктів;
- відсутність можливості масштабування лише по горизонталі чи по вертикалі (зменшення розміру фотографії зберігає оригінальне співвідношення сторін);
- при масштабуванні не враховується вміст (контент) зображення.

Програма Lit Photo [3] дозволяє швидко зменшити розмір зображення або його роздільну здатність. Крім того, Lit Photo – це дуже просте у використанні застосунок для обрізки і повороту фотографій (рис.1.4).

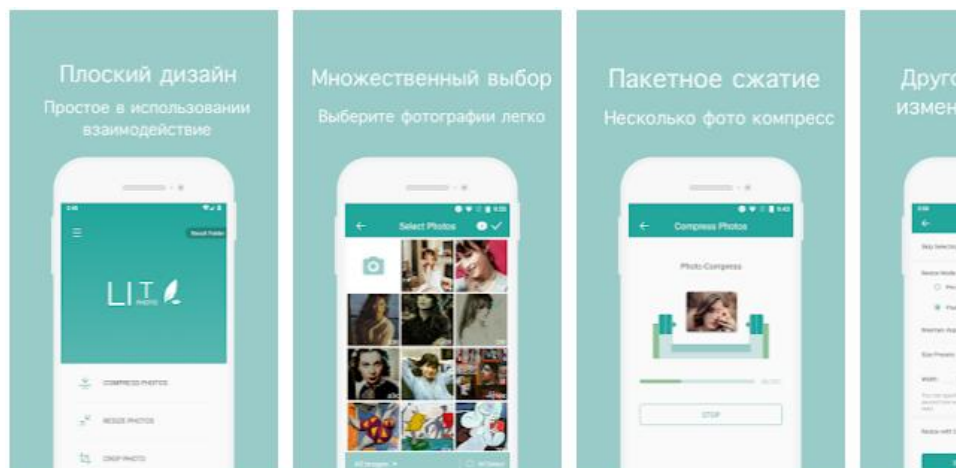


Рисунок 1.4 – Інтерфейс мобільного застосунку Lit Photo

Lit Photo використовує розумні методи стиснення з втратами, щоб зменшити розмір файлу із зображенням. Через вибіркового зменшення кількості кольорів в зображенні потрібно менше байтів для зберігання даних. Ефект майже невидимий, але він дуже сильно впливає на розмір файлу. Користувачу не потрібно вручну зберігати зображення зі зміненим розміром, тому що вони автоматично зберігаються в окремій папці з назвою «LitPhoto» і доступні через вбудований додаток «Галерея».

До переваг Lit Photo можна віднести:

- пакетне стиснення і зміна розміру (кілька фотографій в процесі)
- стиснення фотографій до нестандартного розміру
- можливість повороту фотографій на 90 ° за годинниковою стрілкою для обрізки
- можливість поділитися зображенням через соціальні мережі (Instagram, Facebook, Line, Google+, WeChat та ін.) або по електронній пошті.

До недоліків програми можна віднести відсутність функції модифікування зображення та не можливість врахування вмісту (контенту) зображення.

Наступна програма Codenia [3] дозволяє швидко і легко змінювати розмір зображень, використовуючи одну з наступних чотирьох одиниць виміру: пікселі, міліметри, сантиметри, дюйми (рис.1.5). Застосунок надає можливість зберегти, відправляти по електронній пошті, друкувати або ділитися кінцевим зображенням. Обрізка зображення можлива за допомогою мультисенсорних жестів. Використовувані розміри зображень зберігаються і можуть бути швидко обрані зі списку. Зображення можна повертати на 90 ° вправо і вліво.

До переваг програми Codenia можна віднести доданий редактор фотографій, що має можливості роботи з фільтрами, текстом, наклейками та ін. Якість JPG та шлях збереження вихідних зображень може бути змінені в налаштуваннях.

Як і в попередніх програмах недоліком програми є відсутність функції модифікування зображення з врахуванням вмісту (контенту) зображення.

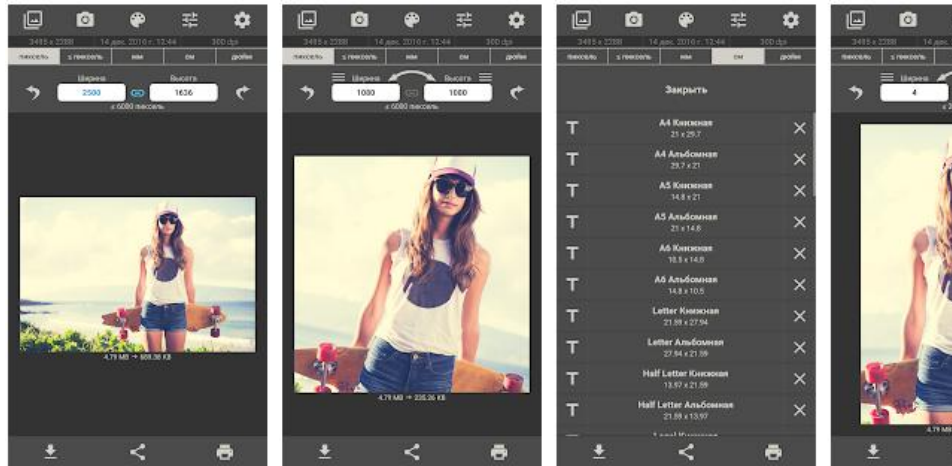


Рисунок 1.5 – Інтерфейс мобільного застосунку Codenia

Результати порівняльного аналізу функціоналу мобільних застосунків для масштабування зображень наведені в табл. 1.1. Програма Seam Resize – це власний застосунок, що буде реалізовано в роботі на основі алгоритму Seam carving.

1.3 Вибір інструментів для проектування

Перед створенням програмного коду мобільного застосунка необхідно виконати проектування майбутнього програмного забезпечення. Це завдання в свою чергу потребує обґрунтування вибору інструментів для проектування.

Розглянемо найбільш популярні засоби для проектування програмного забезпечення. До них відносяться:

- BPMN Модель Бізнес-Процесів;
- побудова блок-схем;
- створення ER-діаграм;
- UML-діаграми.

Модель BPMN є стандартом для моделювання бізнес-процесів, що надає графічну нотацію для визначення складної семантики бізнес-процесу у вигляді діаграми BPD. Така діаграма ґрунтується на представленні бізнес-процесу у вигляді блок-схеми, що семантично схожа на діаграму діяльності [4].

Таблиця 1.1 – Порівняльний аналіз функціоналу програм аналогів

Характеристики	Photo Resizer	Lit Photo	Codenia	Seam Resize
Простий інтерфейс	+	+	+	+
Вибір розміру зображення	+	+	+	+
Функція порівняння зображень поруч	+	–	–	+
Наявність функції обрізки зображення	+	+	+	+
Поворот зображення	–	+	+	–
Пакетне стиснення	+	+	+	–
Масштабування по горизонталі/вертикалі	–	+	+	+
Аналіз вмісту зображення	–	–	–	+
Наявність функції модифікування зображення	–	–	–	+

Блок-схема (англ. flowchart) дозволяє представити алгоритм розв'язування або аналізу задачі за допомогою геометричних елементів (блоків), які позначають операції, потік, дані тощо [5].

Схема «сутність-зв'язок» (також ERD або ER-діаграма) – це різновид блок-схеми, де показано, як різні «сутності» пов'язані між собою всередині системи [6]. ER-діаграми найчастіше застосовуються для проектування і на-

лагодження реляційних баз даних. У ER-моделях і моделях даних зазвичай виділяють до трьох рівнів деталізації: концептуальна, логічна та фізична модель даних. ER-діаграма демонструє зв'язки і відносини між елементами, тому вона відображають тільки реляційну структуру.

UML (англ. Unified Modeling Language) – уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування [7]. Це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, яка називається UML-моделлю. Діаграми дають можливість представити систему у такому вигляді, щоб її можна було легко перевести в програмний код. Це дозволяє збільшити ефективність реалізації програмних систем кілька разів і помітно поліпшити якість кінцевого продукту. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Серед найбільш відомих UML діаграм слід вказати наступні [8, 9]:

- структурні діаграми: класів, компонент, розсортування, об'єктів, пакетів;
- діаграми поведінки: діяльності, станів, прецедентів;
- діаграми взаємодії: послідовності, синхронізації.

Найбільш часто при використанні моделі BPMN застосовуються такі пакети: Sybase Power Designer; Eclipse; Visio + BPMN; AcuaLogic BPMN. Для створення блок-схем, а також побудови ER-діаграм використовується пакет Microsoft Visio, що можна назвати найбільш універсальним інструментом проектувальника програмного забезпечення. Для розробки ER-діаграм також застосовуються ERWin і згаданий вище пакет Sybase Power Designer.

Для побудови UML-діаграм може бути використано програмний пакет StarUML [10]. Це проект з відкритим кодом для розробки швидких, гнучких, функціональних платформ UML/MDA для 32-розрядних систем Windows. Платформа StartUML є аналогом таких комерційних проектів, як Rational Rose, Together та інших.

Таким чином, після виконаного аналізу при проектуванні програмного забезпечення будуть використані наступні засоби проектування: нотація UML для моделювання структури, поведінки та взаємодії програмної системи та блок-схема для опису алгоритмів модифікації зображень.

1.4 Вибір архітектурного шаблону

Архітектурний шаблон – це узагальнене рішення поширеного завдання в архітектурі ПЗ, що часто використовується в заданому контексті. Передбачається, що мобільний за стосунок, що розроблюється буде мати клієнт-серверну архітектуру.

Згідно парадигмі клієнт-серверної архітектури один або кілька клієнтів і один або кілька серверів спільно з базовою операційною системою і середовищем взаємодії утворюють єдину систему, що забезпечує розподілені обчислення, аналіз і представлення даних.

Функціональність програми повинна бути розподілена між клієнтської і серверної частиною (рис.1.6).

У класичному випадку дана схема функціонує наступним чином [11, 12]:

- клієнт формує і посилає запит на сервер;
- сервер робить необхідні маніпуляції з даними, формує результат і передає його клієнту;
- клієнт отримує результат, відображає його на пристрої виведення і чекає подальших дій користувача.

Цикл повторюється, поки користувач не закінчить роботу з сервером.

Недоліком архітектури «клієнт-сервер» є те, що сервер може бути вузьким місцем щодо продуктивності, а також єдиною точкою відмови. Змінювати прийняте рішення про розміщення функціональних можливостей (на клієнті або на сервері) після створення системи – це зазвичай складно і дорого.

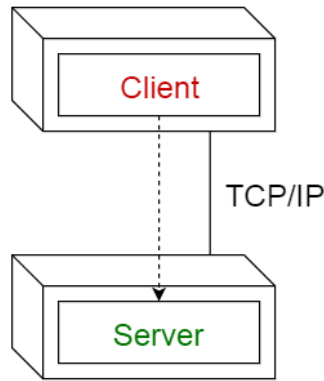


Рисунок 1.6– Клієнт-серверна архітектура

Не існує формального переліку існуючих архітектурних патернів. Але на сьогоднішній день можна виділити основні (часто використовувані) підходи:

- багаторівнева архітектура;
- архітектура, керована подіями (Event-Driven Architecture);
- сервіс-орієнтована (мікросервісна) архітектура.

Проведемо аналіз переваг та недоліків найпопулярніших архітектурних шаблонів. На основі цього аналізу обґрунтуємо вибір архітектурного шаблону для мобільного застосунку.

Багаторівнева (багатоланкова) архітектура є однією з найвідоміших архітектур, в якій кожен шар виконує певну функцію. Залежно від потреб можна реалізувати будь-яку кількість рівнів (рис.1.7). Перевагами застосування такої архітектури є простота розробки (в основному через те, що цей вид архітектури всім знаком) і простота тестування. Серед недоліків можна виділити можливі складнощі з продуктивністю і масштабуванням – всьому виною необхідність опрацювання запитів і даних по всіх рівнях (знову ж таки, в тому випадку, якщо всі рівні є закритими) [12].

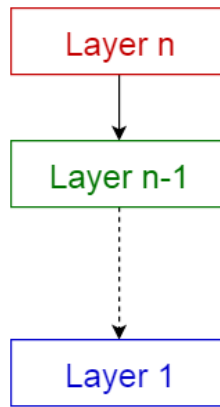


Рисунок 1.7 – Багаторівневий шаблон

Найбільш відома структура це триланкова архітектура (рис.1.8).

Клієнт — це інтерфейсний компонент, який представляє перший рівень, власне застосунок для кінцевого користувача.

Сервер застосунків розташовується на другому рівні і тут зосереджена більша частина бізнес-логіки.

Сервер бази даних забезпечує зберігання даних і виноситься до третього рівня. Зазвичай це стандартна реляційна або об'єктно-орієнтована СКБД.

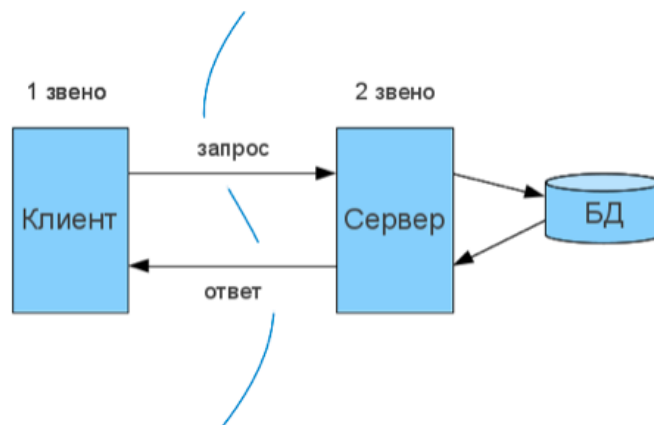


Рисунок 3.7 – Триланкова архітектура

Шаблон «шина подій» в основному, взаємодіє з подіями і складається з 4 основних компонентів (рис.1.8): джерело події, прослуховувач події, канал і шина подій. Джерела розміщують повідомлення для певних каналів на шині

подій. Прослуховувачі підписуються на певні канали. Прослуховувачі отримують повідомлення про появу повідомлень, розміщених на каналах з їх підписки. Шаблон часто використовується для розробки на Android, наприклад, при створенні сервісу повідомлень. Недоліком архітектури керованої подіями є низька продуктивність і відновлення після помилок [12].

Шаблон на основі мікросервісів дозволяє створювати застосунки у вигляді наборів сервісів: кожен сервіс розгортається і масштабується незалежно і має власну «границю», яка обслуговується за допомогою API (рис.1.9). Архітектуру на основі мікросервісів використовують для розгортання серверних корпоративних застосунків, що підтримують різні браузері і нативні мобільні клієнти [12].

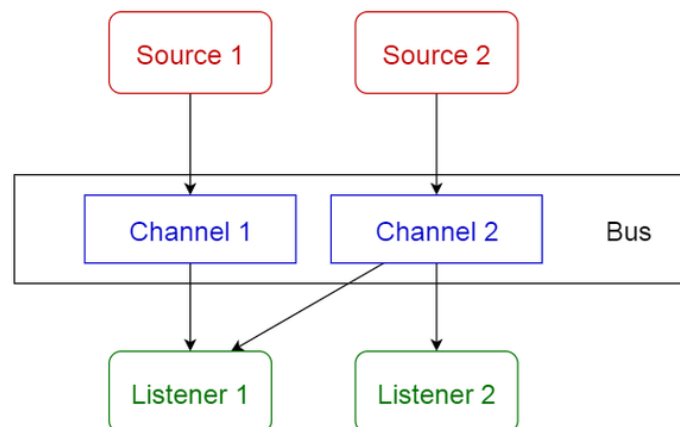


Рисунок 1.8 – Шаблон «шина подій»

Недоліки: системи повинні бути спроектовані так, щоб витримувати збої в роботі сервісів, а це вимагає більш ретельного моніторингу систем. Додаткові витрати ресурсів на організацію роботи сервісів і взаємодії подій. Крім того, буде потрібно більше пам'яті.



Рисунок 1.9 – Архітектура на основі мікросервісів

Таким чином, на підставі проведеного аналізу для реалізації мобільного застосунку була обрана триланкова клієнт-серверна архітектура: клієнт – сервер – сервер БД. Клієнтом є android-застосунок. Це одна з розповсюджених нескладних архітектур, яка найбільш підходить для затребуваного функціоналу застосунку, що розроблюється.

1.5 Вибір мови програмування

Стисло розглянемо популярні мови програмування та обґрунтуємо вибір засобу розробки для мобільного застосунку за набором критеріїв, що наведені в табл.1.2.

Swift – це нова мова програмування, яка містить в синтаксисі краще від мови Сі. Створювалася та розроблялася виключно компанією Apple для розробки програм під iOS та macOS [13].

Переваги Swift: наявність власної стандартної бібліотеки та шаблонів для розробки; висока швидкість роботи та компіляції кодів; можливість безкоштовного використання.

Недоліки Swift: є новим засобом розробки програм, тому існує можливість появи задач, що важко вирішуються в процесі роботи; розроблена виключно для платформи Apple (iOS і macOS).

PHP – мова програмування, що розроблена для генерування і роботи з HTML-сторінками і базами даних на веб-сервері. Є одним з лідерів серед мов, які використовуються для створення динамічних веб-сайтів [14].

Переваги PHP: розповсюджується під ліцензією (PHP license), є вільним ПЗ; підтримка підключення БД; наявність власних бібліотек та розширень; наявність великого вибору навчальних матеріалів; сумісність роботи з великою кількістю платформ та операційних систем: є самою популярною мовою програмування для створення веб-застосунків.

Недоліки PHP: орієнтована в основному на веб-розробку; часті можливі протиріччя в коді (код містить в собі декілька мов, в чому і полягає складність); можливі проблеми з безпекою систем, що розроблюються. Мова спрямована на створення як веб-розробок, так і для написання різних програмних модулів та систем [15].

Python – високорівнева мова програмування загального призначення, орієнтована на підвищення продуктивності розробника та чисельність коду. Синтаксис Python не відрізняється громіздкістю та складністю читання. В той же час стандартна бібліотека включає великий обсяг корисних функцій [16].

Переваги Python: наявність і доступність навчальної інформації; спрямованість на створення як веб-розробок, так і різного роду та складності інформаційних систем; наявність стандартної бібліотеки з необхідним функціоналом; можливість розробок для різних цільових платформ.

Недоліки Python: у зв'язку зі складністю функціонала мови, її високим рівнем, швидкість проектів, що розроблюються, може виявитися невисокою відносно інших засобів програмування; наявність платної ліцензії Python Software Foundation License, можливість придбання якої може бути не у кожного бажуючого користувача.

Мова програмування C++ – типізована мова програмування загального призначення, що компілюється статично. Підтримує різні парадигми програ-

мування, але в порівнянні з Сі, найбільша увага приділена підтримці об'єктно-орієнтованого та узагальненого програмування.

Переваги С++: підтримка різних стилів програмування; розробка програм та інформаційних систем для різного виду платформ; висока продуктивність, що пояснюється створенням складних програмних засобів; наявність власної бібліотеки та шаблонів; можливість відкритого доступу до програми для її використання; безліч відкритих джерел для вивчення мови.

Недоліки С++: складність вивчення; у зв'язку зі складністю мови можливе зменшення швидкості роботи компілятора і програмних продуктів.

Kotlin – це мова програмування з відкритим вихідним кодом. Вона відноситься до нових статично типізованих мов, що працює поверх JVM і розробляється компанією JetBrains [17]. Kotlin використовується для створення Android-застосунків такими лідерами бізнесу як Uber, Atlassian, Pinterest, Evernote. Kotlin гарний вибір для розробки застосунків на Android. Це можна побачити навіть з того факту, що Android Studio поставляється з вбудованою підтримкою Kotlin, як і з підтримкою Java.

Переваги Kotlin: відкритий вихідний код; інтуїтивно зрозумілий лаконічний синтаксис, що значно підвищує продуктивність і швидкість написання коду; гарна сумісність з Java; висока безпека за рахунок зниження кількості збоїв; простота вивчення.

Недоліки Kotlin: часто виникають проблеми з компіляцією кода; мова має невелику спільноту і мало ресурсів для вивчення.

С# – мова програмування, яка поєднує об'єктно-орієнтовані і контекстно-орієнтовані концепції. Відноситься до родини мов з С-подібним синтаксисом, який близький до С++ та Java.

Переваги С#: підтримка різних парадигм програмування; наявність власної стандартної бібліотеки, яка спрощує написання коду; висока продуктивність, швидкість і безпека; безкоштовний доступ до середовища розробки програмних продуктів; безліч джерел для вивчення мови та вирішення проблем, що виникають.

Недоліки C#: орієнтованість на платформу Windows; наявність безкоштовного використання лише для маленьких компаній та студентів, а придбання ліцензій обійдеться дорого.

Java – об’єктно-орієнтована мова програмування, що поєднує в собі можливості таких мов, як C, C++, і може вирішувати складні задачі в процесі роботи [18]. Java є основою розробок на платформі Android. Може працювати на різних платформах (Windows, Unix і macOS, а також веб-браузерах).

Переваги Java: мова високого рівня з простим зрозумілим синтаксисом для вивчення; безліч ресурсів для вивчення; політика безпеки даних; має C-подібний синтаксис; крос-платформний, працює з різними платформами; можливість підключення баз даних; можливість доступу к безлічі бібліотек; широкий спектр варіантів використання мови: як для створення застосунків та інформаційних продуктів, так і для створення складних система та програм [19].

Недоліки Java: платне використання для розробок комерційного ПЗ (безкоштовні версії старі).; низька швидкість через великого функціонала і можливостей мови; складність синтаксису.

Результати порівняльного аналізу мов програмування представлені в табл.1.2.

Таким чином, з врахуванням створення програмного продукту для ОС Android, була обрана саме мова програмування Java, яка поряд з іншими перевагами: крос-платформністю, наявності бібліотек і фреймворків, має широкі можливості для створення мобільних застосунків для ОС Android.

1.6 Вибір середовища розробки

На даний момент існує кілька платформ для створення мобільних застосунків для Android:

- 1) Android Studio [20], заснована на мові Java і Android SDK;
- 2) Eclipse [21], що підтримує мову Java і Android Development Tools;

3) Xamarin [22], заснована на мові програмування C # і платформі .NET.

Таблиця 1.2 – Порівняльний аналіз сучасних мов програмування

	Swift	PHP	Python	Kotlin	C#	Java
Читабельність	+	+	+	+	+	+
Суміщення роботи на декількох цільових платформах	–	+(веб-розробки)	+	+(орієнтація на Android)	+	+(орієнтація на Android)
Гнучкість	+	+	+/-	+	+	+
Продуктивність	+	+	–	+	+	+
Доступність джерел для вивчення мови програмування	–	+	+	+/-	+	+
Можливість безкоштовного використання	+	+	–	+	+/-	+/-
Наявність власного набору можливостей	+	+	+	+	+	+
Можливість суміщення роботи з файлами та базами даних	+	+	+	+	+	+
Безпека даних	+	+/-	+	+	+	+

Android Studio – це інтегроване середовище розробки (IDE) для роботи з платформою Android, є офіційною середовищем розробки застосувань для платформи Android.

Має наступні особливості:

- розширений редактор макетів: WYSIWYG, здатність працювати з UI компонентами за допомогою Drag-and-Drop, функція попереднього перегляду макета на декількох конфігураціях екрану;
- різні види збірок і генерація кількох .apk файлів;
- рефакторинг коду;
- статичний аналізатор коду (Lint), що дозволяє знаходити проблеми продуктивності, несумісності версій і інше;
- шаблони основних макетів і компонентів Android;
- підтримка розробки додатків для Android Wear і Android TV;
- вбудована підтримка Google Cloud Platform, яка включає в себе інтеграцію з сервісами Google Cloud Messaging і App Engine.

Eclipse – вільне інтегроване середовище розробки крос-платформних застосунків. Android Development Tools (ADT) – це плагін для Eclipse IDE, що дозволяє розробляти застосунки для платформи Android. Дане середовище розробки дозволяє створювати настільки ж ефективні програми для Android, але тому що Android Studio є інструментом, спеціально створеним для мобільного розробки, то вона дозволяє простіше і швидше створювати застосунки в порівнянні з Eclipse.

Xamarin – це фреймворк для крос-платформної розробки мобільних застосунків з використанням мови C#. Xamarin володіє наступними перевагами і недоліками:

- дозволяє створювати кросплатформні мобільні застосунки (iOS, Android, Windows Phone);
- не вимагає знань стандартних засобів реалізації Android застосунків (заснований на C# і платформі .NET);
- платна ліцензія;
- неповна підтримка стандартних API платформи;
- відносно низька продуктивність.

З огляду на переваги і недоліки розглянутих інструментів, для реалізації програмної системи було обране середовище розробки Android Studio.

1.7 Вибір фреймворків та бібліотек

JavaEE-стандарт для побудови enterprise рішень і для побудови web застосунків. Основна структурна одиниця це сервлет. Даний клас є обробником http-запитів. В якості параметру він приймає класи http-запиту і http-відповіді, тобто необхідно самому формувати html, передаючи потрібні рядки потоку виведення http-відповіді. Для побудови застосунку даний підхід є сильно низькорівневим, але крім сервлету даний клас включає інші технології, які перетворюються в сервлет, наприклад JSP-сторінки. Проте, у даного засобу є велика перевага, а саме безліч класів і технологій, найпоширеніші – це Enterprise Java Bean і Dependency injection (створення зв'язку агрегації і композиції між класами). EJB – це класи для роботи з даними. Так само плюсом є підтримка патерна проектування MVC.

JavaEE все ж не є оптимальним рішенням, але існує Java Spring фреймворк [23], який базується на даному стандарті і при цьому розширює його. Він включає Spring MVC (для розробки сайтів), Spring Boot (для легкого підключення інших модулів). Spring підтримує свої Beans і теж підтримує Dependency injection, JPA Hibernate. Hibernate це ORM від Spring, що дозволяє працювати з СКБД за допомогою об'єктної моделі. Працювати можна як в реляційними базами даних, так і з NoSql, при цьому робота з відображеними об'єктами проводиться за допомогою анотацій. Так само не можна не згадати про нову мову запитів JPQL, яка працює вже з Java-об'єктами. Це робить застосунок масштабованим. Також слід згадати про можливість роботи з REST сервісами. Звичайно, у даного фреймворка є великі функціональні можливості. До мінусів цього фреймворка можна віднести підвищені вимоги до апаратної частини ЕОМ. Так само, як і JavaEE для розгортання Spring потрібно контейнер сервлетів [24]. Незважаючи на все це, Spring володіє високою

швидкодією і надійністю. В якості альтернативи Spring може бути розглянутий фреймворк Spark.

Spark – це безкоштовний фреймворк з відкритим вихідним кодом і предметно-орієнтована мова, написана на Java. За замовчуванням він працює на вбудованому веб-сервері Jetty, але його можна налаштувати для роботи на інших веб-серверах. Однак, він не підтримує шаблон модель - представлення - контролер, який використовується в Spring MVC. Замість цього Spark призначений для «швидкого створення веб-застосунків на Java з мінімальними зусиллями» [25].

До недоліків Spark можна віднести відсутність великої кількості матеріалів, доступних в Інтернеті в порівнянні, наприклад, з фреймворком Spring Boot, що знижує продуктивність і швидкість створення коду при виникненні певних проблем.

Тому, не дивлячись на підвищені вимоги до апаратури та системи, Spring має перевагу перед іншими фреймворками зростанням навантаження на сервер і в швидкодії. На відміну від своїх конкурентів, Spring це щось більше ніж фреймворк для веб розробки (за це відповідає тільки один модуль це Spring MVC), так на Spring можна працювати з хмарними технологіями, або підключити кластер серверів до проекту [26].

Враховуючи все вищевикладене та з урахуванням вибору мови програмування Java, буде доцільним обрати саме фреймворк Spring для серверної частини мобільного застосунку. Spring Security буде використовуватися для реалізації серверної безпеки, а також реєстрації та авторизації.

1.8 Вибір системи керування базами даних

Вибір системи керування бази даних (СКБД) будемо здійснювати шляхом порівняльного аналізу трьох найбільш популярних баз даних: SQLite, MySQL і PostgreSQL.

SQLite є бібліотекою, скомпонованою разом із застосуваннями, що використовують її [27]. Як і всі вбудовані бази даних, *SQLite* надає значний набір інструментів для роботи з різними наборами даних при зменшених накладних витратах порівняно з базами даних, побудованих за клієнт-серверною моделлю. Завдяки тому, що вся база даних зберігається в єдиному файлі, *SQLite* є дуже швидкою і ефективною СКРБД, яка використовується як у вбудованих системах, так і на виділених машинах.

Переваги *SQLite*: вся база складається з єдиного файлу даних, що забезпечує найвищий рівень переносимості; незважаючи на свою полегшену структуру, в основі *SQLite* лежить SQL і його стандарти; *SQLite* зручна для розробників додатків, оскільки складається всього з одного файлу, а її бібліотека написана на Сі і має безліч прив'язок до інших мов програмування.

Недоліки *SQLite*: відсутність розмежування прав доступу; неможливість використання в великих проектах; неможливість підвищення ефективності.

MySQL – найпопулярніша база даних для серверів з високим навантаженням. Це багатофункціональний програмний продукт з відкритим кодом, який взаємодіє з величезним числом сайтів і веб-застосунків. Крім того, *MySQL* характеризується порівняно низьким порогом входження. Незважаючи на неповну імплементацію стандартів SQL, *MySQL* може похвалитися дуже високим функціоналом [28]. *MySQL*, на відміну від *SQLite*, реалізована відповідно до специфікації клієнт-серверної архітектури, тому додатки звертаються до бази даних за допомогою служби *MySQL*.

Переваги *MySQL*: легкість установки і наявність візуальних засобів управління; підтримка, прямо або побічно, великого SQL-функціоналу; наявність вбудованих засобів безпеки; масштабованість і робота з великими обсягами даних; ігнорування певних стандартів збільшує швидкість роботи і продуктивність *MySQL*.

Недоліки *MySQL*: БД далеко не універсальна і володіє певними обмеженнями функціоналу, які «подобаються» далеко не всіх програм; не так на-

дійна, як хотілося би, у порівнянні з іншими СКРБД; незважаючи на те, що технічно MySQL являє собою програмне забезпечення з відкритим кодом, процес її розробки фактично припинений.

СКБД MySQL використовують у разі необхідності здійснення великої кількості розподілених транзакцій; реалізації вимог високої надійності; підвищених вимог до гнучкості і універсальності для роботи з сайтами і веб-застосунками; у разі відповідності вимогам індивідуального проекту. Краще не використовувати MySQL якщо існує необхідність точного відповідності стандартам SQL, при необхідності багатопотоковості і паралелізму, а також при підвищених вимогах до функціоналу – наприклад, необхідність наявності повноцінного текстового пошуку.

PostgreSQL – це найбільш просунута об'єктно-реляційна база даних. Відмінністю PostgreSQL від інших СКБД є інтеграція об'єктно-орієнтованої та / або реляційної моделей. Завдяки своїй структурі, PostgreSQL має високий рівень паралелізму і вражаючою продуктивністю. Незважаючи на те, що PostgreSQL не так популярна, як MySQL, існує безліч програм сторонніх виробників, що значно полегшують роботу з цієї надзвичайно потужної СКБД [29].

Переваги PostgreSQL: PostgreSQL – це ПЗ з відкритим вихідним кодом, що відповідає всім стандартам СКРБД; велике співтовариство користувачів; наявність великої кількості застосувань сторонніх виробників для настройки та управління; розширення функціоналу за рахунок створення збережених процедур; PostgreSQL – НЕ реляційна, а об'єктно-реляційна база даних, що підтримує всі властивості ООП.

Недоліки PostgreSQL: незадовільна продуктивність на простих операціях читання; не користується особливою популярністю, незважаючи на розвинену співтовариство; мало хостерів, що надають підтримку PostgreSQL.

СКБД PostgreSQL слід використовувати у разі необхідності виконання великої кількості призначених для користувача процедур, коли пріоритетом є надійність зберігання даних і їх цілісність; при запланованій міграції на пла-

тні СКРБД (наприклад, Oracle); при наявності складних структур даних. Швидкість читання не відноситься до сильних сторін PostgreSQL. Якщо дана умова існує, то PostgreSQL – не найкращий вибір.

Результати порівняльного аналізу розглянутих баз даних наведені в табл.1.3.

Таблиця 1.3 – Порівняльний аналіз СКБД

	MySQL	PostgreSQL	SQLite
Архітектура	Клієнт-сервер	Клієнт-сервер	Вбудована БД
Підтримка веб-хостингами (веб-серверами)	Дуже гарна. Переважна більшість хостингів PHP мають підтримку MySQL	Задовільна. Підтримка PostgreSQL часто зустрічається	Задовільна. Підтримка SQLite часто зустрічається, можна направити запит в підтримку про установку необхідного модуля
Документація	Англомовна на офіційному сайті		
Простота розробки та адміністрування	Адміністрування через консоль, безліч додатків для різних платформ для локального адміністрування, адміністрування через веб-сервер		
Надійність збереження даних	Надійність забезпечується сервером БД. Є механізми резервного копіювання		Вся БД зберігається в одному файлі

Таким чином, за результатами порівняльного аналізу була обрана потужна об'єктно-реляційна база даних PostgreSQL, що підтримує всі властивості ООП, а також забезпечує надійність зберігання даних та їх цілісність.

2 ПРОЄКТУВАННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Мета та задачі ІС

Розробляема інформаційна система – це мобільний застосунок, основною функцією якого є надання користувачам сервісу контекстно-залежного редагування зображень.

Для модифікації зображень буде використовуватися алгоритм контентно-залежного масштабування, який дозволяє змінювати розмір зображення за рахунок пошуку сімів (швів) – найменш важливих пікселів для подальшого їх видалення чи дублювання. Цей алгоритм базується на методі Seam carving, який використовує функцію енергії, що визначає важливість пікселей. Окрім зміни розміру цей підхід застосовується і для видалення зайвих об'єктів, визначених користувачем.

Мета ІС: виконати масштабування та модифікацію зображення з урахуванням його контенту за визначеними користувачем розмірами. Застосунок має бути зручним для кожного користувача та зберігати модифіковані зображення в його особистому кабінеті.

Цільова аудиторія: будь-які користувачі, що бажають виконати редагування власних зображень.

Головними функціями мобільного застосунку є:

- модифікування, наприклад, видалення зайвих об'єктів, зі збереженням загального контексту зображення;
- контекстно-залежне масштабування зображення по горизонталі;
- контекстно-залежне масштабування зображення по вертикалі;
- контекстно-залежне масштабування зображення зі збереженням оригінального співвідношення сторін;
- можливість визначення власного розміру зображення;
- можливість збереження зображень у БД.

Інформаційні потоки системи зображено на рисунку 2.1.

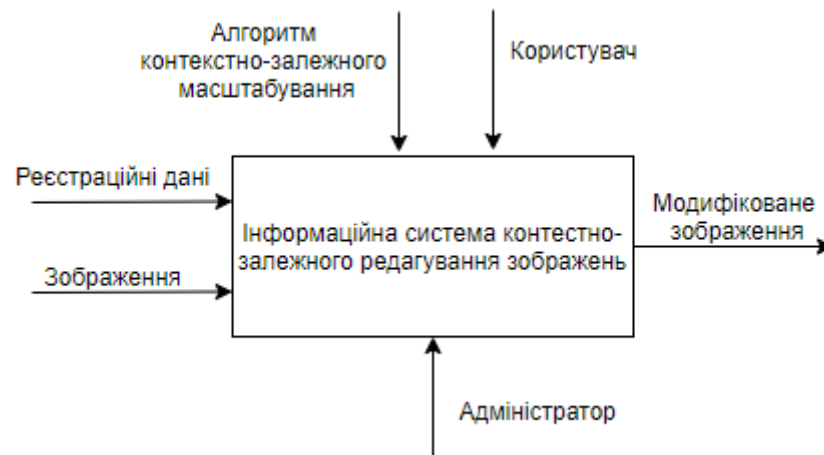


Рисунок 2.1 – Діаграма інформаційних потоків

2.2 Типи користувачів

У мобільного застосунку заплановано три типи користувачів – це адміністратор, користувач сервісу та гість.

Адміністратор – здійснює модерацію системи, може видалити користувачів та зображення з бази даних.

Користувач – може авторизуватися, змінити розмір зображення з врахуванням контенту, видалити зайвий об’єкт з зображення, зберегти/переглянути/видалити зображення.

Гість – може зареєструватися, змінити розмір зображення з врахуванням контенту, видалити зайвий об’єкт з зображення.

2.3 Функціональні вимоги

1. Реєстрація в застосунку

F1.1 Функція реєстрації за допомогою форми реєстрації.

Дана функція доступна користувачу Гість.

FR1.1.1 Користувач повинен заповнити форму реєстрації у якій є поля логін, і пароль. Інакше має спливати підказка, що якийсь поле не заповнено.

FR1.1.2 При успішній авторизації користувач перенаправляється на головну сторінку системи.

FR1.1.3 У випадку некоректного вводу паролю система виводить повідомлення: «Пароль має бути не менше 8 символів та не більше 64 символів, а також містити у собі латинські літери, цифри та спеціальні знаки, наприклад @{}|\.»

FR1.1.4 У випадку введення неправильного логіна система виводить повідомлення: «Логін має бути не менше 6 символів та не більше 64 символів».

FR1.1.5 У випадку, коли користувач з введеним логіном вже існує, система виводить повідомлення: «Користувач з вказаним логіном вже існує. Спробуйте інший логін»

2. Авторизація в застосунку

F2.1 Функція авторизації за допомогою форми авторизації

Дана функція доступна користувачам Адміністратор та Користувач.

FR2.1.1 Користувачам необхідно заповнити поля для входу, для того щоб потрапити в свій кабінет. При успішній авторизації користувач перенаправляється на головну сторінку системи.

FR2.1.2 У випадку некоректного вводу даних система виводить повідомлення: «Невірний логін чи пароль»

FR2.1.3. Кожний користувач при авторизації повинен ввести логін та пароль.

FR2.1.4. При успішній авторизації користувач потрапляє в робочий кабінет

3. Управління акаунтом

F3.1 Функція редагування даних акаунту

Дана функція доступна користувачам Користувач.

FR3.1.1 Користувач може змінити будь-яке значення з полів: ім'я та прізвище, логін, пароль, номер телефону. Після успішного редагування даних, усі зміни одразу відображуються на акаунті.

FR3.1.2 У випадку некоректного вводу логіну система виведе повідомлення: «Логін має бути не менше 6 символів та не більше 64 символів»

FR3.1.3 У випадку некоректного вводу пароля система виведе повідомлення: «Пароль має бути не менше 8 символів та не більше 64 символів, а також містити у собі латинські літери, цифри та спеціальні знаки, наприклад @{} \.»

FR3.2 Функція видалення акаунту.

Дана функція доступна користувачам Користувач.

FR3.2.1 Користувач має підтвердити або скасувати видалення свого акаунта у модальному вікні.

FR3.2.2 У разі згоди акаунт користувача та всі дані пов'язані з ним будуть видалені з системи.

FR3.2.3 У разі відмови користувач буде перенаправлений на сторінку його акаунта, при цьому усі дані пов'язані з ним не будуть видалені з системи.

4. Управління зображеннями

FR4.1 Функція зміни розміру зображення з врахуванням контенту

Дана функція доступна користувачам Гість, Користувач.

FR4.1.1 Користувачі може змінити розміри завантаженого у систему зображення, обрав нові розміри в пікселях по горизонталі та/чи по вертикалі.

FR4.1.2 Користувач має можливість зручного вводу нових розмірів зображення.

FR4.1.3 У випадку некоректного вводу розмірів зображення, система покаже повідомлення: «Вказані некоректні розміри»

FR4.1.4 У випадку успішного завдання користувачем розмірів зображення та при натисканні відповідної кнопки запуску процесу система ство-

рює модифіковане зображення, яке додається на сторінку перегляду зображень.

F4.2 Функція видалення зайвих об'єктів з зображення

Дана функція доступна користувачам Гість, Користувач.

FR4.2.1 Користувач може видалити об'єкти, які він виділив на зображенні.

FR4.2.2 Користувач має можливість зручного вибору об'єктів, які потрібно видалити із зображення.

FR4.2.3 У випадку коли користувач не виділив жодного об'єкту і натискає відповідну кнопку, система покаже повідомлення: «Жодного об'єкту не обрано».

FR4.2.4 У випадку, коли користувач виділив об'єкти для видалення на зображенні та натиснув кнопку запуску відповідного процесу, система створює модифіковане зображення, яке додається на сторінку перегляду зображень.

F4.3 Функція видалення зображення

Дана функція доступна користувачу Адміністратор, Користувач.

FR4.3.1 Користувач має підтвердити або скасувати видалення зображення у модальному вікні.

FR4.3.2 У випадку підтвердження зображення буде видалено з системи.

FR4.3.3 У випадку скасування система перенаправить користувача на сторінку з усіма його зображеннями.

F4.4 Функція збереження зображення

Дана функція доступна користувачу Користувач.

FR4.4.1 Користувач має підтвердити або скасувати збереження зображення у модальному вікні.

FR4.4.2 У випадку підтвердження зображення буде збережено у БД.

FR4.4.3 У випадку скасування система перенаправить користувача на сторінку з усіма його зображеннями.

F4.5 Функція перегляду зображень

Дана функція доступна користувачу Користувач.

FR4.5.1 Користувач має підтвердити або скасувати запит на перегляд усіх своїх зображень у модальному вікні.

FR4.5.2 У випадку підтвердження зображення будуть завантажені з БД у вікно перегляду.

FR4.5.3 У випадку скасування система перенаправить користувача на головну сторінку.

FR4.5.4 Користувач має можливість зручного перегляду зображень.

F5. Управління користувачами

F5.1 Функція видалення акаунта користувача

Дана функція доступна користувачу Адміністратор.

FR5.1.1 Користувач має підтвердити або скасувати видалення акаунта Користувача у модальному вікні.

FR5.1.2 У випадку підтвердження акаунт Користувача буде видалено з системи.

FR5.1.3 У випадку скасування система перенаправить користувача на головну сторінку.

2.4 Нефункціональні вимоги

NF1. Серверне застосування працює лише при наявності підключення до мережі Інтернет.

NF2. Застосунок має запускатися на смартфонах з ОС Android версії 5.1 і вище.

NF3. Надійність бази даних визначається токеном шифрування.

NF4. Система повинна мати зручний інтерфейс та швидко працювати.

NF5. Система доступна цілодобово за посиланням, при умові безперебійної роботи хостингу та проплаченого домену.

Вимоги визначають можливості застосування, отже на підставі списку вимог можна побудувати діаграму варіантів використання ІС.

Варіант використання, прецедент (англ. Use Case) – у розробці програмного забезпечення та системному проектуванні це опис поведінки системи, як вона відповідає на зовнішні запити. Іншими словами, різновид використання описує, «хто» і «що» може зробити з розглянутою системою.

Для детальнішого розгляду процесу є можливість побудови більш детальної діаграми використання, яка зображена на рис.2.2.

Ця діаграма надасть більше розуміння про можливі варіанти поведінки користувача у обраному сценарії. Аналогічні діаграми можуть бути запрошені для будь-якого процесу.

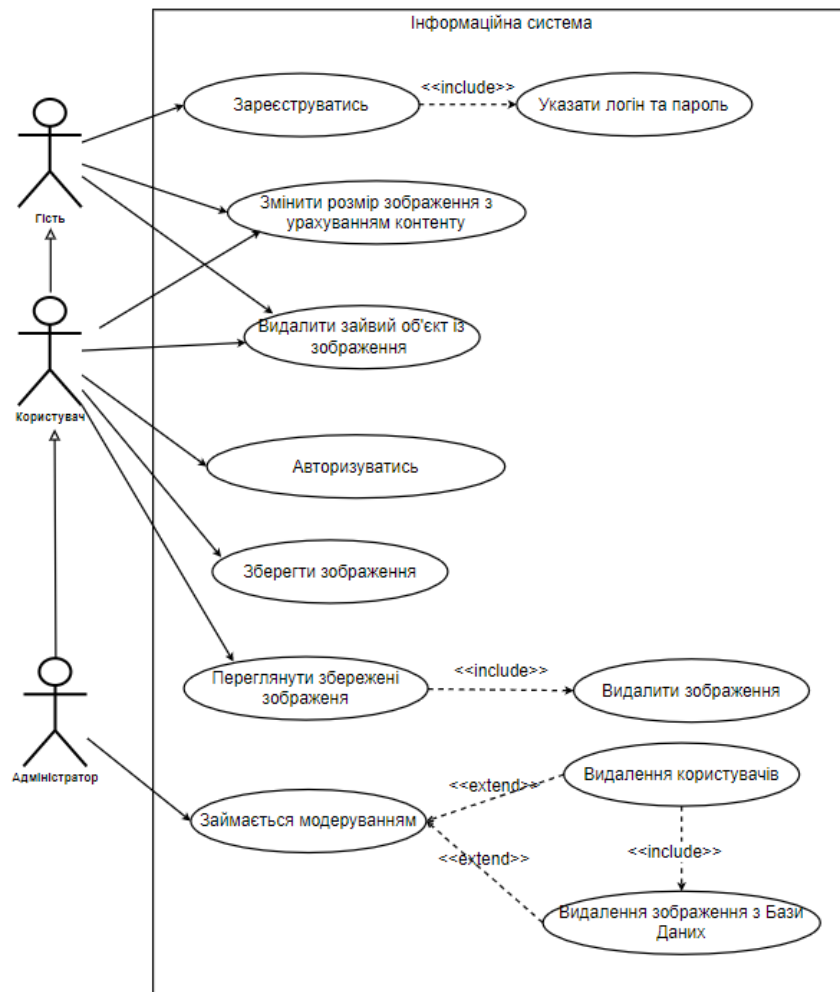


Рисунок 2.2 – Діаграма прецедентів

2.5 Ідентифікація архетипу ІС

Архетип даної системи – Mobile Application. Мобільний застосунок розроблений під платформу Android.

Система для контентно-залежного редагування зображень призначена для модифікації зображень, які часто є фотографіями отриманими за допомогою камери смартфонів користувачів, з метою розміщення у соціальних мережах, тому обґрунтованим є вибір саме архетипу Mobile Application для ІС, що розроблюється.

2.6 Користувацький інтерфейс (UI View)

При відкриванні застосунку користувач потрапляє на головний екран (рис. 2.3 а). Користувачу пропонується вийти у свій акаунт (Sign In), або, якщо користувач ще незареєстрований, пройти етап реєстрації (Sign Up). Екран реєстрації користувача зображений на рис.2.3 б.

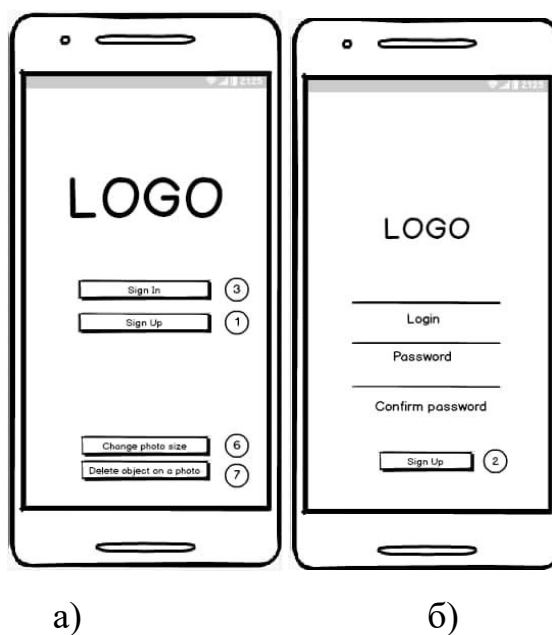


Рисунок 2.5 – Мокапи інтерфейсу застосунку: а) головна сторінка;
б) сторінка реєстрації

Якщо користувач вже зареєстрований у застосунку, то він може потрапити на його головну сторінку вказавши свій логін і пароль на сторінці авторизації (рис.2.6 а).

У випадку успішного входу на головній сторінці застосунку відображується привітання, а також стають активними кнопки управління зображеннями: завантажити зображення, змінити розмір зображення та видалити зайві об'єкти на зображенні.

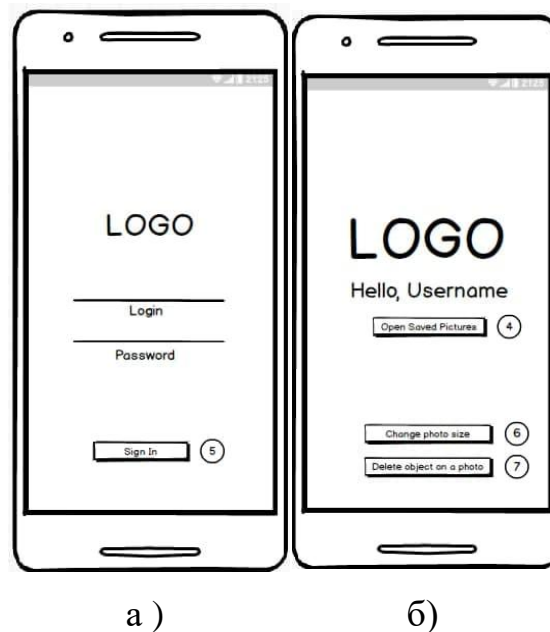


Рисунок 2.6 – Мокапи інтерфейсу застосунку: а) сторінка входу;
б) сторінка привітання

На рис. 2.7 а та 2.7 б наведені мокапи вікон редагування зображень: зміни розміру та вибору об'єктів для видалення відповідно.

Вікна містять зручні вбудовані інструменти для здійснення відповідних модифікацій.

Після вибору нових розмірів зображення, користувач може натиснути

кнопку Upload для перегляду результату. Аналогічно після вибору на зображенні об'єктів для видалення, користувач може натиснути кнопку Delete Object для перегляду відредагованого зображення.

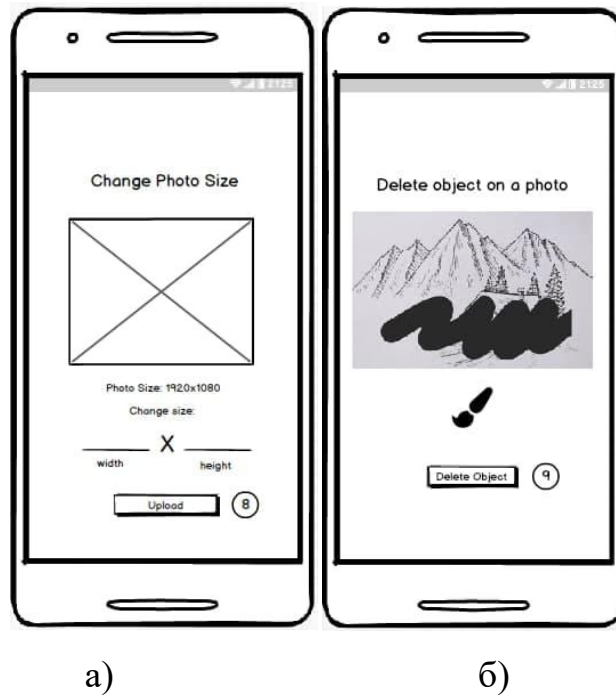


Рисунок 2.7 – Мокапи інтерфейсу застосунку: а) вікно вибору розміру зображення; б) вікно вибору об'єктів для видалення

На рис 2.8 а наведений мокап вікна перегляду результатів редагування зображення. В даному вікні користувач також може зберегти отримане зображення у своєму смартфоні чи в БД за стосунку натиснувши відповідну кнопку.

Всі збережені у БД зображені доступні користувачу для перегляду у вікні перегляду зображень, мокап якого наведено на рис.2.8.

На рисунку 2.9 зображено діаграму станів переходів між сторінками мобільного застосунку.

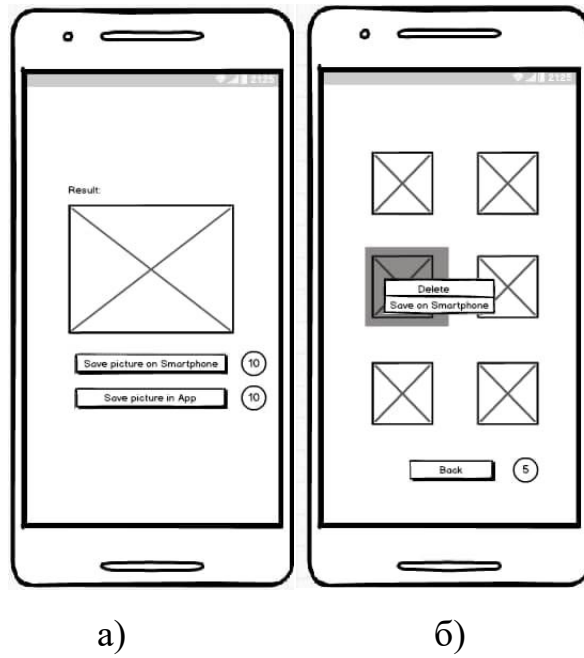


Рисунок 2.8 – Мокапи інтерфейсу застосунку: а) вікно перегляду результатів редагування зображення; б) вікно перегляду збережених зображень

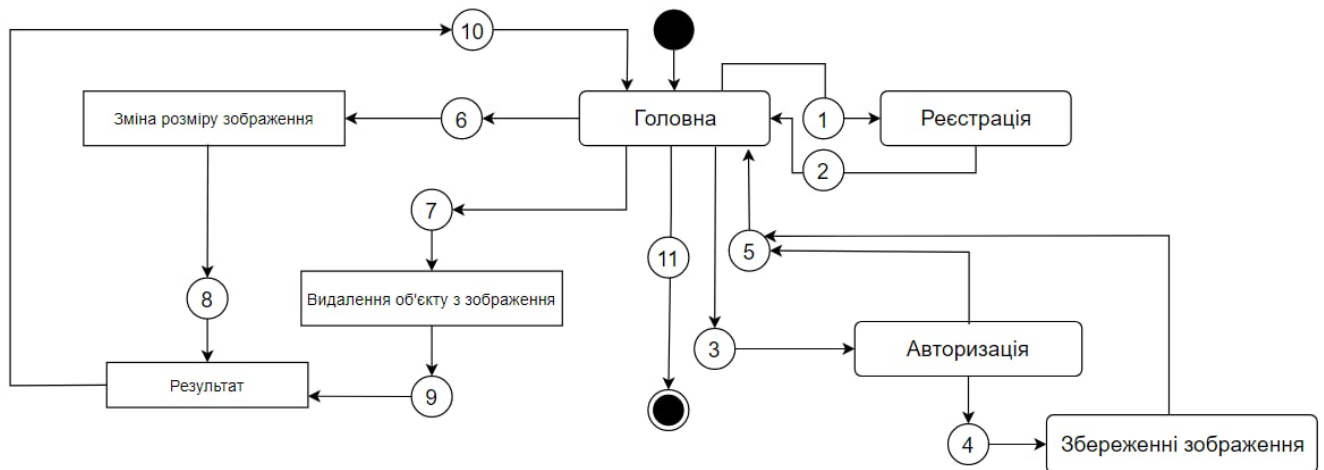


Рисунок 2.7 – Діаграма станів переходів між сторінками

2.7 Уявлення розгортання ІС (Deployment View)

Застосовання працює на смартфоні та на веб-сервері. На веб-сервері було створено ряд функцій, які виконують основну частину завдання, а саме отримання та редагування зображень. Функції одна за одною виконують дії над сутністю даних та після остаточної обробки формують результат у вигляді модифікованого зображення, переданого до клієнтського застосунка.

Для демонстрації цього процесу була побудована діаграма розгортання (рис 2.8), яка дозволяє побачити наявність фізичних з'єднань – маршрутів передачі інформації між апаратними пристроями, задіяними в реалізації системи.

Діаграма розгортання призначена для візуалізації елементів і компонентів програми, які існують лише на етапі її виконання.

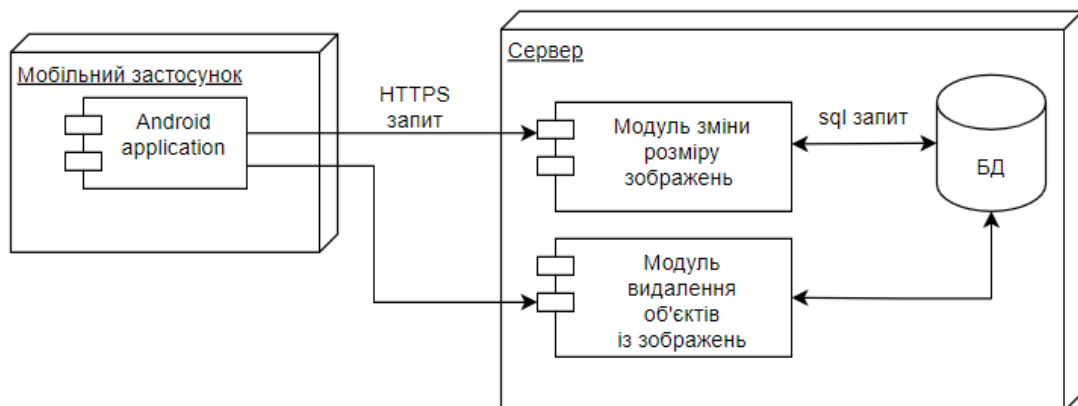


Рисунок 2.8 – Діаграма розгортання (DeploymentView)

2.8 Уявлення даних ІС (DataView)

Список класів, що використовуються при роботі з БД (рис.2.9):

- User (містить інформацію про користувачів системи);
- Image (містить інформацію про зображення);

– Album (містить інформацію про збережені у БД зображення – альбом користувача).

Для наочного представлення структури локальної бази даних побудовано схему БД (рис.2.9). У базі зберігається інформація про користувачів. Може бути лише один користувач з унікальним ідентифікатором. Він може мати безліч зображень (зв'язок один до багатьох). Клас Album дозволяє встановити зв'язок між користувачами та їх зображеннями.

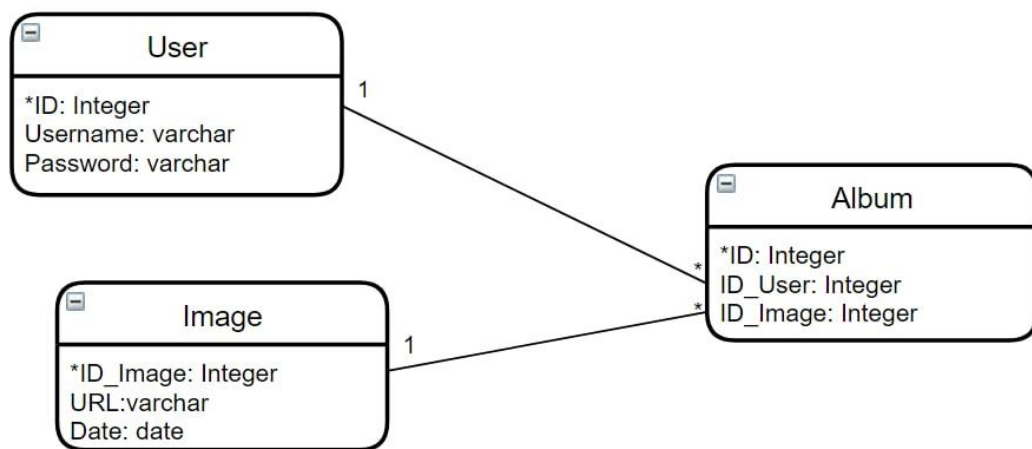


Рисунок 2.9 – Схема бази даних застосунка

Специфікація БД:

– тип зберігання даних буде база даних – PostgreSQL, призначена для збереження інформації про користувачів та їх зображення;

– для роботи з БД будуть використовуватись додаткові класи – класи DAO, що дозволить додавати, редагувати, створювати записи у базі та встановлювати безпосередній зв'язок з БД.

3. РЕАЛІЗАЦІЯ СИСТЕМИ

3.1 Уявлення про структуру проекту ІС

Android проект представлений 3 корневими папками: app, src та Gradle Scripts. Папка app включає в собі 3 підпапки:

- manifests містить файли конфігурацій або файли маніфесту за стосунку;
- в java знаходиться вихідний код програми;
- res містить файли використовуваних в Android застосунку ресурсів (картинки, стилі, розмірності для різних пристроїв і т.д.).

На рис.3.1 представлений вміст папки java. В цій папці знаходяться класи типу Fragment та Activity, які відповідають за графічний інтерфейс мобільного застосунку.

В папці Model знаходяться класи в яких реалізована робота з даними, запит даних з сервера, збереження в БД.

В папці Repository знаходяться класи DAO, які відповідають за виконання патерну для збереження об'єктів бізнес-області в базі даних. В цьому класі міститься CRUD методи для конкретної сутності.

3.2 Уявлення про класи ІС

На рисунку 3.2 зображена діаграма класів інформаційної системи. Вона складається з дуже великої кількості класів, тому було вирішено обрати лише класи, що реалізують алгоритм контекстно-залежного редагування зображень Seam Carving.

Розроблена програма складається з двох окремих модулів:

1) Зміни розміру зображення – збільшення та зменшення розміру зображення, враховуючи його вміст на основі обчислення градієнтних функцій енергії та оператора Собеля з побудовою карти енергії на його основі. Функція зміни розміру зображення реалізована в класі ChangeSize.

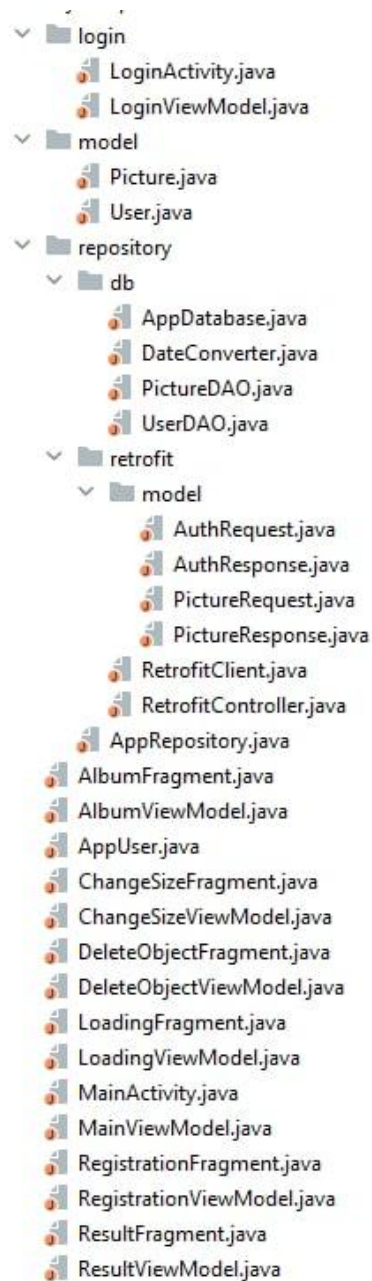


Рисунок 3.1 – Структура проекту

2) Видалення зайвих об'єктів з зображення – для видалення об'єктів користувачу треба виділити цільовий об'єкт, який потрібно видалити. Пікселям цього об'єкту призначається мінімальна енергія. Потім видаляються «сіми» із зображення, поки не зникнуть усі виділені пікселі. Функція Видалення зайвих об'єктів реалізована в класі DeleteObject.

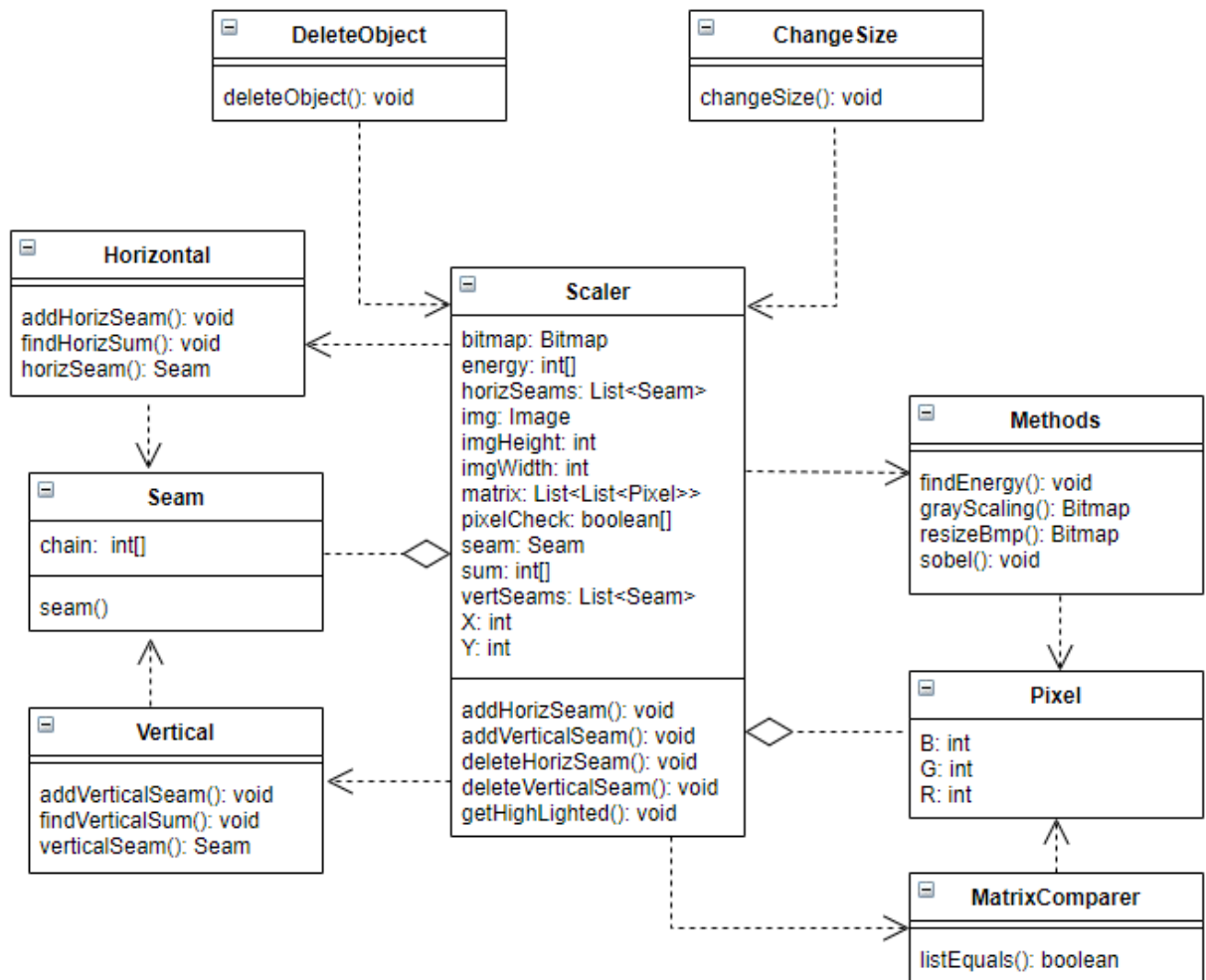


Рисунок 3.2 – Діаграма класів IC

Клас `Seam` призначений для створення об'єкту, що інкапсулює властивості окремого сіму. Для побудови горизонтального і вертикального ланцюжку сімів (набору пікселів з найменшим сумарним значенням енергії) створені класи `Horizontal` та `Vertical`, які містять методи, що дозволяють знайти наступний у шляху сім і повернути його як об'єкт класу `Seam`.

Для реалізації алгоритму знаходження карти енергії пікселів з використанням градієнтного значення `findEnergy()` та оператора Собеля `sobel()` створені спеціальні методи, що знаходяться у класі `Methods`. Шлях сімів представлений в програмі у вигляді зв'язаного списку, що містить об'єкти типу `Seam`. Для кольорового зображення створена структура для збереження значень каналів R, G і B – атрибутів класу `Pixel`. Клас `Scaler` – це клас, що мето-

ди для масштабування зображень, він дозволяє видаляти і додавати знайдені сіми.

3.3 Інфраструктурне уявлення ІС (Infrastructure View)

Для розгортання інформаційної системи необхідна операційна система Android мінімальної версії 4.1. Пристрій повинен мати підключення до мережі Інтернет з мінімальною швидкістю.

Сервер розгорнутий на хмарній платформі Heroku. За правилами безкоштовного користування платформою будуть визначені наступні параметри: RAM – 512 МБ. Також PostgreSQL у безкоштовній версії має лише 1 Гб обсягу сховище та до неї можна відключитися не більше 20 разів одночасно.

3.4 Документація ІС

Для роботи системи треба наявність смартфона з операційною системою Android 4.1 і вище та підключення до мережі Інтернет. Користувач має можливість скачати мобільний застосунок на сторінці у Google Play.

Інструкція по збірці програмного коду проекту. Для збірки проекту потрібно в системі Android Studio створити .apk файл та запустити його на віртуальному емуляторі чи на фізичному пристрої (смартфоні).

При завантаженні мобільного застосунка Pihover з'являється екран (рис.3.3 а), який пропонує користувачу авторизуватися чи пройти реєстрацію, якщо це гість. Екран реєстрації гостя представлений на рис.3.3б. Гість повинен заповнити форму реєстрації у якій є поля логін, і пароль. Інакше спливає підказка, що якесь поле не заповнено. У випадку некоректного вводу паролю і логіна система виводить повідомлення з правилами їх вводу, наприклад, «Логін має бути не менше 6 символів та не більше 64 символів». У випадку, коли користувач з введеним логіном вже існує, система виводить повідом-

лення: «Користувач з вказаним логіном вже існує. Спробуйте інший логін». При успішній авторизації користувач перенаправляється сторінку з привітанням (рис.3.3 в).

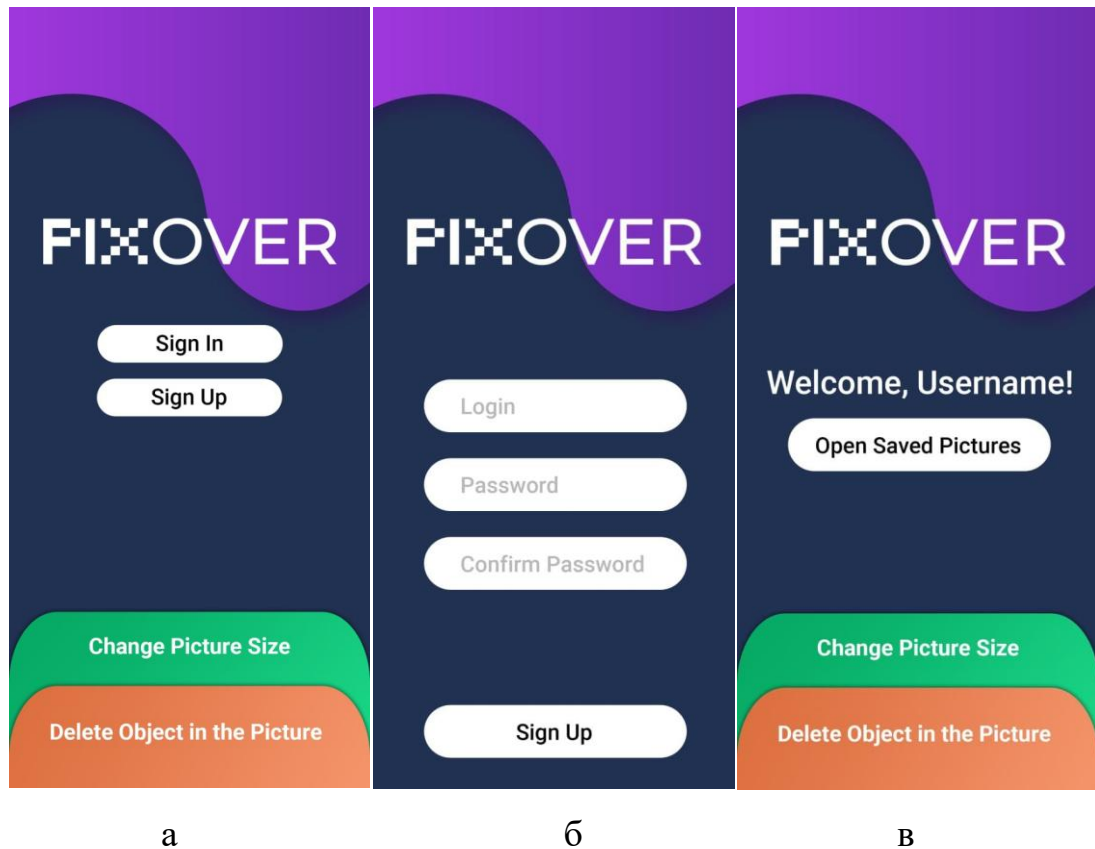


Рисунок 3.3 – Інтерфейс застосунку Fixover: а) екран входу; б) екран реєстрації; в) головна сторінка ІС

На рис. 3.4 представлені результати роботи застосунку, що демонструють його можливості в редагуванні зображень. Користувач має можливість завантажити зображення зі смартфона чи БД системи та вказати його нові розміри (рис.3.4а). Натиснув кнопку «Change Size» користувач зможе отримати масштабоване зображення з новими розмірами (рис.3.4б). В програмі реалізовано алгоритм Seam carving, що представляє собою підхід зміни розміру зображення, що усвідомлено видаляє чи додає пікселі неважливих частин зображення, дозволяє зберігати основний контент зображення.

Користувач має можливість зберегти отримане нове зображення в БД системи.

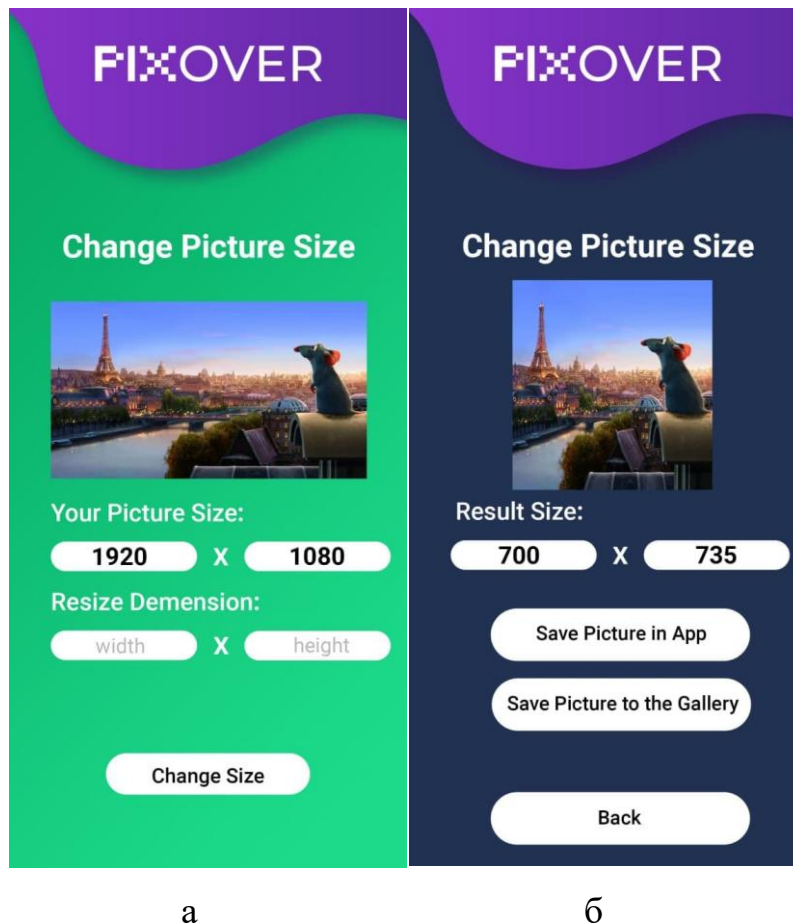


Рисунок 3.4 – Інтерфейс застосунку РіхOVER: а) сторінка зміни розміру зображення; б) результат масштабування зображення

Алгоритм Seam carving також застосовується для видалення об'єктів з зображення. Реалізовано наступну обчислювальну схему:

- 1) Будується енергетична карта зображення.
- 2) Визначається кумулятивна карта енергії зображення.
- 3) Встановлюються мінімальні значення енергії для обраних користувачем піксель.
- 4) Для кожної ітерації:

- визначається оптимальний «сім» (в потрібному напрямку – горизонтальний чи вертикальний);
- видаляється оптимальний «сім» із зображення, тим самим зменшуючи розмір зображення.

Приклади результату видалення об'єктів наведено на рис. 3.5. На рис.3.5 а користувач виділів зайвий об'єкт, який був видален з зображення (рис.3.5б).

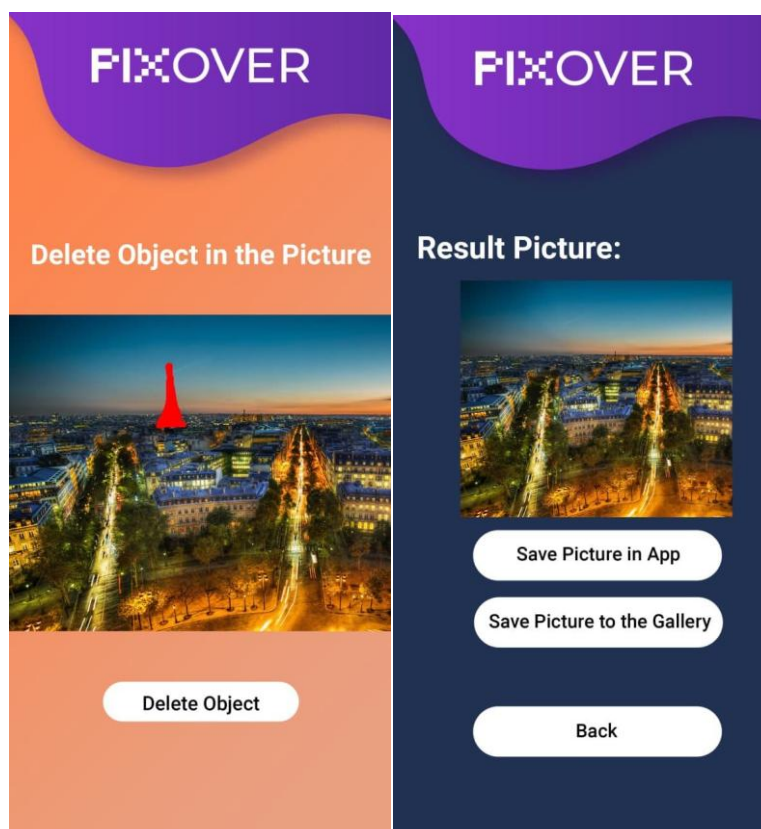


Рисунок 3.5 – Інтерфейс застосунку РіхOVER: а) сторінка видалення об'єкту; б) результат видалення об'єкту з зображення

ВИСНОВКИ

В процесі виконання бакалаврської роботи були закріплені як теоретичні, так і практичні навички проектування та реалізації біометричної системи доступу працівників із автоматичним температурним скринінгом.

В роботі був проведений аналіз існуючих аналогів, що надають можливість ідентифікувати людину за біометрикою та контролювати графік роботи працівників на підприємстві, на підставі якого визначені їх переваги та недоліки для створення власного бачення оптимальної системи.

Було розглянуто та обрано технології для створення продукту згідно з сучасними тенденціями, а також беручи до уваги важливість найменшої ціни при розробці. Система складається з трьох частин: вебсервер, фронтенд частина та апаратна частина з камерою за модифікованою версією термометру.

Сервер написано на Java за допомогою фреймворку Spring, фронтенд частину розроблено за допомогою Angular. Щодо апаратної частини, то був використаний мікрокомп'ютер Raspberry Pi для створення запитів та використання нейромережі. Для зберігання даних використано СУБД MySQL.

В роботі було сформовано функціональні та нефункціональні вимоги до інформаційної систем. Побудовано діаграму прецедентів, визначені типи користувачів та їх функції. Також було розроблено мокапи майбутнього застосунку з потрібним функціоналом.

Усю роботу ПП було описано у діаграмах – логічному уявленні, уявленні розгортання, слоїв, процесів, даних. Роблено реляційну модель бази даних. Прототип системи був реалізований та протестований на власних зображеннях.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Дудзяний І.М. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007.– 108 с.
2. Java – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/Java> (дата звернення 19.05.2023)
3. Коноваленко І.В., Марущак П.О. Платформа .NET та мова програмування C# 8.0: навчальний посібник, Тернопіль: ФОП Паляниця В. А., 2020. 320 с.
4. Цеслів О.В. Основи програмування та веб-дизайн: Навч. посіб. – К.,2020. 149с. URL: https://ela.kpi.ua/bitstream/123456789/40499/1/OP_veb-dyzain.pdf (дата звернення 19.05.2023)
5. Python – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/Python> (дата звернення 19.02.2021)
6. Spring Framework – Вікіпедія. (загол. з екрана). URL:https://uk.wikipedia.org/wiki/Spring_Framework (дата звернення 14.04.2021)
7. Керівництво з Spring. URL: <https://proselyte.net/tutorials/spring-tutorial-full-version/introduction/> (дата звернення 14.04.2023)
8. PostgreSQL – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/PostgreSQL> (дата звернення 14.04.2021)
9. MySQL – Вікіпедія. (загол. з екрана). URL: <https://uk.wikipedia.org/wiki/MySQL> (дата звернення 14.04.2021)
10. Табунщик Г. В. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.

11. Авраменко В.С., Авраменко А.С. Проектування інформаційних систем: навчальний посібник. Черкаси: Черкаський національний університет ім. Б. Хмельницького, 2017. – 434 с.: іл.
12. Raspberry Pi 3 Model B. URL: <https://uk.rs-online.com/web/generalDisplay.html?id=raspberrypi> (дата звернення: 14.05.2023)
13. Blum, Jeremy (2019). Exploring Arduino: Tools and Techniques for Engineering Wizardry (2nd ed.). Wiley. ISBN 978-1119405375
14. Monk, Simon (2022). Programming Arduino: Getting Started with Sketches (3rd ed.). McGraw-Hill Education. ISBN 978-1264676989
15. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. Житомир : Вид. О. О. Євенок, 2020. 184 с